On Mathematical Optimization for the Visualization of Frequencies and Adjacencies as Rectangular Maps

Emilio Carrizosa, Vanesa Guerrero, and Dolores Romero Morales

Journal article (Accepted manuscript)

CITE: On Mathematical Optimization for the Visualization of Frequencies and Adjacencies as Rectangular Maps. / Carrizosa, Emilio; Guerrero, Vanesa; Morales, Dolores Romero. In: European Journal of Operational Research, Vol. 265, No. 1, 2018, p. 290-302.

DOI: 10.1016/j.ejor.2017.07.023

Uploaded to CBS Research Portal: January 2019

On Mathematical Optimization for the visualization of frequencies and adjacencies as Rectangular Maps

Emilio Carrizosa[1], Vanesa Guerrero*[1], and Dolores Romero Morales[2]

[1]Instituto de Matemáticas de la Universidad de Sevilla (IMUS), Seville, Spain
{ecarrizosa, vguerrero}@us.es
[2]Copenhagen Business School, Frederiksberg, Denmark
drm.eco@cbs.dk

Abstract

In this paper 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. 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 at its core. Our experimental results demonstrate that our matheuristic provides rectangular maps with a good fit in both the frequency distribution and the adjacency relation.

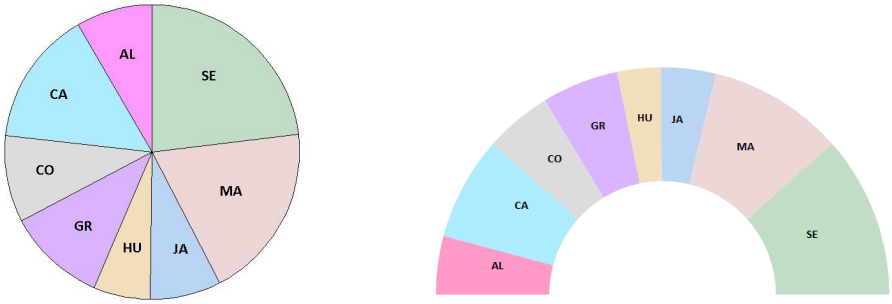
Keywords: Mixed Integer Linear Programming, Visualization, Multidimensional Scaling, Rectangular Maps, Frequencies and Adjacencies

*Corresponding author

1 Introduction

It is critical to enable analysts to observe and interact with data, using appropriate visualization tools, [30, 36]. Operations Research arises as a powerful area of knowledge to give answers to new challenges in Visualization, [41, 43].

A natural and frequent task is to depict a set of individuals $V = \{v_1, \ldots, v_N\}$, to which there is attached a frequency distribution, $\omega = (\omega_1, \ldots, \omega_N)$, with $\sum_{r=1}^{N} \omega_r = 1$, see, e.g., [49]. Market share, vote intention or population rates, just to name a few, are usual examples. In order to visualize frequencies, a common approach 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 (a) and (b) respectively, and rectangular maps [5, 26], see Figures 1 (c) and (d). In this kind of representations, holes are not allowed, thus, receiving the name of planar space-filling visualization maps.



(a) Pie Chart

(b) Fan Chart

(c) Rectangular map

(d) Rectangular map

Figure 1: Examples of planar space-filling visualization maps

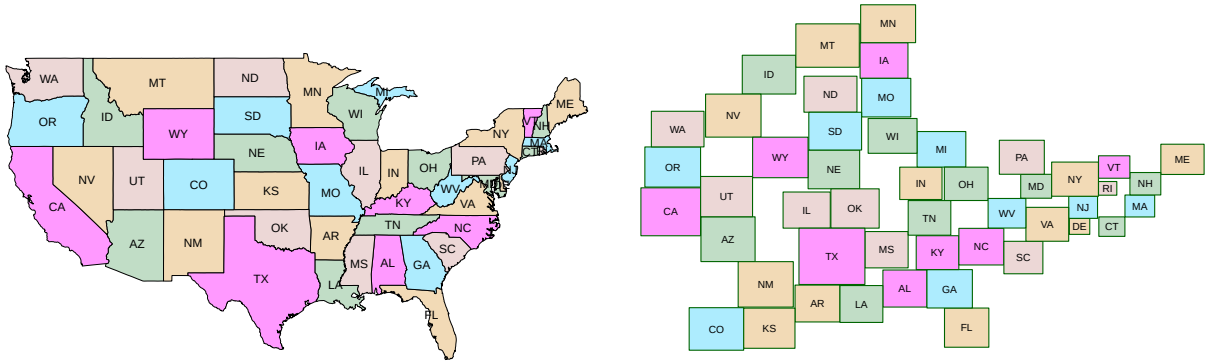
A planar space-filling map to visualize the frequencies attached to individuals in a bounded set Ω of the plane can be found by constructing the portions of the desired area and putting them together to fill Ω. This is straightforward in the case of the pie or fan charts: for a permutation $\sigma(1), \sigma(2), \ldots, \sigma(N)$ 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 in [25]. For the case of rectangular maps, where Ω is the unit

square and portions are rectangles, the same approach, illustrated in Figure 1 (c), 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 1 (d). The flexible layout offered by rectangular maps is also desirable when, in addition to frequencies, we are interested in visualizing proximity, measured by adjacencies, which is the subject of this paper.

The nature of the proximity can be diverse, a classical example being geographical proximity. A well-known problem in Cartography is the representation of geographical regions with relatively simple shapes, such as rectangles, whose areas represent a magnitude such as population rates or vote intention, as well as the relative position between regions is maintained, [42, 55]. One of the most popular visualization tools for this is rectangular cartograms, which were first introduced in [46] and have been further investigated in, e.g., [8, 19, 26, 33]. The approaches developed in the literature to obtain rectangular cartograms take advantage of the geographical relative positions of the individuals (countries, cities, etc.) to obtain a cartogram, and thus their approaches cannot be directly extended to more general data structures. Figure 2 depicts a rectangular cartogram for the geographical area of the states in the U.S. built using the Recmap package in R [45], see Figure 2 (b).

When dealing with proximity, a common approach in the literature has been to represent *close* individuals as *close* portions in the visualization map, see [1, 9, 10, 18, 24, 25] and references therein. A grid map [20] is a visualization tool that represents 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 (c) depicts the grid map built for the 48 contiguous states in the U.S., see Figure 6-L_2^2 in [20], representing 56 adjacencies of the 105 present in the actual map, see Figure 2 (a). With the methodology described in Section 4, we are able to represent 63 adjacencies of the 105 present in Figure 2 (a), see Figure 2 (d). In this paper, our goal is to propose a mathematical optimization formulation and a suitable solution approach to build rectangular maps to visualize the frequency distribution $\omega = (\omega_1, \ldots, \omega_N)$ and the proximity between the individuals, measured by an adjacency matrix $E = (e_{rs})$. As far as the authors are aware, this is a novel problem in the literature.

Throughout this paper, the weighted graph $G = (V, E, \omega)$ will model the set V of individuals, attached with the binary relation (adjacency) E and the frequency distribution ω. Similarly, we denote by $G^{\mathbf{P}} = (V, E^{\mathbf{P}}, \omega^{\mathbf{P}})$, 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, $(v_r, v_s) \in E^{\mathbf{P}}$ if portions P_r and P_s are adjacent, i.e., their borders intersect in more than one point, while for the node weights $\omega^{\mathbf{P}}$, $\omega_r^{\mathbf{P}}$ is equal to the area of the rectangle P_r. In general, one cannot guarantee the existence of a rectangular map that satisfies area and adjacency requirements on the rectangles, i.e., $\omega^{\mathbf{P}} = \omega$ and $E^{\mathbf{P}} = E$, see [33]. This is especially the case when the graph G to be represented is not planar. See [2, 6] for further 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, [18, 26, 57]. In this paper,



(a) The U.S. map

(b) Recmap for the U.S.

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	TX	SC	VA	DE
AZ	TX	LA	MS	AL	GA	FL	NC

(c) Grid map for the U.S. built in [20]

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

(d) Grid map for the U.S. built with the ECPA methodology in Section 4

Figure 2: Visualizations for the U.S.

we develop a new tool 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, [11].

Although not focused on 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 [3, 54]. 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_r and v_s are linked, then the corresponding portions P_r and P_s are adjacent in the subdivision. Some characterizations of planar graphs that admit a rectangular dual can be found in [6, 14, 32]. Rectangular duals are also related with Facility Layout, [4, 29, 47], whose aim is to find a layout which minimizes the flow between a set of

facilities of given areas, and Graph Drawing, [17, 31, 44, 53]. These frameworks use very ad-hoc approaches and either disregard the proper representation of adjacencies, frequencies, or are not space-filling.

In this paper, the problem of building rectangular maps which simultaneously 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 in e.g. [1, 20, 23, 38, 52], 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 MultiDimensional Scaling (MDS), [34], 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, [46].

Since our visualization model is a novel one, there are no ready available techniques for it in the literature. Therefore, we compare our rectangular maps with those obtained by solving the MILP formulation with a commercial solver under a time limit. In our experimental section we present results for three examples and conclude that we obtain a better fit in area and adjacency relation in less computing time.

The remainder of the paper is structured as follows. In Section 2 we introduce the optimization model to build (K, L)-rectangular maps. In Section 3 we formulate the problem as an MILP. In Section 4 we present an algorithm to compute (K, L)-rectangular maps. Section 5 is the experimental section. Section 6 concludes the paper with a summary and lines for future research. Finally, the Appendix includes the values of the frequencies for the three datasets considered in the experimental section.

2 The problem

Given a set of individuals $V = \{v_1, \ldots, v_N\}$, a (K, L)-rectangular map has associated a weighted graph $G^{\mathbf{P}} = (V, E^{\mathbf{P}}, \omega^{\mathbf{P}})$, in which $(v_r, v_s) \in E^{\mathbf{P}}$ if portions P_r and P_s are adjacent, i.e., they touch in at least one cell, and $\omega^{\mathbf{P}}$ denotes the rectangles' areas. An ideal (K, L)-rectangular map representation of a given graph $G = (V, E, \omega)$ 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_r is a rectangle made up of a collection of cells of the (K, L)-grid in which Ω is divided, $r = 1, \ldots, N$.

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

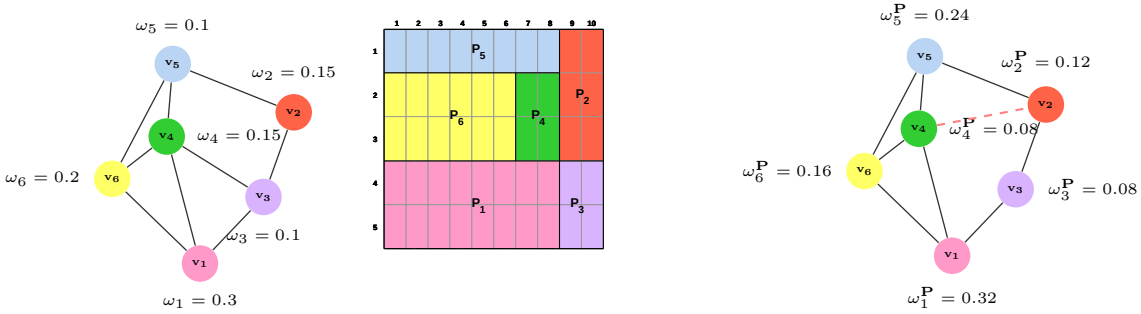
(C4) $\omega_r^{\mathbf{P}} = \omega_r$, namely $\dfrac{1}{K \times L}|P_r| = \omega_r$, where $|P_r|$ denotes the number of cells in P_r, $r = 1, \ldots, N$.

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 1 (c). However, including conditions (C3) and (C4) as hard requirements may make the problem infeasible, [33]. 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

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

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 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 ω to the area of the rectangles in $\omega^{\mathbf{P}}$ weighed by parameter λ_3. Thus, different values of $\boldsymbol{\lambda}$ yield different rectangular maps, highlighting the different aspects involved.

Figure 3 illustrates the concept of (K, L)-rectangular map, using as G the weighted graph plotted in Figure 3 (a), where $N = 6$, $|E| = 9$ and $\omega = (0.3, 0.15, 0.1, 0.15, 0.1, 0.2)$. Figure 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 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 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) $G = (V, E, \omega)$ (b) $(5, 10)$-rectangular map of G (c) $G^{\mathbf{P}} = (V, E^{\mathbf{P}}, \omega^{\mathbf{P}})$: $E \cap E^{\mathbf{P}}$ solid line, $\overline{E} \cap E^{\mathbf{P}}$ dashed line.

Figure 3: A $(5, 10)$-rectangular map for G; $|E \cap E^{\mathbf{P}}| = 8$, $|\overline{E} \cap E^{\mathbf{P}}| = 1$, $\sum_{r=1}^{N} |\omega_r^{\mathbf{P}} - \omega_r| = 0.24$.

3 An MILP formulation for building rectangular maps

In this section we formulate the problem $(RM)_{\boldsymbol{\lambda}}$ as an MILP. We present the decision variables in Section 3.1, the objective function in Section 3.2, and the constraints in Section 3.3. The complete formulation is given in Section 3.4. In what follows, indices r and s are used for portions, i for rows of the grid and j for columns.

3.1 Decision variables

The binary variables which control whether the cell (i, j) belongs to the portion P_r or not are stated as x_{rij} and defined as

$$x_{rij} = \begin{cases} 1 & \text{if cell } (i,j) \text{ belongs to portion } P_r \\ 0 & \text{otherwise.} \end{cases}$$

Thanks to these variables, a portion P_r can be expressed as $P_r = \{(i, j) : x_{rij} = 1, i = 1, \ldots, K, j = 1, \ldots, L\}$.

In order to model adjacencies between portions P_r and P_s, binary variables z_{rs} are defined as

$$z_{rs} = \begin{cases} 1 & \text{if portion } P_s \text{ is adjacent to portion } P_r \\ 0 & \text{otherwise.} \end{cases}$$

Observe that x and z-variables are closely related: if P_r and P_s are two adjacent portions, then $z_{rs} = 1$ and $x_{rij} = x_{si'j'} = 1$, where (i', j') is either equal to $(i-1, j)$ or $(i+1, j)$ or $(i, j+1)$ or $(i, j-1)$.

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

$$u^1_{rsij} = \begin{cases} 1 & \text{if portion } P_s \text{ is adjacent to portion } P_r \text{ at cell } (i,j) \text{ from above} \\ 0 & \text{otherwise.} \end{cases}$$

Similarly, we can define u^2_{rsij}, u^3_{rsij}, and u^4_{rsij}, which indicate if portions P_r and P_s are adjacent from below, to the left or to the right, respectively. Observe that also x and u-variables are closely related, since $u^1_{rsij} = x_{rij} \cdot x_{si-1j}$, $u^2_{rsij} = x_{rij} \cdot x_{si+1j}$, $u^3_{rsij} = x_{rij} \cdot x_{sij+1}$ and $u^4_{rsij} = x_{rij} \cdot x_{sij-1}$.

Finally, φ_r and ψ_r are positive real variables to linearize the area deviation $|\omega_r^{\mathbf{P}} - \omega_r|$, i.e., $|\omega_r^{\mathbf{P}} - \omega_r| = \varphi_r + \psi_r$ and $\omega_r^{\mathbf{P}} - \omega_r = \varphi_r - \psi_r$.

We illustrate these variables using the $(5, 10)$-rectangular map in Figure 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_{4ij}, z_{4s} and u^l_{4sij} are zero. Finally, $\varphi_4 = 0$ and $\psi_4 = 0.07$.

3.2 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,

$$\lambda_1 \sum_{\substack{r,s=1\ldots N \\ (r,s)\in E}} z_{rs} - \lambda_2 \sum_{\substack{r,s=1\ldots N \\ (r,s)\in \overline{E}}} z_{rs} - \lambda_3 \sum_{r=1,\ldots,N} (\varphi_r + \psi_s), \tag{1}$$

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

3.3 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.

$$\sum_{r=1,\ldots,N} x_{rij} = 1, \qquad i = 1,\ldots,K,\ j = 1,\ldots,L, \tag{2}$$

$$\sum_{\substack{i=1,\ldots,K \\ j=1,\ldots,L}} x_{rij} \geq 1, \qquad r = 1,\ldots,N, \tag{3}$$

$$\sum_{\substack{\min\{i,i'\}\leq i''\leq \max\{i,i'\} \\ \min\{j,j'\}\leq j''\leq \max\{j,j'\}}} x_{r\,i''\,j''} \geq (|i-i'|+1)\cdot(|j-j'|+1)\cdot(x_{rij} + x_{ri'j'} - 1), \qquad \begin{array}{l} r = 1,\ldots,N, \\ i, i' = 1,\ldots,K, \\ j, j' = 1,\ldots,L, \end{array} \tag{4}$$

$$\sum_{\substack{i=2,\ldots,K \\ j=1,\ldots,L}} u^1_{rsij} + \sum_{\substack{i=1,\ldots,K-1 \\ j=1,\ldots,L}} u^2_{rsij} + \sum_{\substack{i=1,\ldots,K \\ j=1,\ldots,L-1}} u^3_{rsij} + \sum_{\substack{i=1,\ldots,K \\ j=2,\ldots,L}} u^4_{rsij} \geq z_{rs}, \qquad r, s = 1,\ldots,N,\ r \neq s, \tag{5}$$

$$x_{rij} + x_{s\,i-1\,j} \leq z_{rs} + 1, \qquad r, s = 1,\ldots,N,\ r \neq s, i = 2,\ldots,K,\ j = 1,\ldots,L, \tag{6}$$

$$x_{rij} + x_{s\,i+1\,j} \leq z_{rs} + 1, \qquad r, s = 1,\ldots,N,\ r \neq s, i = 1,\ldots,K-1,\ j = 1,\ldots,L, \tag{7}$$

$$x_{rij} + x_{s\,i\,j+1} \leq z_{rs} + 1, \qquad r, s = 1,\ldots,N,\ r \neq s, i = 1,\ldots,K,\ j = 1,\ldots,L-1, \tag{8}$$

$$x_{rij} + x_{s\,i\,j-1} \leq z_{rs} + 1, \qquad r, s = 1,\ldots,N,\ r \neq s, i = 1,\ldots,K,\ j = 2,\ldots,L, \tag{9}$$

$$u^1_{rsij} \leq x_{rij}, \qquad r, s = 1,\ldots,N,\ r \neq s,\ i = 1,\ldots,K,\ j = 1,\ldots,L, \tag{10}$$

$$u^1_{rsij} \leq x_{s\,i-1\,j}, \qquad r, s = 1,\ldots,N,\ r \neq s,\ i = 2,\ldots,K,\ j = 1,\ldots,L, \tag{11}$$

$$x_{rij} + x_{s\,i-1\,j} \leq u^1_{rsij} + 1, \qquad r, s = 1,\ldots,N,\ r \neq s,\ i = 2,\ldots,K,\ j = 1,\ldots,L, \tag{12}$$

$$u^2_{rsij} \leq x_{rij}, \qquad r, s = 1,\ldots,N,\ r \neq s,\ i = 1,\ldots,K,\ j = 1,\ldots,L, \tag{13}$$

$$u^2_{rsij} \leq x_{s\,i+1\,j}, \qquad r, s = 1,\ldots,N,\ r \neq s,\ i = 1,\ldots,K-1,\ j = 1,\ldots,L, \tag{14}$$

$$x_{rij} + x_{s\,i+1\,j} \leq u^2_{rsij} + 1, \qquad r, s = 1,\ldots,N,\ r \neq s,\ i = 1,\ldots,K-1,\ j = 1,\ldots,L, \tag{15}$$

$$u^3_{rsij} \leq x_{rij}, \qquad r,s=1,\dots,N,\ r\neq s,\ i=1,\dots,K,\ j=1,\dots,L, \tag{16}$$

$$u^3_{rsij} \leq x_{s\,i\,j+1}, \qquad r,s=1,\dots,N,\ r\neq s,\ i=1,\dots,K,\ j=1,\dots,L-1, \tag{17}$$

$$x_{rij} + x_{s\,i\,j+1} \leq u^3_{rsij} + 1, \qquad r,s=1,\dots,N,\ r\neq s,\ i=1,\dots,K,\ j=1,\dots,L-1, \tag{18}$$

$$u^4_{rsij} \leq x_{rij}, \qquad r,s=1,\dots,N,\ r\neq s,\ i=1,\dots,K,\ j=1,\dots,L, \tag{19}$$

$$u^4_{rsij} \leq x_{s\,i\,j-1}, \qquad r,s=1,\dots,N,\ r\neq s,\ i=1,\dots,K,\ j=2,\dots,L, \tag{20}$$

$$x_{rij} + x_{s\,i\,j-1} \leq u^4_{rsij} + 1, \qquad r,s=1,\dots,N,\ r\neq s,\ i=1,\dots,K,\ j=2,\dots,L, \tag{21}$$

$$\frac{1}{KL}\sum_{\substack{i=1,\dots,K\\ j=1,\dots,L}} x_{rij} - \omega_r = \varphi_r - \psi_r, \qquad r=1,\dots,N, \tag{22}$$

$$x_{rij},\ z_{rs},\ u^l_{rsij} \in \{0,1\}, \qquad r,s=1,\dots,N,\ r\neq s,\ i=1,\dots,K,\ j=1,\dots,L,\ l=1,\dots,4, \tag{23}$$

$$\varphi_r, \psi_r \geq 0, \qquad r=1,\dots,N. \tag{24}$$

Firstly, note that condition (C1) is satisfied thanks to the definition of the x-variables and constraint (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 rectangular map, constraint (3) ensures that at least one cell is allocated for every individual. The rectangular-shaped requirement in (C2) is stated by constraint (4), which forces that for every pair the cells (i,j) and (i',j') belonging to the same portion, P_r, all the $(|i-i'|+1)\cdot(|j-j'|+1)$ cells in-between them must belong also to P_r. Constraint (5) models the correctness of $z_{rs}=1$, i.e., if variable z_{rs} takes the value 1, then, there must be two adjacent cells belonging to portions P_r and P_s 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 (5). On the other hand, constraints (6)–(9) model the correctness of $z_{rs}=0$, 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 (10)–(21) model the fact that variables u are the product of two x variables, as noted in Section 3.1, [39]. Constraint (22) ensures the correctness of the absolute value in the area deviation in the objective function. Finally, the variables' type is modeled with constraints (23) and (24).

3.4 Writing the problem as an MILP

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

$$\begin{array}{ll} \max & (1) \\ \text{s.t.} & (2)\text{–}(24). \end{array} \tag{$RML)_{\boldsymbol{\lambda}}$}$$

In a first attempt, we solved $(RML)_{\boldsymbol{\lambda}}$ using a commercial MILP solver under a time limit. In our experimental section, we will illustrate that 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 core, since this MILP formulation, with a few decision variables fixed to a given value, is solved in each iteration.

4 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, see [4], and Cartography, where it is customary to impose that each portion must contain a point, which is usually the centroid of the geographical region, [18, 26, 57].

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 $q_r \in P_r$, $r = 1, \ldots, N$. We define the set of locating cells $\mathcal{C}$ as,

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

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

$$\begin{array}{lll} \max & (1) & \\ \text{s.t.} & (2)\text{–}(24) & \\ & x_{rij} = 1 \qquad & (r,i,j) \in \mathcal{C}. \end{array} \qquad (RML)_{\boldsymbol{\lambda},\mathcal{C}}$$

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 Multidimensional Scaling (MDS), [7], the MultiDimensional Scaling for (K, L)-rectangular maps, which can be applied as long as a dissimilarity measure is given, [11].

4.1 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_r whose areas are close to ω_r and which are related through an adjacency relation. See [1, 31, 37] and references therein for other uses of MDS for planar visualization maps.

Let $D = (d_{rs})$ be the shortest path distance matrix between all nodes of graph $G = (V, E, \omega)$. We want to find N points which lie in Ω, $q_r = (q_r^1, q_r^2)$, contained in N rectangles defined by their NW and SE corners, (a_r^{NW}, b_r^{NW}) and (a_r^{SE}, b_r^{SE}) respectively. These rectangles, called in

what follows MDS rectangles, are surrogate of the rectangular portions P_r 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 in [4] 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 $\|q_r - q_s\|_1$ between locating points q_r and q_s should follow the same pattern than the distance d_{rs}. Hence, we impose that

$$\|q_r - q_s\|_1 \approx \kappa d_{rs}, \tag{25}$$

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. See [28, 35, 58, 59] for other MDS applications using the ℓ_1 norm.

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

$$(a_r^{SE} - a_r^{NW})(b_r^{NW} - b_r^{SE}) \approx \omega_r. \tag{26}$$

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:

$$\max\left\{0, \min\{a_r^{SE}, a_s^{SE}\} - \max\{a_r^{NW}, a_s^{NW}\}\right\} \cdot \max\left\{0, \min\{b_r^{NW}, b_s^{NW}\} - \max\{b_r^{SE}, b_s^{SE}\}\right\} \approx 0. \tag{27}$$

With this notation, the MDS for (K, L)-rectangular maps is stated as the problem of finding rectangles, identified by their corner coordinates (a_r^{NW}, b_r^{NW}) and (a_r^{SE}, b_r^{SE}), and points q_r within minimal violation of soft constraints (25)–(27). This is expressed as the following nonlinear nonsmooth continuous optimization problem:

$$\begin{aligned}
\min\ & \gamma_1 \sum_{r,s=1}^{N} (d_{rs} - \kappa\|q_r - q_s\|_1)^2 + \gamma_2 \sum_{i=r}^{N} \left(\left(a_r^{SE} - a_r^{NW}\right)\left(b_r^{NW} - b_r^{SE}\right) - \omega_r\right)^2 && (28)\\
& + \gamma_3 \max\left\{0, \min\{a_r^{SE}, a_s^{SE}\} - \max\{a_r^{NW}, a_s^{NW}\}\right\} \cdot \max\left\{0, \min\{b_r^{NW}, b_s^{NW}\} - \max\{b_r^{SE}, b_s^{SE}\}\right\}\\
\text{s.t.}\ & \\
& 0 \le a_r^{NW} \le q_r^1 \le a_r^{SE} \le 1, \qquad r = 1, \ldots, N && (29)\\
& 0 \le b_r^{SE} \le q_r^2 \le b_r^{NW} \le 1, \qquad r = 1, \ldots, N && (30)\\
& \kappa > 0, && (31)
\end{aligned}$$

where κ is a scaling variable and γ_1, γ_2, $\gamma_3 \ge 0$ 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 $r \neq s$, where q_r and q_s 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_r and q_s 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

$$q_r \in P_r, \ \forall r \in R, \tag{32}$$

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

4.2 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 built by the MDS framework described in Section 4.1. 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 Figure 4.

Algorithm 1 Cell Perturbing Algorithm (CPA)

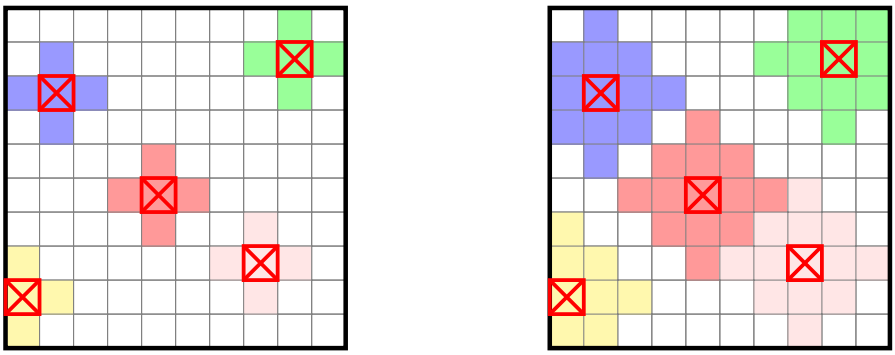
Input: The set of locating cells derived from locating points obtained with MDS for (K, L)-rectangular maps, $\mathcal{C}^{MDS}$, and a perturbing procedure, $perturb(\cdot)$.

1: $\mathcal{C}^0 \leftarrow \mathcal{C}^{MDS}$;
2: Solve $RML_{\boldsymbol{\lambda},\mathcal{C}^0}$;
3: **repeat**
4: $\quad\mathcal{C}^* \leftarrow perturb(\mathcal{C}^0)$;
5: $\quad$Solve $RML_{\boldsymbol{\lambda},\mathcal{C}^*}$;
6: $\quad$**if** $v(\overline{RML_{\boldsymbol{\lambda},\mathcal{C}^*}}) > v(\overline{RML_{\boldsymbol{\lambda},\mathcal{C}^0}})$ **then**
7: $\quad\quad\overline{RML_{\boldsymbol{\lambda},\mathcal{C}^0}} \leftarrow \overline{RML_{\boldsymbol{\lambda},\mathcal{C}^*}}$;
8: $\quad\quad\mathcal{C}^0 \leftarrow \mathcal{C}^*$;
9: $\quad$**end if**
10: **until** stop condition is met

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

Figure 4: Cell Perturbing Algorithm (CPA) pseudocode

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



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

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

of a cell (i,j) as the set of cells formed by those which are at distance lower or equal than ρ, namely

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

Figure 5 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 5 (a) and 5 (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) are allowed, namely there cannot be a locating cell belonging to two rectangles simultaneously. Other more sophisticated 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 the 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.

4.3 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 Figure 6.

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

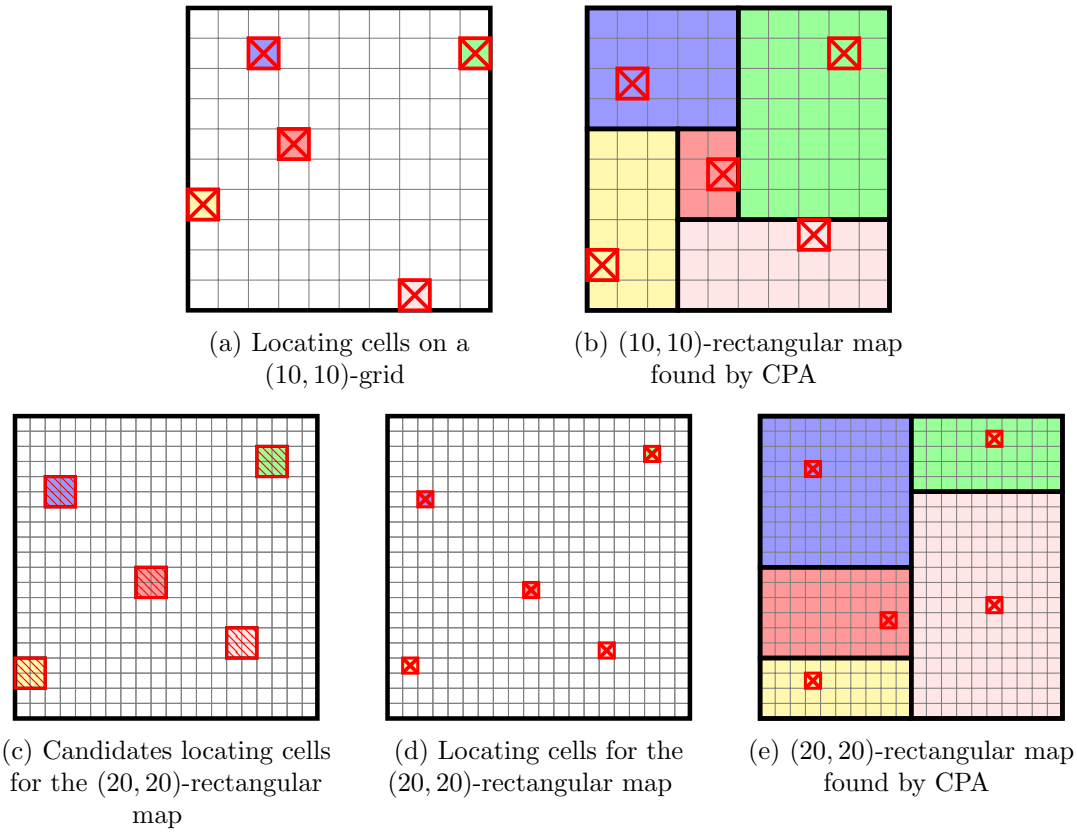
Figure 6: Embedded Cell Perturbing Algorithm pseudocode

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 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 7 illustrates the ECPA algorithm with a $(10, 10)$ and $(20, 20)$-grids and 5 individuals. In Figure 7 (a), the set of 5 locating cells, found via the MDS procedure, are depicted as "$\times$" on a $(10, 10)$-grid. A $(10, 10)$-rectangular map obtained by performing some iterations of CPA is shown in Figure 7 (b). Observe how the locating cells have changed via the $perturb(\cdot)$ procedure in CPA in Figures 7 (a) and 7 (b). In Figure 7 (c), the candidates to become locating cells on a $(20, 20)$-grid are dashed, whereas Figure 7 (d) contains the resulting locating cells from the subdividing procedure. Finally, Figure 7 (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 "$\times$".

5 Experimental results

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 5.1 we describe the three datasets used in the experiments and in Section 5.2 how the ECPA has been implemented. We have claimed that our MILP cannot be solved to optimality by commercial solvers. This is shown in Section 5.3, calling for sophisticated matheuristic approaches. We then discuss the fit of the $(20, 20)$-rectangular maps generated by



(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 7: Illustration of ECPA

ECPA in terms of the adjacency relation and the areas.

5.1 Datasets

The first example, `Blood`, consists of the weighted graph which models the proportion of people in the U.S. in each blood group [50], taking 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_r and v_s are adjacent if either v_r can donate blood to v_s, or viceversa. In the second example, `Netherlands`, the individuals are the provinces of The Netherlands, and the data represented is their (normalized) population, see [51]. 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, see [16].

Figures 8 (a), 9 (a) and 10 (a) show, respectively, the `Blood`, `Netherlands` and `Germany` graphs. The settings of each dataset are included in Table 2 in the Appendix.

5.2 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 ($\sum_{r=1}^{N} |\omega_r - \omega_r^{\mathbf{P}}|$), as stated in conditions (C3) and (C4) in Section 2. Finally, we consider $\boldsymbol{\lambda} = \left(\frac{1}{|E|}, \frac{1}{|\overline{E}|}, 1\right)$.

The locating points are obtained by solving the MDS for rectangular maps given by (28)–(31) 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, [56].

The ECPA has been coded in AMPL, [22], and all the MILPs involved have been solved with CPLEX v12.6, [13], 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 $T = 2$ 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.

In order to demonstrate the need for a sophisticated matheuristic such as ECPA, the quality of our solution approach is tested against the so-called CRUDE heuristic, in which $(RML)_{\boldsymbol{\lambda}}$ is solved by an MILP commercial package using a time limit. In our experimental results, we run CRUDE with a $(20, 20)$-grid using CPLEX with a time limit of 3 hours. Note that our preliminary tests when taking the same embedding as ECPA, with $T = 2$ levels, yielded no solution. Therefore, we have been obliged to start with a coarser grid, and thus use $T = 3$ levels. This means that we solve $(RML)_{\boldsymbol{\lambda}}$ in a $(5, 5)$-grid with a time limit of one hour, its solution is given as starting to the $(10, 10)$-grid with a time limit of one hour, and finally, its solution is given to the $(20, 20)$-grid with a time limit of one hour.

5.3 Results

The performance of CRUDE and ECPA can be found in Table 1 for $\boldsymbol{\lambda} = \left(\frac{1}{|E|}, \frac{1}{|\overline{E}|}, 1\right)$.

Clearly, the results of CRUDE are worse for each criterion, for the three largest datasets. For `Blood`, which is the smallest, the results are worse for the third criterion. In all cases, the overall time is higher in CRUDE. Below, we illustrate that ECPA, although of a heuristic nature, obtains good representations of the considered graphs as rectangular maps. Note that our visualization model is a novel one, and therefore there are no other techniques we can benchmark ECPA against ready available in the literature.

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. The overall time to obtain this solution was approximately two hours. 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 undi-

Table 1: CRUDE and ECPA heuristic approaches.

		$\lvert E \cap E^{\mathbf{P}} \rvert$	$\lvert \overline{E} \cap E^{\mathbf{P}} \rvert$	$\sum_{r=1}^{N} \lvert \omega_r - \omega_r^{\mathbf{P}} \rvert$	Time
`Blood`	CRUDE	17	0	0.468	3 hours
	ECPA	17	0	0.072	2 hours
`Netherlands`	CRUDE	16	5	0.416	3 hours
	ECPA	22	3	0.122	21 minutes
`Germany`	CRUDE	14	14	0.438	3 hours
	ECPA	28	7	0.290	27 minutes

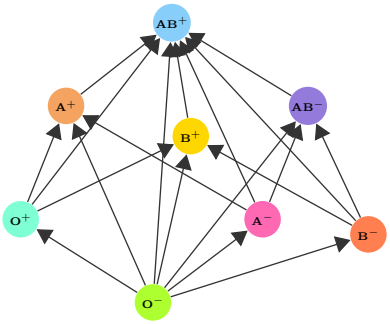
rected). Observe that the relationships between the different groups are well represented through the adjacency relation in the $(20, 20)$-rectangular map, at the same time that the percentage of people belonging to each group is very accurately depicted. Varying the value of $\boldsymbol{\lambda}$ we 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. The so-obtained maps are not depicted in the paper for the sake of abbreviation.

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 deviation of 0.122. The overall time to obtain this solution was approximately 21 minutes. Varying the value of $\boldsymbol{\lambda}$ 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.

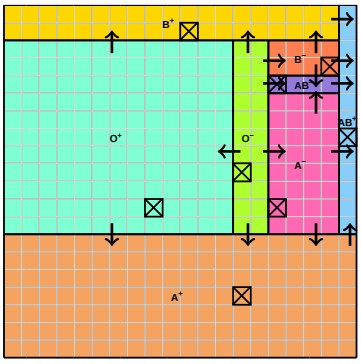
For the `Germany` graph, we obtained a $(20, 20)$-rectangular map in which 28 out of 28 adjacencies are reproduced, 7 false adjacencies are added and with an area deviation of 0.290. The overall time to obtain this solution was approximately 27 minutes. For different values of $\boldsymbol{\lambda}$, 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}.

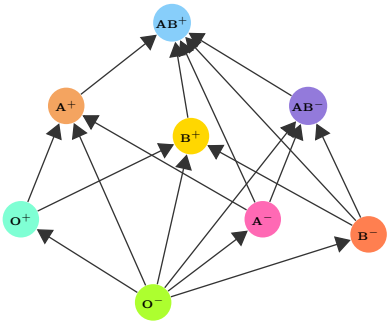
The output of our experimental results for ECPA is presented in Figures 8–10. Figure 8 (a) depicts the `Blood` graph G, Figure 8 (b) the $(20, 20)$-rectangular map obtained as detailed in Section 5.2 with the locating cells marked with a "$\times$", and Figure 8 (c) the graph associated to the $(20, 20)$-rectangular map, $G^{\mathbf{P}}$, 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 9 and 10, respectively.



(a) `Blood` graph, G

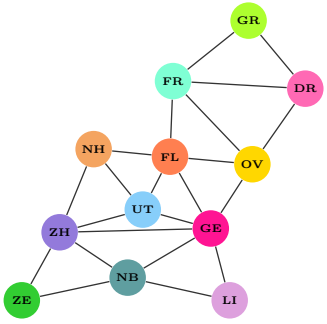


(b) $(20, 20)$-rectangular map

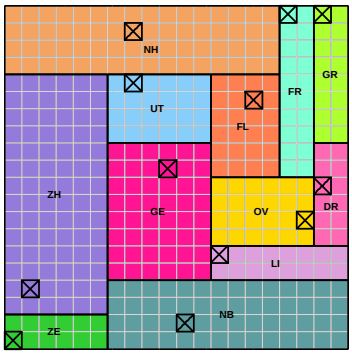


(c) Graph associated with the $(20, 20)$-rectangular map, $G^{\mathbf{P}}$

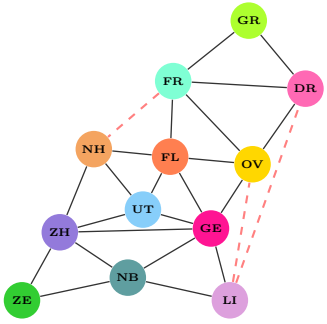
Figure 8: `Blood` $(20, 20)$-rectangular map with $|E \cap E^{\mathbf{P}}| = 17$, $|\overline{E} \cap E^{\mathbf{P}}| = 0$, $\sum_{r=1}^{N} |\omega_r^{\mathbf{P}} - \omega_r| = 0.072$.



(a) `Netherlands` graph

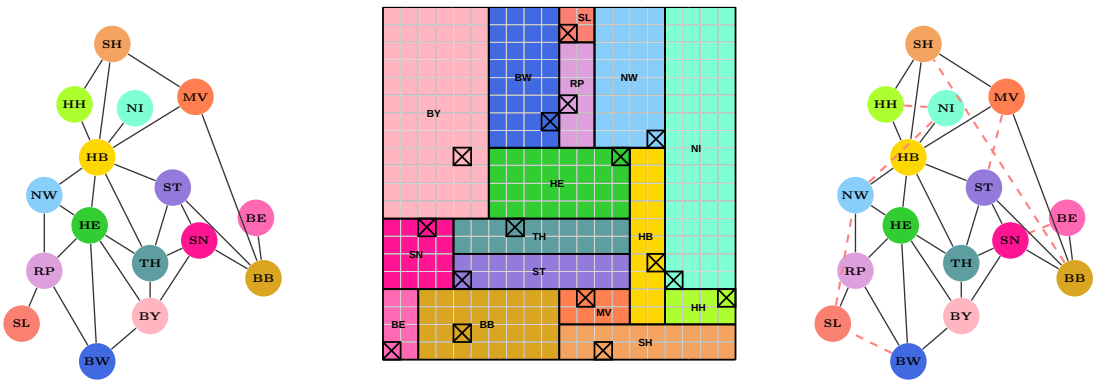


(b) $(20, 20)$-rectangular map



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

Figure 9: `Netherlands` $(20, 20)$-rectangular map with $|E \cap E^{\mathbf{P}}| = 22$, $|\overline{E} \cap E^{\mathbf{P}}| = 3$, $\sum_{r=1}^{N} |\omega_r^{\mathbf{P}} - \omega_r| = 0.122$.



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

Figure 10: `Germany` $(20, 20)$-rectangular map with $|E \cap E^{\mathbf{P}}| = 28$, $|\overline{E} \cap E^{\mathbf{P}}| = 7$, $\sum_{r=1}^{N} |\omega_r^{\mathbf{P}} - \omega_r| = 0.290$.

6 Conclusions and future research

In this paper we have developed a new Mathematical Optimization approach to address the problem of representing by means of rectangular maps a set of individuals, to which frequencies and adjacencies are attached. 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 as an MILP. Due to its hard combinatorial structure, a tailored MultiDimensional Scaling 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.

There are several interesting lines for future research, mainly based on the study of other applications of Mathematical Optimization to visualization frameworks. First, ECPA could be embedded into a metaheuristic such as Variable Neighborhood Search, [40], to speed up the procedure. Second, the so-called "segment moving heuristic", [33], could be customized to our problem in order to improve the approximation made in the areas after having a (K, L)-rectangular map. Nevertheless, even if we were able to detect the rectangles whose sizes can be changed, and thus, the segments that can be moved without destroying the rectangular shapes, the adjacencies structure could be dramatically changed by such movements. Hence, local changes are difficult to detect due to the rigid structure of the map and this approach deserves further study. Third, we are studying the problem of representing each node of the graph G by a connected union of grid cells, not necessarily with a rectangular shape [14, 15]. Having less rigid shapes than rectangles has two advantages, namely, the proximities between individuals can be represented more accurately, while better results in terms of area deviations can be

achieved. Fourth, we would like to customize the technique of representing a set of individuals with attached frequencies and proximities as a rectangular map to detect communities in graphs, [21], by analyzing the adjacencies represented in the rectangular map. Finally, our method can also be applied to visualize hierarchical data, in which inside every rectangle a new rectangular map has to be represented by taking into account adjacencies with neighboring rectangles and its inner rectangular maps, [12, 27, 48]. However, the mathematical optimization treatment of these extensions is not trivial and thus further research is still needed.

Acknowledgement. *We thank the reviewers for their thorough comments and suggestions, which have been very valuable to strengthen the quality of the paper. This research is funded in part by Projects MTM2015-65915-R (Spain), P11-FQM-7603 and FQM-329 (Andalucía), all with EU ERD Funds.*

References

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

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

[3] M. F. Anjos and F. Liers. Global approaches for facility layout and VLSI floorplanning. In *Handbook on Semidefinite, Conic and Polynomial Optimization*, pages 849–877. Springer, 2012.

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

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

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

[7] I. Borg and P. J. F. Groenen. *Modern Multidimensional Scaling: Theory and Applications*. Springer, 2005.

[8] K. Buchin, B. Speckmann, and S. Verdonschot. Evolution strategies for optimizing rectangular cartograms. In *Geographic Information Science*, pages 29–42. Springer, 2012.

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

[10] E. Carrizosa, V. Guerrero, and D. Romero Morales. Visualizing proportions and dissimilarities by space-filling maps: a large neighborhood search approach. *Computers & Operations Research*, 78:369–380, 2017.

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

[12] S. Clémençon, H. De Arazoza, F. Rossi, and V. C. Tran. Hierarchical clustering for graph visualization. In *19th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, pages 227–232, 2011.

[13] IBM ILOG CPLEX. http://www.ilog.com/products/cplex/, 2014.

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

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

[16] Destatis, Statistisches Bundesamt. Area and population. www.destatis.de, 2015. Retrieved on: 2015-01-14.

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

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

[19] D. Eppstein, E. Mumford, B. Speckmann, and K. Verbeek. Area-universal rectangular layouts. In *Proceedings of the Twenty-Fifth Annual Symposium on Computational Geometry*, pages 267–276. ACM, 2009.

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

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

[22] R. Fourer, D. M. Gay, and B. W. Kernighan. *AMPL: A Modeling Language for Mathematical Programming*. Duxbury Press, Brooks/Cole Publishing Company, Stamford, 2002.

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

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

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

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

[27] I. Herman, G. Melançon, and M. S. Marshall. Graph visualization and navigation in information visualization: A survey. *IEEE Transactions on Visualization and Computer Graphics*, 6(1):24–43, 2000.

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

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

[30] D. Keim, G. Andrienko, J. D. Fekete, C. Görg, J. Kohlhammer, and G. Melançon. Visual analytics: Definition, process, and challenges. In *Information Visualization*, pages 154–175. Springer, 2008.

[31] M. Klimenta and U. Brandes. Graph drawing by classical multidimensional scaling: new perspectives. In *Graph Drawing*, volume 7704, pages 55–66. Springer, 2013.

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

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

[34] J. B. Kruskal and M. Wish. *Multidimensional Scaling*, volume 11. Sage, 1978.

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

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

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

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

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

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

[41] M. J. Mortenson, N. F. Doherty, and S. Robinson. Operational research from taylorism to terabytes: A research agenda for the analytics age. *European Journal of Operational Research*, 241(3):583–595, 2015.

[42] S. Nusrat and S. Kobourov. The state of the art in cartograms. *Computer Graphics Forum*, 35(3):619–642, 2016.

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

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

[45] C. Panse. Rectangular Statistical Cartograms in R: The recmap Package. *arXiv preprint arXiv:1606.00464*, 2016.

[46] E. Raisz. The rectangular statistical cartogram. *Geographical Review*, pages 292–296, 1934.

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

[48] B. Shneiderman and C. Dunne. Interactive network exploration to derive insights: Filtering, clustering, grouping, and simplification. In *Graph Drawing*, volume 7704, pages 2–18. Springer, 2013.

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

[50] Stanford Blood Center. Blood type in U.S. http://bloodcenter.stanford.edu/, 2014. Retrieved on: 2014-11-19.

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

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

[53] R. Tamassia, editor. *Handbook of Graph Drawing and Visualization*. CRC press, 2013.

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

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

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

[57] J. Wood and J. Dykes. Spatially ordered treemaps. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1348–1355, 2008.

[58] A. Žilinskas and J. Žilinskas. Branch and bound algorithm for multidimensional scaling with city-block metric. *Journal of Global Optimization*, 43(2):357–372, 2009.

[59] J. Žilinskas. Parallel branch and bound for multidimensional scaling with city-block distances. *Journal of Global Optimization*, 54(2):261–274, 2012.

Appendix

Table 2 contains, for each dataset used in Section 5: The number of individuals N, the number of edges and its complement, the label of each individual, their full name in `Netherlands` and `Germany` cases and the frequencies ω.

Table 2: Graphs settings.

`Blood`			`Netherlands`			`Germany`		
N	$\lvert E\rvert$	$\lvert\overline{E}\rvert$	N	$\lvert E\rvert$	$\lvert\overline{E}\rvert$	N	$\lvert E\rvert$	$\lvert\overline{E}\rvert$
8	19	9	12	22	44	16	28	92
ω			ω			ω		
O^-		0.066	GR	Groningen	0.035	HH	Hamburg	0.0021
O^+		0.374	FR	Friesland	0.038	NI	Lower Saxony	0.1334
A^-		0.063	DR	Drenthe	0.029	BE	Berlin	0.0025
A^+		0.357	NH	Noord Holland	0.163	SH	Schleswig-Holstein	0.0441
B^-		0.015	FL	Flevoland	0.024	MV	Mecklenburg-Vorpommern	0.0649
B^+		0.085	OV	Overijssel	0.068	HB	Bremen	0.0011
AB^-		0.006	ZH	Zuid Holland	0.212	ST	Saxony-Anhalt	0.0573
AB^+		0.034	UT	Utrecht	0.074	NW	North Rhine-Westphalia	0.0955
			GE	Gelderland	0.120	SN	Saxony	0.0516
			ZE	Zeeland	0.023	HE	Hesse	0.0591
			NB	Noord Brabant	0.147	TH	Thuringia	0.0453
			LI	Limburg	0.067	RP	Rhineland-Palatinate	0.0556
						BY	Bavaria	0.1976
						BW	Baden-Württemberg	0.1001
						SL	Saarland	0.0072
						BB	Brandenburg	0.0826