technische universiteit eindhoven
Department of Mathematics and Computer Science

Master's Thesis

Optimizing Regular Edge Labelings

by
Sander Verdonschot

Supervisors
dr. K.A. Buchin
dr. B. Speckmann

Eindhoven, October 28, 2010

Abstract

A regular edge labeling of an irreducible triangulation G uniquely defines a rectangular dual of G. Rectangular duals find applications in various areas: as floor plans of electronic chips, in architectural designs, as rectangular cartograms, or as treemaps. An irreducible triangulation can have many regular edge labelings and hence many rectangular duals. Depending on the specific application different duals might be desirable. In this thesis we consider optimization problems on regular edge labelings and show how to find optimal or near-optimal ones for various quality criteria. We evaluate our optimization methods by applying them to generate high quality rectangular cartograms. Furthermore, we show how to efficiently enumerate the regular edge labelings of an irreducible triangulation. Since the running time of the enumeration algorithm depends on the number of regular edge labelings, we also consider the maximal number of regular edge labelings an irreducible triangulation can have. We show that every irreducible triangulation with n vertices has less than $O(4.6807^n)$ regular edge labelings and that there are irreducible triangulations with $\Omega(3.0426^n)$ regular edge labelings.

Contents

Chapter 1

Introduction

Rectangular cartograms were introduced in 1937 by Raisz [23] as a new kind of thematic maps. They represent each region (a country or state for example) as a rectangle, whose area corresponds to a geographic variable like population or gross domestic product. As an example, a rectangular cartogram depicting the area of the provinces of the Netherlands is shown to the right. These cartograms are usually constructed by hand, although recent work by van Kreveld and Speckmann [25] has automated a large part of the construction process. One step that has not been automated yet is finding the right layout, which consists of determining for each pair of adjacent regions whether they should share a horizontal or vertical boundary. The number of options for this assignment is surprisingly large, with maps with as little as 60 regions, such as Europe or the United States, having billions of possibilities. This makes it very hard to find a good layout by hand.

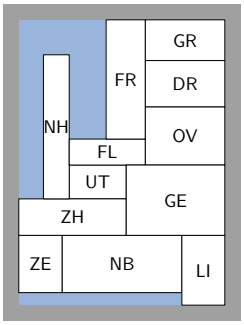


Figure 1.1: An area cartogram of the Netherlands.

In this thesis, we provide algorithms to find optimal or near-optimal layouts and give bounds on the number of layouts one map can have. We evaluate our algorithms by using them to generate high quality rectangular cartograms.

1.1 Rectangular partitions

A *rectangular partition* of a rectangle R is a partition of R into a set $\mathcal{R}$ of non-overlapping rectangles such that no four rectangles in $\mathcal{R}$ meet at one common point. A *rectangular dual* of a plane graph G is a rectangular partition $\mathcal{R}$, such that (i) there is a one-to-one correspondence

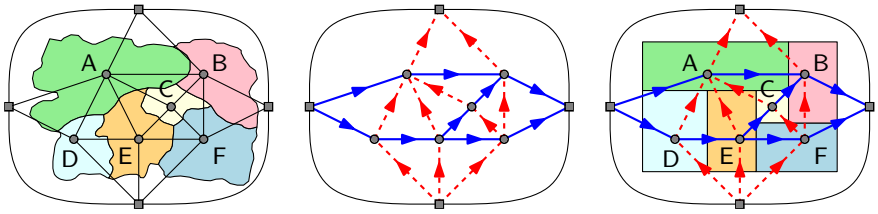


Figure 1.2: A subdivision and its augmented dual graph G, a regular edge labeling of G, and a corresponding rectangular dual.

between the rectangles in $\mathcal{R}$ and the nodes in G; (ii) two rectangles in $\mathcal{R}$ share a common boundary if and only if the corresponding nodes in G are connected. Rectangular duals find applications in various areas: from floor plans of electronic chips or architectural designs, to rectangular cartograms and treemaps.

Not every plane graph has a rectangular dual. A plane graph G has a rectangular dual $\mathcal{R}$ with four rectangles on the boundary of $\mathcal{R}$ if G is an *irreducible triangulation*: (i) G is triangulated and the exterior face is a quadrangle; (ii) G has no separating triangles (a 3-cycle with vertices both inside and outside the cycle) [6, 20]. A plane triangulated graph G has a rectangular dual if and only if we can augment G with four external vertices such that the augmented graph is an irreducible triangulation.

The equivalence classes of the rectangular duals of an irreducible triangulation G correspond one-to-one to the *regular edge labelings* of G. A regular edge labeling of an irreducible triangulation G is a partition of the interior edges of G into two subsets of red and blue directed edges such that: (i) around each inner vertex in clockwise order we have four contiguous sets of incoming blue edges, outgoing red edges, outgoing blue edges, and incoming red edges; (ii) the left exterior vertex has only outgoing blue edges, the top exterior vertex has only incoming red edges, the right exterior vertex has only incoming blue edges, and the bottom exterior vertex has only outgoing red edges (see Fig. 1.1, red edges are dashed). Kant and He [18] show how to find a regular edge labeling and construct the corresponding rectangular dual in linear time. Regular edge labelings are also studied by Fusy [16] who calls them *transversal pairs of bipolar orientations*. The process from subdivision to rectangular dual is illustrated in Fig. 1.2.

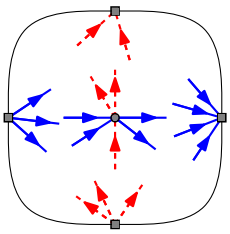


Figure 1.3: The local conditions on a regular edge labeling.

An irreducible triangulation can have many regular edge labelings and hence many rectangular duals. Depending on the specific application different duals might be desirable. For example, *sliceable* duals—which can be obtained by recursively slicing a rectangle by horizontal and vertical lines—are popular in VLSI design. Not every irreducible triangulation has a sliceable dual. A full characterization of the graphs that do is lacking, but Yeap and Sarrafzadeh [26] prove that irreducible triangulations without separating 4-cycles have a sliceable dual. *Area-universal* duals have the nice property that any assignment of areas to rectangles

Figure 1.4: A simplified map of Europe with two different rectangular duals of its dual graph. Luxembourg and Moldavia have been removed and "sea regions" have been added to ensure that the dual graph is an irreducible triangulation.

can be realized by a combinatorially equivalent rectangular dual. Again, not every irreducible triangulation has an area-universal dual, but Eppstein *et al.* [13] show how to find such a dual if it exists.

We are particularly interested in the application of rectangular duals to rectangular cartograms. In this context it is desirable that the direction of adjacency between the rectangles of the dual follows the spatial relation of the regions of the underlying map as closely as possible. Consider the two rectangular duals of the dual graph of a map of Europe shown in Fig. 1.4. The left dual will lead to a recognizable cartogram, whereas the right dual (with France east of Germany and Hungary north of Austria) is useless as basis for a cartogram. Both rectangular duals stem from the same graph G and correspond to two different valid regular edge labelings of G.

Previous work on finding regular edge labelings that lead to cartograms with geographically suitable adjacency directions has focused on regular edge labelings that satisfy *user-specified* constraints on a subset of the edges of the input graph. Eppstein and Mumford [12] show how to find regular edge labelings that satisfy user-specified orientation constraints, if such labelings exist for the given set of constraints. Van Kreveld and Speckmann [25] search through a user-specified subset of the regular edge labelings. Every labeling in this subset is considered acceptable with respect to adjacency directions. In contrast, we consider quality measures that take all edges of G into account and do not concentrate on a fixed, user-specified subset.

1.2 Results and organization

In this thesis we consider optimization problems on regular edge labelings and show how to find optimal or near-optimal ones for various quality criteria. We evaluate our optimization methods by applying them to generate high quality rectangular cartograms. We also give upper and lower bounds on the number of regular edge labelings.

We first recap some previous results and matrix theory in Chapter 2. In Chapter 3, we show how to use reverse search to enumerate all regular edge labelings of an irreducible triangulation G and thus find optimal regular edge labelings for any given quality measure. Since the running time and hence the feasibility of our enumeration algorithm depends on the number of regular edge labelings, we bound the number of regular edge labelings G can have in Chapter 4. The number of regular edge labelings of G can be exponential in the number of vertices n; we first show simple upper and lower bounds of 8^n and $2^{n-O(\sqrt{n})}$. We then prove much tighter bounds, showing that G has less than $O(4.6807^n)$ regular edge labelings and that there are irreducible triangulations with $\Omega(3.0426^n)$ regular edge labelings.

In Chapter 5 we show how to find optimal or near-optimal regular edge labelings for rectangular cartogram construction. This step of the construction pipeline has been performed essentially manually in previous work and can now finally be fully automated. We consider two general quality criteria: (*i*) the relative position of the rectangles, which we compute in three different ways and (*ii*) the *cartographic error* of the resulting cartogram. Our optimization focuses on finding labelings that perform well on both aspects. For smaller maps (the provinces of the Netherlands) enumeration of all regular edge labelings is feasible and we hence can find optimal solutions. For larger maps (the countries of Europe or the contiguous states of the US) enumeration is infeasible. Fortunately the diameter of the distributive lattice of regular edge labelings is comparatively small and hence simulated annealing (described

in Chapter 3) performs well. We present extensive experimental results that show that our method can find regular edge labelings which result in visually pleasing and recognizable cartograms with small cartographic error.

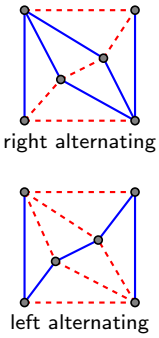
Chapter 2

Preliminaries

In this chapter, we summarize some earlier findings on regular edge labelings and matrix theory that we use in the remainder of the thesis.

2.1 Regular edge labelings

We use several definitions and properties of regular edge labelings that were introduced or proven by Fusy [16]. A *regular edge coloring* is a regular edge labeling, with the directions of the edges omitted. An *alternating 4-cycle* is an undirected 4-cycle in which the colors of the edges alternate between red and blue. There are two kinds of alternating 4-cycles, depending on the color of the interior edges incident to the cycle. If these are the same color as the next clockwise cycle edge, the cycle is *right alternating*, otherwise it is *left alternating*.



Lemma 1 (Fusy [16], Proposition 2, Lemma 1 and Theorem 2)

(a) A regular edge coloring uniquely determines a regular edge labeling.

(b) A regular edge labeling (of an irreducible triangulation) induces no monochromatic triangles.

(c) The set of regular edge labelings of a fixed irreducible triangulation is a distributive lattice. The flip operation consists of switching the edge-colors inside a right alternating 4-cycle and updating the directions, turning it into a left alternating 4-cycle.

A *lattice* is a partially ordered set in which each pair of elements has a unique least upper bound (also called the supremum) and greatest lower bound (called the infimum). Given two elements a and b of the lattice, the *join* operation, written as $a \vee b$, gives the supremum of a and b, while the *meet* operation, written as $a \wedge b$, gives the infimum of a and b. A *distributive lattice* is a lattice in which the join and meet operations distribute over each other: $a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c)$ and $a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c)$. The most common example of a distributive lattice is formed by the subsets of a finite set, with one subset being below another if the first is a subset of the second. In this case, join and meet correspond to union and intersection, respectively. The distributive lattice formed by all regular edge labelings of a small graph is shown in Fig. 2.1.

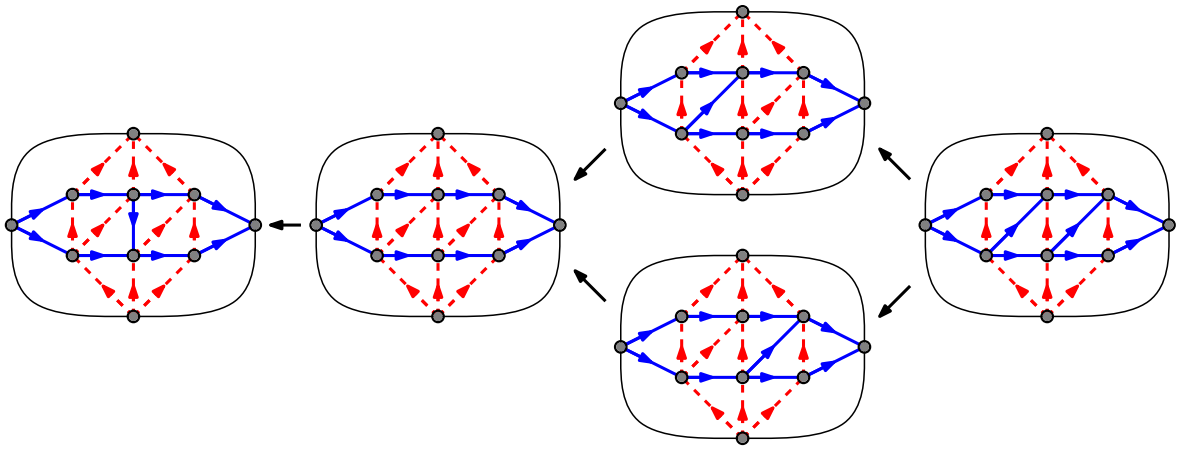


Figure 2.1: The distributive lattice formed by all regular edge labelings of a simple graph. If two labelings are connected by an arrow, switching the colors of all edges inside a right alternating 4-cycle of the originating labeling results in the destination labeling.

Its existence has several consequences. First of all, there is a unique minimal and maximal labeling, respectively the regular edge labeling with no right alternating 4-cycles and the labeling with no left alternating 4-cycles. Any labeling can be transformed into the minimal labeling by repeatedly applying the flip operation until there are no right alternating 4-cycles left. This can be seen as moving "down" the lattice. Further, it allows us to use Birkhoff's representation theorem [7], which states that any element of a finite distributive lattice can be represented by a downward-closed subset of the join-irreducible elements of the lattice. A subset is *downward-closed* if for any element in the subset, all smaller elements are also included. An element is *join-irreducible* if it is neither the minimal element, nor the join of two smaller elements. A regular edge labeling is join-irreducible if (and only if) it has exactly one right alternating 4-cycle. Thus, the theorem allows us to represent each labeling by a downward-closed subset of labelings that only have a single right alternating 4-cycle. Recently, Eppstein and Mumford [12] used this theorem to find regular edge labelings that satisfy a set of user-specified orientation constraints.

A move up the lattice now corresponds to adding a single join-irreducible labeling to the representing subset. This implies that all upward paths between two given elements of the lattice have the same length, as this length equals the difference in size between the subsets representing the two labelings. This, in turn, implies that the *diameter* of the graph (the maximum length of any shortest path between two vertices) equals the distance between the minimal and the maximal labeling. Moreover, to find this distance, we only have to find a single path between them. This can easily be done by starting at the maximal labeling and counting the number of times we need to apply the flip operation to end up at the minimal labeling.

2.2 Perron-Frobenius theory

In the following we summarize the matrix theory needed for our our lower bound computation in Chapter 4. For details we refer to textbooks on matrices [17, 21].

We assume from now on that A is a nonnegative $n \times n$ matrix. A matrix is *nonnegative* if all its elements are nonnegative. The matrix A is *irreducible* if for each (i, j) there is a $k > 0$ such that $(A^k)_{ij} > 0$. Consider the directed graph with adjacency matrix A, where we interpret every non-zero element as an adjacency. The matrix A is irreducible if and only if the associated graph is strongly connected. The matrix A is *primitive* if there is a $k > 0$ such that all elements of A^k are positive. An irreducible matrix with a positive diagonal entry is primitive.

Theorem 2 ([17, 21]) *Let A be a primitive non-negative matrix with maximal eigenvalue λ.*

(a) The eigenvalue λ is positive and it is the unique eigenvalue of largest absolute value. It has a positive eigenvector, and it is the only eigenvalue with nonnegative eigenvector.

(b) Let $f_A(x) = \min_{x_i \neq 0} \frac{(Ax)_i}{x_i}$ and $g_A(x) = \max_{x_i \neq 0} \frac{(Ax)_i}{x_i}$. Then $f_A(x) \leqslant \lambda$ for all nonnegative non-zero vectors x, and $g_A(x) \geqslant \lambda$ for all positive vectors x. If $f_A(x_0) = \lambda$ then x_0 is an eigenvector of A corresponding to λ.

(c) Let x be a nonnegative non-zero vector. Then $A^t x / \|A^t x\|$ converges to an eigenvector with eigenvalue λ. Consequently,

$$\lim_{t \to \infty} f_A(A^t x) = \lim_{t \to \infty} g_A(A^t x) = \lambda.$$

Chapter 3

Traversing the lattice

Many optimization problems can be solved using a linear search through all candidate solutions. The advantages of this method are that it is very straight-forward and guaranteed to find an optimal solution. For this purpose, we present an algorithm to enumerate all labelings of a given graph that is based on the reverse search algorithm developed by Avis and Fukuda [5]. The largest disadvantage of a linear search is that there can be a huge number of candidate solutions, making the running time prohibitively high. To avoid this, other optimization techniques have been developed that aim to reduce the time required while still finding a good solution by directing the search towards promising solutions. We describe how to apply one such technique, simulated annealing [19], to regular edge labelings. Both algorithms search by traversing the distributive lattice formed by all regular edge labelings of a graph.

3.1 Reverse search

Reverse search, proposed by Avis and Fukuda [5], is a general method for enumerating structures that fulfill two criteria: (*i*) there must be a concept of "neighboring" structures such that the structures form a graph; (*ii*) there must be a local search operation that moves through this graph in a deterministic way and ends up at a local optimum. The local search defines a forest on the graph, of which each tree is rooted at a local optimum. If the local optima are known and we have a way of enumerating all neighbors of a structure, then we can traverse these trees by starting at a local optimum and testing for each neighbor if the local search ends up at our current structure when applied to that neighbor. If it does, we traverse the edge and recurse.

Regular edge labelings match these criteria: the distributive lattice provides a natural underlying graph structure, while the flip operation is a good choice for our local search, since we know that we will reach the minimal labeling if we apply it often enough. This provides an excellent starting point for our enumeration. There is one last detail we need to take care of: the local search needs to be deterministic, but a labeling can have multiple right alternating 4-cycles. We can fix this by imposing an ordering on the 4-cycles of the graph and taking the first one. One possible ordering is to sort the vertices lexicographically by their x- and y-coordinates, using this ordering to sort the edges lexicographically by their lower endpoint and higher endpoint and finally using this order on the edges to sort the cycles lexicographically by their lowest edge and the non-adjacent edge.

Avis and Fukuda give an implementation of their algorithm if an upper bound δ on the number of neighbors a structure can have is known and the graph is given by an *adjacency oracle*. Given an integer $0 \leqslant k < \delta$, this adjacency oracle should return the k-th neighbor of the current structure, or $\perp$ if that neighbor does not exist. The algorithm then performs a depth-first search through the graph, by checking for each $0 \leqslant k < \delta$ if that neighbor exists and if it does, if the local search traverses this edge. After handling all neighbors of this neighbor, it returns to the previous structure by using the local search on the neighbor and finding the right value for k where it left off, which is the k for which the adjacency oracle returns the neighbor we just handled.

In order to use this implementation, we need an upper bound δ on the number of neighbors a labeling can have. In our case, we can use the number of 4-cycles in the graph for this. Using our ordering on the 4-cycles of the input graph, we let the oracle return the resulting labeling after flipping the colors of all edges inside the k-th 4-cycle, or $\perp$ if this 4-cycle is not alternating.

The running time of the enumeration is $O(\delta t(\text{oracle})\Lambda + t(\text{local search})\delta\Lambda)$, where Λ is the number of regular edge labelings of the graph. In our case $\delta = O(n^2)$, since a 4-cycle can be defined by its two non-adjacent edges and there are a linear number of edges. The oracle takes linear time, as it might have to switch the color of linearly many edges and the local search takes quadratic time, as it might have to evaluate all 4-cycles to find the first right-alternating one. This gives a total running time of $O(n^4\Lambda)$. If the 4-cycles are precomputed and stored, the algorithm requires $O(n^2)$ space. Finding the 4-cycles on the fly would reduce the space requirement to $O(n)$, but would significantly increase the running time in practice.

3.2 Simulated annealing

Simulated annealing [19] is a well-known global optimization technique for discrete problems with an underlying graph structure. It starts as a random walk through the graph and increasingly favors moving to better solutions, ending up as a greedy hill-climber. This gradually changing behavior is caused by a control parameter, called the *temperature*, which starts out high and decreases with time. Most versions always accept moves to better solutions, but use an *acceptance probability function* to determine if they should move to a worse solution or not. This probability depends on two factors: the difference in quality between the current solution and the one under consideration, and the current temperature. The probability of moving to a worse solution decreases as the quality difference becomes bigger, so it favors solutions that are close in quality to the current solution. The higher the temperature, the higher the probability that a bad move is taken.

There are many good choices for both the temperature and acceptance probability functions. We use a typical static cooling schedule and acceptance probability function [1] for our experiments. Specifically, given two labelings with qualities q_1 and q_2, the probability that our algorithm moves to the worse labeling is $e^{|q_1-q_2|/T}$, where T is the current temperature. We assume that all qualities lie between 0 and 1. We let the temperature decrease exponentially as $T = 0.002^t$, where t is the current time, varying from 0 initially to 1 at the end of the process. The base factor of 0.002 can be increased to produce more random behavior, or decreased to produce more greedy behavior. During the process, we maintain the best solution found so far, which is returned at the end.

Chapter 4

Counting regular edge labelings

Here we prove that every irreducible triangulation with n vertices has less than $O(4.6807^n)$ regular edge labelings and that there are irreducible triangulations with $\Omega(3.0426^n)$ regular edge labelings. Before we present our bounds, we review some additional related work.

A *bipolar orientation* of a graph is an acyclic orientation of the edges such that there is exactly one vertex without any incoming edges (called the *source*) and exactly one without any outgoing edges (called the *sink*). A regular edge labeling gives two bipolar orientations, one on the red edges and one on the blue edges. One might therefore expect that the number of regular edge labelings is related to the number of bipolar orientations, which Felsner and Zickfeld [14] showed to be in $O(3.97^n)$, while also showing that there are planar graphs with $\Omega(2.91^n)$ bipolar orientations. Any regular edge labeling can be turned into a bipolar orientation by adding a source and a sink vertex and connecting the new source to the red and the blue sources and connecting the red and the blue sinks to the new sink. However, in this way many regular edge labelings can be mapped to the same bipolar orientation. Conversely, although Kant and He [18] developed an algorithm that produces a regular edge labeling from the directions of the edges, not every bipolar orientation can be turned into a regular edge labeling this way. This is caused by the fact that bipolar orientations only require each (non-source and non-sink) vertex to have an in- and outdegree of at least one, whereas regular edge labelings require an in- and outdegree of at least two, one blue and one red.

Counting all regular edge labelings of all n-vertex irreducible triangulations yields the number of combinatorially different rectangular partitions with n rectangles which is in $\Omega(11.56^n)$ [4] and less than or equal to 13.5^{n-1} [15]. If we consider partitions to be identical when the incidence structure between rectangles and maximal line segments is the same, then the number of different partitions is in $\Theta(8^n/n^4)$ [2].

Many other interesting substructures have been counted in planar graphs. Aichholzer *et al.* [3] list the known upper bounds for other subgraphs contained in a triangulation: Hamiltonian cycles, cycles, perfect matchings, spanning trees, connected graphs and so on. Several of these bounds have been improved recently [9, 10]. Some of the techniques used to count other substructures can be used to count regular edge labelings, but the upper bounds we obtained by applying these techniques were far from the bound we will present in this thesis. Some of these bounds are discussed at the end of this section.

Our upper bound relies on Shearer's entropy lemma [11]. Björklund *et al.* [8] recently used this lemma to obtain $(2-\varepsilon)^n$ algorithms for the TSP problem. In contrast to our application of the lemma, they count vertex sets with certain properties and crucially rely on bounded maximum degree.

4.1 Upper bound

Let $G = (V, E)$ be an irreducible triangulation on n vertices. By Lemma 1(a), a regular edge labeling is uniquely determined by the colors (red and blue) of the edges. Thus, the number of regular edge labelings of G is bounded by the number of edge colorings with two colors. Since G has less than $3n$ edges by Euler's formula, a simple upper bound is $2^{3n} = 8^n$. Most of the colorings that we obtain by coloring the edges independently red or blue do not correspond to a valid regular edge labeling. In the following we refine our bound using Shearer's entropy lemma.

Lemma 3 (Shearer's entropy lemma [11]) *Let S be a finite set and let $A_1, \ldots, A_m$ be subsets of S such that every element of S is contained in at least k of the $A_1, \ldots, A_m$. Let $\mathcal{F}$ be a collection of subsets of S and let $\mathcal{F}_i = \{F \cap A_i : F \in \mathcal{F}\}$ for $1 \leqslant i \leqslant m$. Then we have*

$$|\mathcal{F}|^k \leqslant \prod_{i=1}^{m} |\mathcal{F}_i|.$$

Shearer's entropy lemma allows us to use the local conditions on regular edge colorings to bound the number of regular edge labelings.

Theorem 4 *The number of regular edge labelings of an irreducible triangulations is in $O(4.6807^n)$.*

Proof. Let $G = (V, E)$ be an irreducible triangulation on n vertices. Let S be E with the four edges on the exterior face excluded. For a regular edge labeling $\mathcal{L}$ of G let $E(\mathcal{L})$ be the set of blue edges in $\mathcal{L}$. Let

$$\mathcal{F} := \{E(\mathcal{L}) \mid \mathcal{L} \text{ is a regular edge labeling of } G\}.$$

Since $E(\mathcal{L})$ determines $\mathcal{L}$, the number of regular edge labelings is $|\mathcal{F}|$.

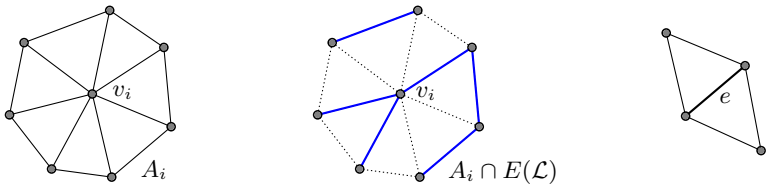


Figure 4.1: A vertex v_i with the corresponding set of edges A_i, a locally consistent choice of blue edges, and an edge e with the four vertices that include e in their A_i.

For the vertices v_i of G ($i = 1, 2, \ldots, n$) let A_i be the set of edges in S of the triangles adjacent to v_i (see Figure 4.1). Every edge $e \in S$ is in four of the sets A_i, namely in the four sets corresponding to the vertices of the two triangles with e as edge. Let F_i be the set of intersections of the set A_i with the sets $E(\mathcal{L})$, i.e., F_i contains all possible ways to choose blue edges around v_i consistent with a regular edge labeling. By Lemma 3 the number of regular edge labelings is bounded by $\prod_{i=1}^{n} |F_i|^{1/4}$.

For a vertex v_i on the outer face there is only one way to choose the colors for the edges in A_i, since the adjacent edges must all have the same color. Since by Lemma 1(b) a regular edge labeling has no monochromatic triangle, the remaining edges in A_i need to be of the

other color. Now let v_i be a vertex that is not on the outer face. We first bound the number of ways in which the edges adjacent to v_i can be colored. We color these edges starting with an arbitrary one and going clockwise from there. For the first edge, we have at most 2 choices and moving clockwise, we need to switch colors exactly four times by the local conditions on regular edge colorings. Therefore the number of choices for the edges adjacent to v_i is bounded by $2\binom{d_i}{4}$, where d_i is the degree of v_i. After coloring the adjacent edges, every triangle in A_i already has two colored edges. We have no choice for the third edge if these two edges have the same color, so we only have a choice for the (at most) four places where we switched colors. Therefore, the number of regular edge labelings of G is bounded by

$$\prod_{i=1}^{n}\left(2^5\binom{d_i}{4}\right)^{1/4} = \left(32^n\prod_{i=1}^{n}\binom{d_i}{4}\right)^{1/4}.$$

Jensen's inequality states that given a concave function f and a set of k values x_i in its domain, $\sum f(x_i) \leqslant kf(\sum x_i/k)$. Since the function $\log\binom{d}{4}$ is concave, this gives us $\sum_{i=1}^{n}\log\binom{d_i}{4} \leqslant n\log\binom{\sum_{i=1}^{n} d_i/n}{4}$. Since by Euler's formula the average degree $\sum_{i=1}^{n} d_i/n$ is bounded from above by 6, we get $\prod_{i=1}^{n}\binom{d_i}{4} \leqslant \binom{6}{4}^n = 15^n$. This yields the bound on the number of regular edge labelings of G of $480^{n/4} < 4.6807^n$. □

4.2 Lower bound

Next, we give lower bounds for the number of regular edge labelings in triangulated grids. We refer to the number of rows of a triangulated grid as its *height* h and to the number of columns as its *width* w. To obtain an irreducible triangulation, we add four vertices to the outside of the grid, connecting one to every vertex on the top row, one to the bottom row, one to the left column, and one to the right column. The total number of vertices of the augmented grid is $n = hw + 4$.

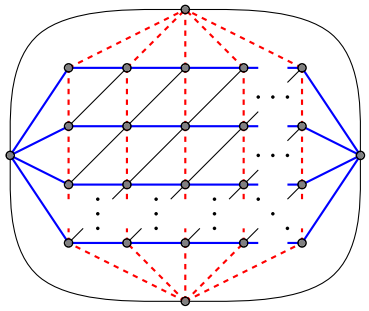


Figure 4.2: The diagonals in the triangulated grid can be colored arbitrarily if all horizonal edges are blue, and all vertical edges red.

A simple lower bound on the number of regular edge labelings in a triangulated grid is $2^{n-h-w-3}$, which is in $2^{n-O(\sqrt{n})}$ for $h = w$. To see this, color all horizontal edges blue and all vertical edges red as shown in Figure 4.2. Now all vertices already have the required four intervals of alternating red and blue edges and these intervals cannot be broken up by the diagonals, as these are all adjacent to intervals of both colors. Therefore all $n - h - w - 3$

diagonals can be colored independently blue or red, which yields the lower bound. This already shows that the number of regular edge labelings is exponential in n, but we expect that there are many more regular edge labelings in a triangulated grid, because fixing the color of all horizontal and vertical edges, i.e., two-thirds of the edges, seems very restrictive.

We will show a better lower bound by only prescribing the edges of every h'th row to be blue, and not prescribing any colors for the edges of columns. (Nonetheless, the edges of the first and last column will be colored red in any regular edge labeling.) We color the parts between the blue rows independently. Therefore, we can assume for the moment that $h = h' + 1$. Larger values of h (relative to h') will not change the analysis, but will improve the lower bound.

Our bounds require the analysis of large matrices, so part of the proof is by computer. We first describe all steps for $h' = 1$, i.e., the edges of all rows are blue. In this case, we can do all calculations by hand. Then we show how to generalize the method for larger values of h'.

We color the triangulated grid from left to right. The edges of the first and last column will need to be colored red, since by Lemma 1(b), a regular edge labeling has no monochromatic triangles. Assume we have colored the triangulated grid up to the ith column. We call the edges of the ith column and the diagonals connecting to this column from the left the ith *extended column*. Assuming we have no restriction from the right, our options for coloring the $(i + 1)$st extended column are determined by the colors of he ith extended column.

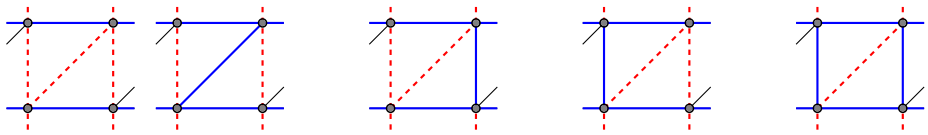


Figure 4.3: The possible labelings of an extended column for the single row case: two options for red to red and one option each for red to blue, blue to red and blue to blue.

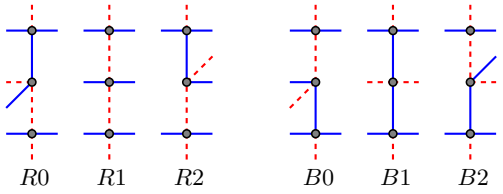
If $h' = 1$, the previous column can be either red or blue, while the color of the previous diagonal does not influence our choices for this column. If the previous column is red, we can make this column red, too, and choose either color for the diagonal. We can also make this column blue, but then the diagonal needs to be red to satisfy the constraints around the top vertex of this column. Likewise, if the previous column was blue, our diagonal needs to be red to satisfy the constraints around the bottom vertex of the previous column. These possibilities are depicted in Figure 4.3.

We can represent these coloring options as a transition matrix M (shown to the right), using the column colors as state. We can compute the number of colorings up to the ith extended column by starting in the red state (represented as $(1, 0)$) and repeatedly multiplying it with M. The resulting vector gives us the number of colorings ending in a red or a blue edge.

From \ To	R	B
R	2	1
B	1	1

Since M has only positive elements, it is primitive. By Theorem 2(c) the ratio between the number of labelings ending in a red column up to the ith extended column and up to the $(i+1)$th extended column converges towards the largest eigenvalue of M. This eigenvalue is $\phi+1 > 2.61803$. So for any $\varepsilon > 0$, we obtain more than $(\phi+1-\varepsilon)^w$ labelings for sufficiently large w. Since we need to add two vertices to add a single column, this yields a lower bound of $(\phi+1-\varepsilon)^{(n-4)/2}$ for sufficiently large width w. If we now increase h, we need to add h vertices to add $h-1$ columns, i.e., we get a lower bound of $(\phi+1-\varepsilon)^{(n-4)(h-1)/h}$, which for sufficiently large h and w is larger than 2.61803^n.

Next we consider the case where $h' > 1$, using $h' = 2$ as an example. Since we prescribe the color of less edges, this will yield a better bound. The largest change from $h' = 1$ is that we need to extend the states with information about the vertices, as just using the colors of the edges is not sufficient to decide how the coloring can be extended. Therefore we include for each vertex the number of color switches there should be in the next extended column. With this information, we can reconstruct the colors of all column edges from a single colored edge, so we will describe the state by the color of the bottom column edge, followed by the number of color switches at each vertex, moving upwards. As an example, the states for $h' = 2$ are given in Fig. 4.4.



$$\begin{pmatrix} 1 & 4 & 1 & 1 & 2 & 5 \\ 0 & 5 & 2 & 2 & 1 & 4 \\ 0 & 2 & 2 & 1 & 1 & 1 \\ 0 & 2 & 2 & 2 & 2 & 1 \\ 0 & 1 & 2 & 1 & 2 & 2 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad \begin{pmatrix} 5 & 2 & 2 & 1 \\ 2 & 2 & 1 & 1 \\ 2 & 2 & 2 & 2 \\ 1 & 2 & 1 & 2 \end{pmatrix}$$

Figure 4.4: The states for $h' = 2$, the corresponding transition matrix and reduced transition matrix.

Some states that can be described in this way cannot in fact be part of a regular edge labeling. We call such states *infeasible*. A state is feasible if it can be reached from the initial all-red state ($R1$ in the case of $h' = 2$) and if the all-red state can be reached from it. Thus a state is feasible if and only if it is in the strongly connected component of the all-red state. For example, looking at the transition matrix for $h' = 2$ given in Figure 4.4, we can see that the first state, $R0$, doesn't have any incoming transitions from other states and the last state, $B2$, doesn't have any outgoing transitions to other states. Therefore these states are both infeasible and the matrix can be reduced to include only the feasible states. In our implementation, we use two depth-first searches through the adjacency graph corresponding to the transition matrix and starting from the all-red state to determine the feasible states. The first search traverses the edges in the usual way to mark all states that are reachable from the all-red state, while the second search traverses each edge backwards, to mark all states from which the all-red state is reachable. The feasible states are exactly those states marked by both searches.

The resulting reduced matrix is primitive, since it is irreducible by construction, and there is always at least one transition from the all-red state to itself by coloring all diagonals and horizontal edges between the two columns blue. When constructing a regular edge coloring, we start with the all-red column and color the columns one by one. By Theorem 2(c) the number of regular edge colorings (and therefore the number of regular edge labelings) increases with each new column by a factor that converges to the largest eigenvalue of the reduced transition matrix. Therefore, a strict lower bound on this eigenvalue $\lambda_{h'}$ of this matrix gives us a strict lower bound for the growth rate per column (ignoring a constant number of initial columns).

We obtain this strict lower bound on $\lambda_{h'}$ by taking a nonnegative non-zero state vector x, multiplying it with the transition matrix and determining the minimum growth rate for the non-zero elements. If the vector is not an eigenvector of A (i.e., the growth rate is not the same for all non-zero states) then the minimum growth rate is a strict lower bound on $\lambda_{h'}$ by Theorem 2(b). Table 4.1 gives minimum and maximum growth rates for $h' = 2$, where

A is the reduced transition matrix and x_0 is the vector with a 1 for the all-red state and 0 otherwise. It shows that the growth rates converge quite rapidly to $\lambda_{h'}$. We use $x = x_0 A^{100}$ for our lower bounds. Since in all the cases that we consider the vector x is positive, we also obtain an upper bound on $\lambda_{h'}$ by the maximum growth factor.

t	1	2	3	4	100
$f_A(x_0A^t)$	6.8	7.56	7.80	7.87	7.89167
$g_A(x_0A^t)$	13	8.77	8.10	7.94	7.89167

Table 4.1: Minimum and maximum growth rates for $h' = 2$. The values for the smaller values of t are rounded to 2 decimal places, while the values for $t = 100$ are rounded to 5 decimal places, although they only start to deviate at the 58th decimal place.

As for the case $h' = 1$, we now use several copies of h' rows beneath each other to obtain a larger triangulated grid. The growth rate per vertex in this way approaches $\lambda_{h'}^{1/h'}$. Our results are given in Table 4.2. [1]

h'	1	2	3	4	5	6	7
$\lambda_{h'}$	2.61803	7.89167	24.5036	76.8353	241.977	763.785	2414.05
$\lambda_{h'}^{1/h'}$	2.61803	2.80921	2.90453	2.96067	2.99746	3.0233	3.04263

Table 4.2: Lower bounds on the growth rate per column and per vertex for different values of h'. Note that these are rounded down, and that our upper bounds on $\lambda_{h'}$ (and $\lambda_{h'}^{1/h'}$) equal the (unrounded) lower bounds up to at least 10 significant digits.

Theorem 5 *The number of regular edge labelings of the triangulated grid is in* $\Omega(3.04263^n)$.

4.3 Other techniques

The *twisted cylinder* is sometimes a good alternative for counting structures that are numerous on the triangulated grid. For example, it has been used to count simple and hamiltonian cycles on planar graphs by Buchin *et al.* [9]. Imagine taking a piece of squared paper and bending it to make the ends meet and form a cylinder. Now instead of lining up the rows with each other, shift everything one square to the right. This produces a single line of squares, twisting itself around the cylinder. This can be drawn as a planar graph, depicted to the right, as if you are looking through the cylinder. The advantage over the triangulated grid is that now cells can be added one at a time, leading to a far less complicated transition matrix than if you would add an entire column at a time.

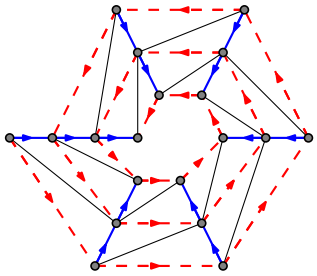


Figure 4.5: A planar drawing of the twisted cylinder with width 6.

[1] Our code for generating the transition matrices and estimating the eigenvalues can be found at `http://www.win.tue.nl/~speckman/demos/LowerBoundREL.zip`

Unfortunately, the twisted cylinder has only few regular edge labelings. Although it resembles the triangulated grid in many ways, the transformation used when drawing the graph in the plane causes the graph to have a very limited number of regular edge labelings. This is easiest to see when looking at the regular edge labeling for the first simple lower bound we gave using the triangulated grid, where every vertical edge was colored red and oriented from bottom to top and every horizontal edge was colored blue and oriented from left to right. Applying this labeling to the twisted cylinder makes both the red and blue edges move inwards. This can never lead to a valid regular edge labeling, as this requires that the blue edges form a bipolar orientation with the west vertex as source and the east vertex as only sink, both on the outside of the spiral.

We also found another upper bound on the number of regular edge labelings, which we sketch in the following. Although it is worse than our current bound, the reasoning may still be interesting.

Let G be an irreducible triangulation with $n+4$ vertices. We call the vertices adjacent to the exterior face *exterior vertices*. All other vertices are *inner vertices*. If we call the average degree of the inner vertices d, the number of triangles adjacent to the exterior vertices is $(6-d)n+2$ by the Euler characteristic. Since by Lemma 1(b) there can be no monochromatic triangles, the coloring of these triangles is already fixed. Again using the Euler characteristic, we can see that there are $(d-4)n$ non-fixed triangles left. We call a corner of a triangle *monochromatic* if its two edges have the same color. If they have a different color, we call it *bichromatic*. Since no triangle may be monochromatic, each triangle must have one monochromatic corner and two bichromatic corners. A regular edge labeling can be defined by its set of monochromatic corners. By letting each non-fixed triangle choose its monochromatic corner, we obtain a bound of $3^{(d-4)n}$ labelings.

Not every assignment of monochromatic corners produces a valid regular edge labeling, as every inner vertex v must have exactly 4 incident bichromatic corners (one at each switch between the four sets of colored edges) and d_v-4 incident monochromatic corners, where d_v is the degree of v. If we now look at a random assignment of monochromatic corners, we can say something about the probability that all vertices will satisfy this condition. Since there are $4n$ bichromatic corners in total, asking that all inner vertices have at most four incident bichromatic corners is the same as asking that they have exactly four. Let A_i denote the event that vertex i has at most four incident bichromatic corners. Then

$$P\Big(\bigcap_i A_i\Big) \;=\; P\Big(A_i \big| \bigcap_{j\neq i} A_j\Big) \cdot P\Big(\bigcap_{j\neq i} A_j\Big).$$

The first factor asks for the probability that vertex i has "few" incident bichromatic corners, given that all other vertices already have "few" incident bichromatic corners. However, this condition makes it more likely that vertex i has many incident bichromatic corners instead, so this condition only decreases the probability of vertex i having few incident bichromatic corners. Therefore this probability is less than or equal to $P\big(A_i\big)$, which implies that $P\big(\bigcap_i A_i\big) \leqslant \prod_i P\big(A_i\big)$.

Now let $p(d_i) = P\big(A_i\big)$. Since $\ln p(d)$ is concave for $d \geqslant 4$, we can use Jensen's inequality to show that $\prod_i P\big(A_i\big) \leqslant p(d)^n$. Thus the probability that any of those $3^{(d-4)n}$ labelings is valid is at most $p(d)^n$. With the average degree being at most 6, this gives us a bound of $p(6)^n \cdot 3^{(6-4)n} \approx 5.84^n$.

Chapter 5

Application to rectangular cartograms

A *rectangular cartogram* is a thematic map where every region is depicted as a rectangle. The area of the rectangles corresponds to a geographic variable, such as total population or total livestock. In this chapter, we describe how to find good regular edge labelings for rectangular cartogram construction and present experimental results. We follow the iterative linear programming method presented by Speckmann *et al.* [24] to build a cartogram from a regular edge labeling. We allow false adjacencies between sea regions only, and bound the maximum aspect ratio of all regions at 12.

5.1 Quality measures

We consider two general quality criteria: the relative position of the rectangles and the cartographic error of the resulting cartogram.

To make a rectangular cartogram as recognizable as possible, it is important that the directions of adjacency between the rectangles of the cartogram follow the spatial relation of the regions of the underlying map as closely as possible. These directions of adjacency are given by a regular edge labeling, so we can assess the recognizability of a rectangular cartogram by looking at its regular edge labeling. We use two quality measures to quantify this. The first measure is based on region centroids [22] and considers the direction between the centroids of two regions as their "true" direction of adjacency. The quality of a single adjacency is then expressed in terms of the deviation from this direction, measured as the smallest angle between the direction suggested by the labeling and the direction between the centroids (see Fig. 5.1 on the left). We refer to this measure as the *angle deviation*.

The angle deviation measure tends to perform quite well, although it can lead to counter-intuitive results in some cases. For example, placing the rightmost region in Fig. 5.1 to the right of the other region seems preferable to placing it below that region, although both have roughly the same angle deviation. Hence we also consider a second measure which is based on the bounding boxes of the regions. The *bounding box separation distance* (bb sep dist) measures the distance these bounding boxes would need to be moved to separate them in the direction indicated by the edge label (see Fig. 5.1 on the right).

Of course we do not have to to consider only the regular edge labeling to determine the recognizability of a cartogram, we can also look at the cartogram itself. The angle deviation measure has a natural translation to the cartogram, by comparing the direction between two centroids in the input map to the actual direction between the centroids of these regions in the cartogram. This is illustrated in Fig. 5.2. Again, we measure the quality of a single

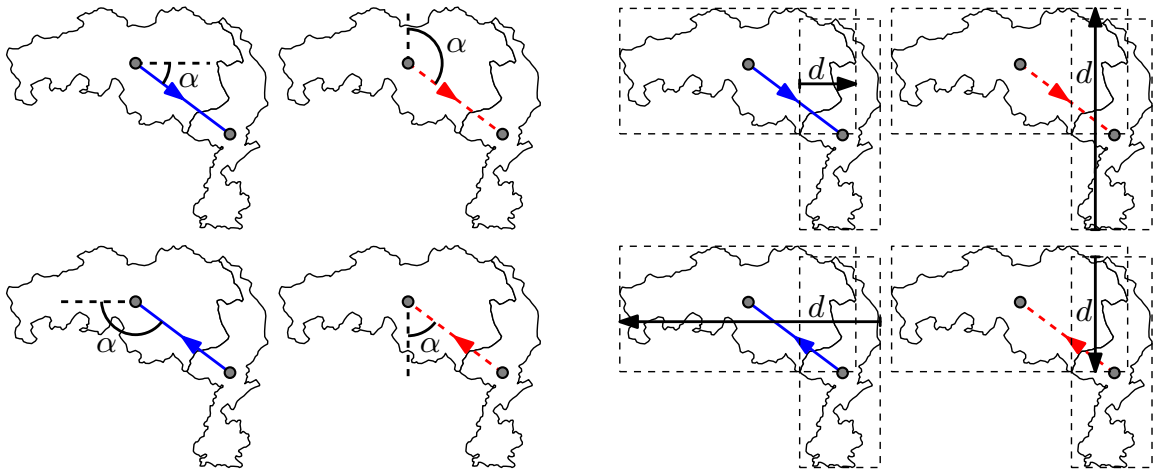


Figure 5.1: The angle deviation (left) and bounding box separation distance measures (right).

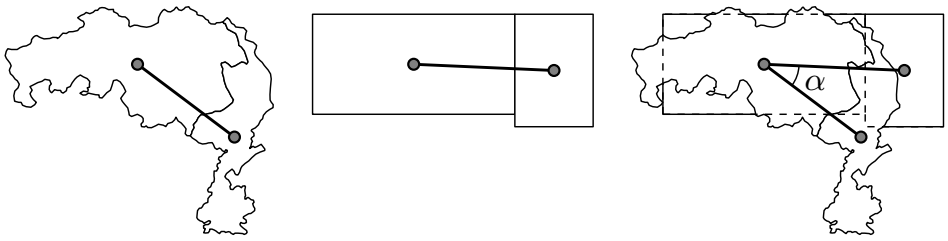


Figure 5.2: Two regions with the direction between their centroids indicated, the corresponding regions in the rectangular cartogram and an overlay of these two, indicating the resulting angle deviation.

adjacency by the smallest angle between these directions. We refer to this measure as the *resulting angle deviation*.

These measure give only the quality for a single adjacency. To compute the quality of an entire regular edge labeling, we consider both the average and the maximum error over all edges of the labeling, except for adjacencies between two sea regions. These are ignored, because they are less relevant for the perception, as the boundaries between them aren't shown in the resulting cartogram. When comparing two labelings using the maximum angle deviation or maximum bounding box separation distance, the second largest errors are compared when the largest are equal, and so on. We also consider a *binary* version of the first two measures, which first determines the "correct" color and direction for each edge by taking the combination that performs best according to the chosen measure and then counts the number of edges that are labeled correctly.

Another important quality criterion for cartograms is the *cartographic error*. This measures how well the areas of the regions in the cartogram match their prescribed areas. It is defined as $|A_c - A_s|/A_s$, where A_c is the area of the region in the cartogram and A_s is the specified area of that region, given by the geographic variable to be shown. As before, we consider both the maximum and average cartographic error over all regions of the cartogram.

We strive to construct cartograms with both low cartographic error and high recognizability, hence we consider various ways to combine the two quality criteria. One possibility is to

take a weighted average, another possibility is to bound the cartographic error at a certain percentage, while optimizing one of the recognizability measures, which we call a *bounded* measure.

To have simulated annealing produce comparable behavior with the various quality measures, we scale each measure to produce values between 0 and 1. This is done differently for each measure. For all angle deviation measures, we divide the resulting quality by π. For the maximum bounding box separation distance, we divide the result by the maximum possible distance, while for the average, we divide the sum of the distances as specified by the labeling by the sum of the maximum distances for each edge. The bounded measures return the quality of measure that is being optimized divided by 10 if the bound on cartographic error is met and add 0.9 to this otherwise.

5.2 The Netherlands

The provinces of the Netherlands, shown to the left in Fig. 5.3, form our first test case. The augmented dual graph of this map has only 408 regular edge labelings, which can be enumerated in less than a second. The corresponding distributive lattice has a diameter of 17. Despite these small numbers, the map is large enough to show interesting trends.

Fig. 5.3 shows cartograms produced by enumerating all labelings and taking the best one according to various quality measures. The first data set shows total population on January 1st 2009, the second total livestock in 2009[1]. The color of a region corresponds to its cartographic error, with red indicating that the region is too small and blue indicating that it is too big. The saturation corresponds to the magnitude of the error, a white region has a cartographic error of at most 5%, while a fully saturated region has a cartographic error of over 30%. The bounded measures optimize the maximum error of their respective measures, while only considering labelings that lead to cartograms with less than 5% maximum cartographic error.

Several interesting trends are visible here. Firstly, the labelings with the lowest average angle deviation and maximal angle deviation are the same, just like the labelings with the lowest average and maximum bounding box separation distance. This is not always the case, for example for the resulting angle deviation these labelings differ. Secondly, both the average and maximum measures perform well with regards to the recognizability of the cartograms, whereas the binary versions perform significantly worse. Although the recognizability measures perform better in this regard, it is interesting to see that the labelings that solely optimize the cartographic error are still very recognizable. Finally, it is clear that a combination of both measures leads to cartograms that are both recognizable and have a low cartographic error, which is exactly what we want to achieve.

Fig. 5.4 shows two scatter plots covering all labelings of the Netherlands. The left plot shows that the average angle deviation and average bounding box separation distance measures generally agree on the quality of labelings. They have a squared Pearson correlation coefficient of 0.76. The right plot shows the average bounding box separation distance and the maximum cartographic error. The correlation between these two is larger than expected, with a squared Pearson correlation coefficient of 0.50, but the truly interesting part is that the relation is mostly one-way. There are several labelings that are considered less recognizable, but still solve well, whereas all recognizable labelings solve well. This effect varies with the data set

[1]Source: Centraal Bureau voor de Statistiek, `http://www.cbs.nl/`

used. The correlation is strongest for the area of the provinces, which is shown in the scatter plot, almost as strong (a squared Pearson correlation coefficient of 0.44) for the livestock data, but negligible for the population data.

Figure 5.3: Population and livestock cartograms of the provinces of the Netherlands.

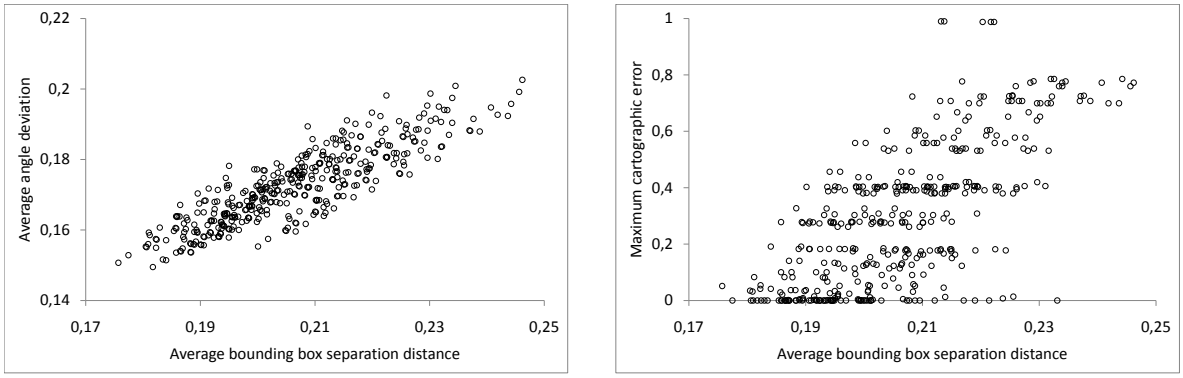


Figure 5.4: Scatter plots of the average bounding box separation distance versus the average angle deviation on the left and versus the maximum cartographic error on the right. The areas of the provinces were used as data set for the cartographic error.

5.3 Europe

The countries of Europe form our second test case. In order to compare our results to those of Speckmann *et al.* [24], we use the same simplified map of Europe, shown in Fig. 5.5, that has added sea regions and Luxembourg and Moldavia removed to ensure that the dual graph is an irreducible triangulation. This is also the reason that the map is slightly out of date (it still contains Serbia and Montenegro as a single country).

With 59 regions (including sea regions), this map is far larger than the map of the Netherlands, which is also reflected in the number of regular edge labelings of its augmented dual. While all labelings of the Netherlands could be enumerated in less than a second, it takes more than a month[2] to enumerate all labelings for Europe, of which there are at least 10700 million. Despite the large number of labelings, the lattice for Europe has a diameter of only 115. This looks promising for the application of simulated annealing, since it means that any labeling can be reached in relatively few steps. In this section, we describe our method for finding good rectangular cartograms of larger maps and present results for Europe.

Because enumeration is infeasible for these larger maps, we use simulated annealing (described in Chapter 3.2) to find good labelings for cartogram construction. From earlier experiments, we know that the choice of the initial labeling can have a large influence on the quality of the resulting cartogram. The magnitude of this influence depends on the number of steps the simulated annealing search is allowed to take: a larger number of steps allows more opportunity for exploring regions of the lattice that are further from the initial labeling. Since the measures that combine recognizability with the cartographic error require the construction of a cartogram for each evaluation, these measures are relatively slow when compared to the pure recognizability measures. Ultimately, these combined measures are still the ones we would like to use as they produce better labelings, but when starting them from the minimal labeling, it could take a large number of steps to reach a good neighborhood. Therefore we start by performing a simulated annealing search for one million steps using the average bounding box separation distance. The best labeling from this search is then used as initial labeling for the final simulated annealing search for 2000 steps with the desired combined measure.

[2]At the time of writing, our enumeration application has been running for 72 days and is still counting.

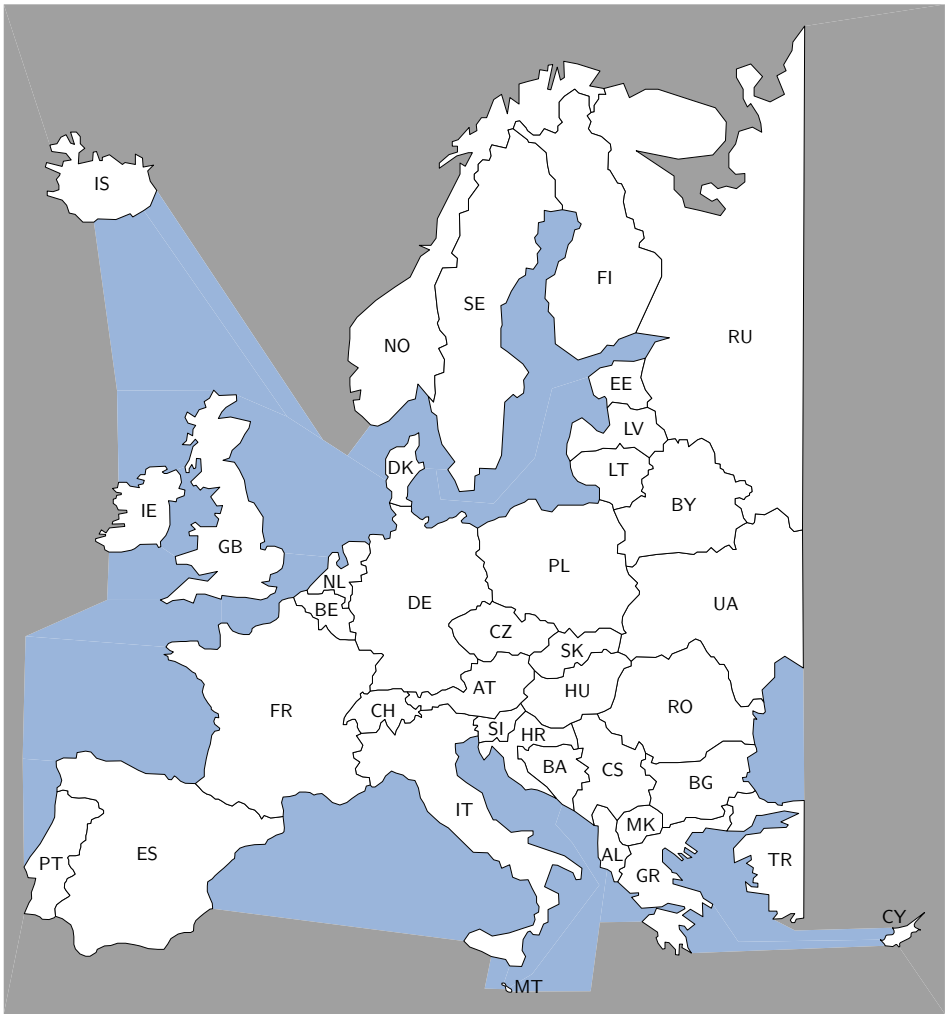


Figure 5.5: The simplified map of Europe we used as input.

Fig. 5.6 shows some results of our implementation. Note that all our cartograms have correct adjacencies for the land regions. The top figure shows the total population of the countries of Europe on January 1st 2008[3], with the populations of Luxembourg and Moldova added to Belgium and Ukraine, respectively and the populations of Serbia, Montenegro and Kosovo aggregated into Serbia and Montenegro. It has an average cartographic error of 0.000002 and a maximum cartographic error of 0.000007. The bottom figure shows the total highway length in Europe and uses the exact same map and data set as the cartograms made by Speckmann *et al.* which were based on user-specified regular edge labelings. It has an average cartographic error of 0.000001 and a maximum error of 0.000007. This is a significant improvement over the results of Speckmann *et al.* who achieved 0.022 average and 0.166 maximum cartographic error. Both cartograms were generated using a weighted average of the average cartographic error with weight 0.7 and the average resulting angle deviation with weight 0.3 as quality measure.

[3]Source: Eurostat, `http://epp.eurostat.ec.europa.eu/`

Figure 5.6: Two rectangular cartograms of Europe. The first cartogram shows the total population of each country and the second cartogram shows the total highway length.

5.4 United States

Our final test case consists of the contiguous states of the United States of America, shown in Fig. 5.7. Again, this is the same map as used by Speckmann *et al.* in order to get comparable results. With 48 states and 9 sea regions, this map is comparable in size to the map of Europe. Its augmented dual has over 10400 million regular edge labelings and the corresponding lattice has a diameter of 278.

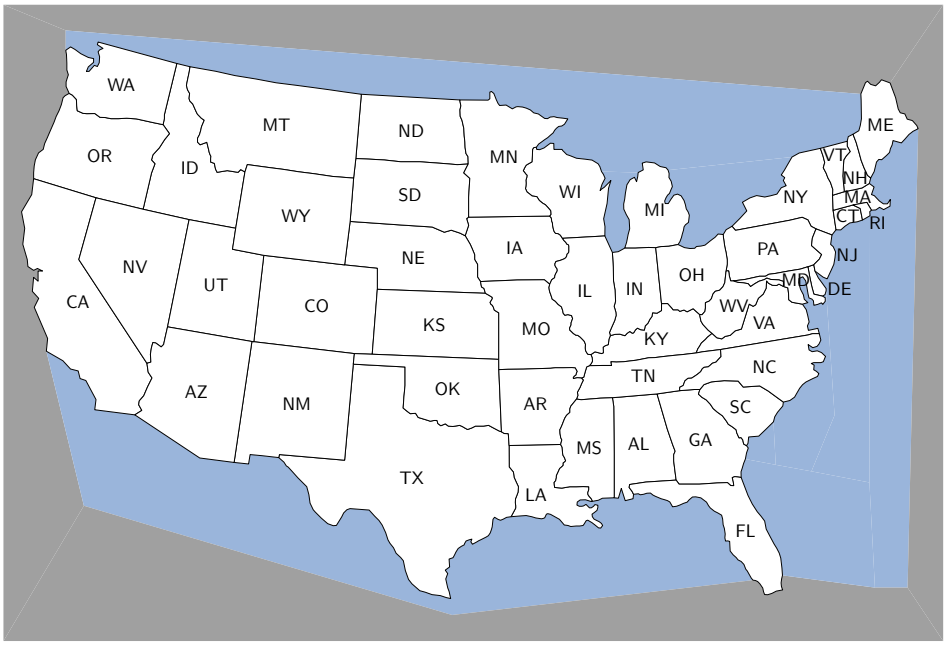


Figure 5.7: The simplified map of the United States we used as input.

Fig. 5.8 shows three cartograms, each depicting the total population of the states[4]. This is the same data set used by Speckmann *et al.* The first cartogram was generated by bounding the average cartographic error on 5% and optimizing the maximum resulting angle deviation. It achieves an average cartographic error of 0.037 and a maximum error of 0.247. The second cartogram was generated by using a weighted average of the average cartographic error with weight 0.7 and the average resulting angle deviation with weight 0.3 and has an average error of 0.039 and a maximum error of 0.207. The third cartogram was again generated using a weighted average, but instead of the resulting angle deviation, it used the average bounding box separation distance with weight 0.3. It has an average error of 0.031 and a maximum error of only 0.140. Again a significant improvement over the results of Speckmann *et al.* of 0.086 average and 0.873 maximum cartographic error.

Fig. 5.9 also shows three cartograms, this time depicting the native population of each state. All three cartograms were generated by bounding the average cartographic error at 5% and optimizing the maximum resulting angle deviation. They achieve average cartographic errors of 0.005, 0.012 and 0.017 respectively, with maximum errors of 0.077, 0.228 and 0.196.

[4]Source: the US Census Bureau, `http://www.census.gov/`

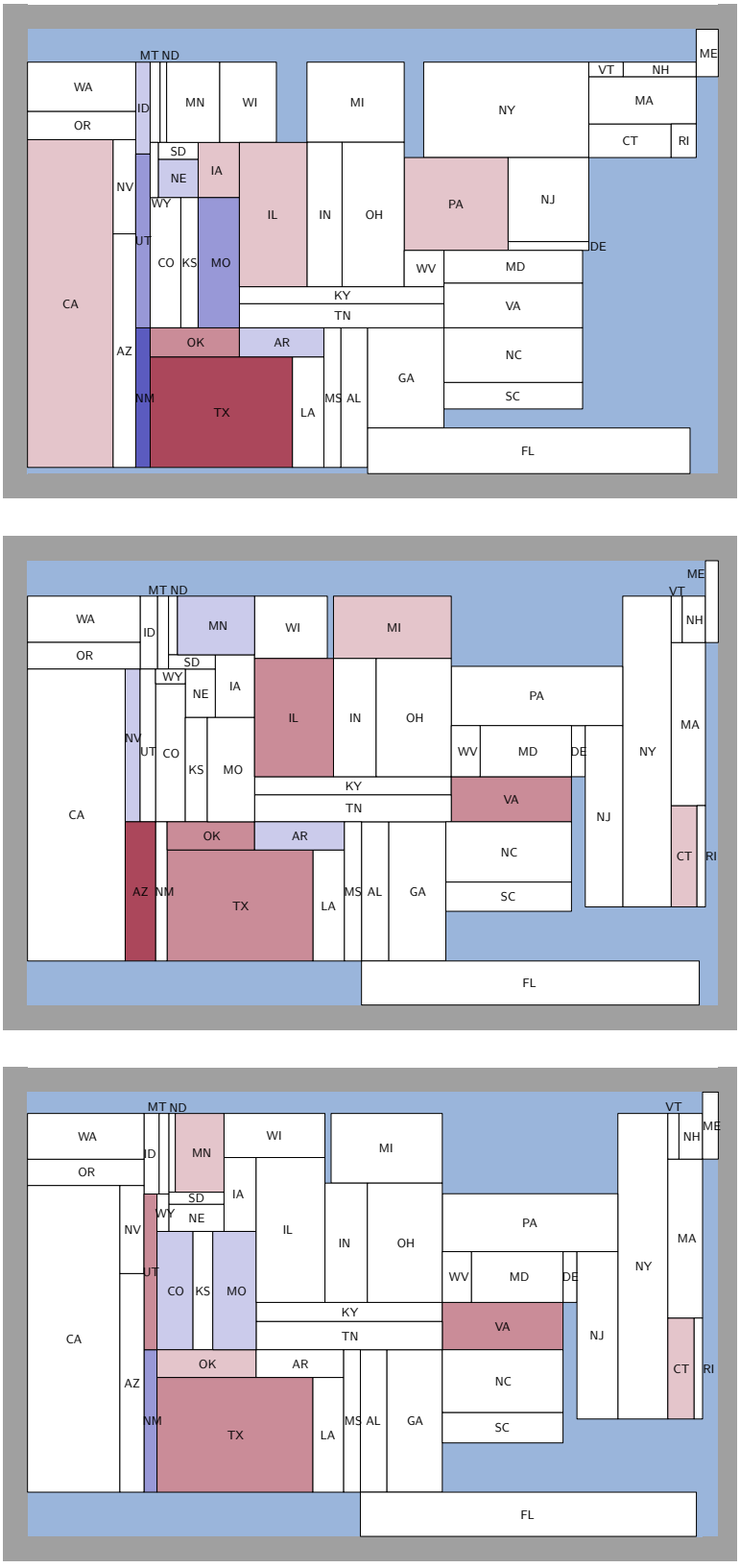


Figure 5.8: Three rectangular cartograms depicting the total population for each of the contiguous states in the United States of America.

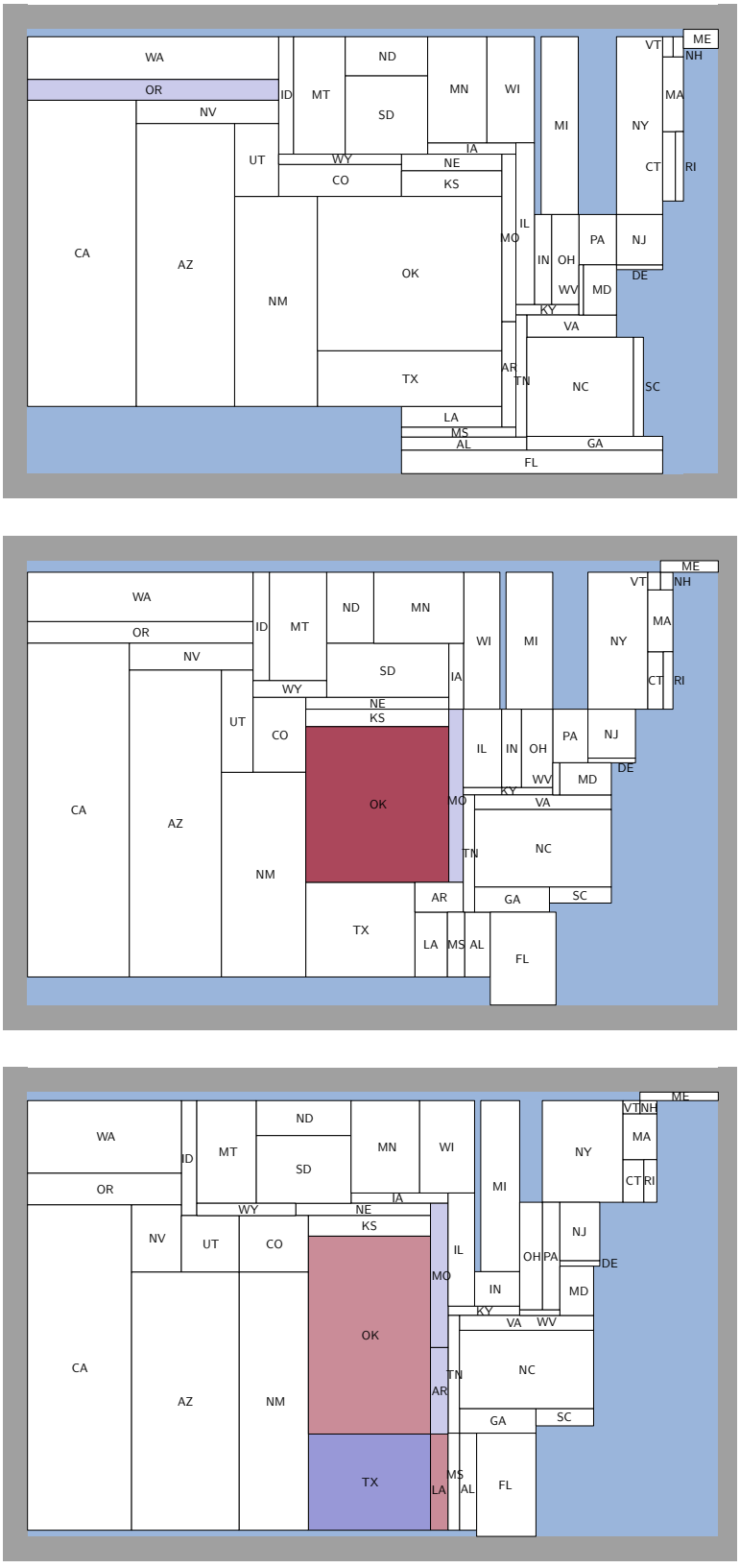


Figure 5.9: Three rectangular cartograms depicting the native population for each of the contiguous states in the United States of America.

Chapter 6

Conclusions and future work

In this thesis we showed how to find optimal or near-optimal regular edge labelings for various quality criteria. We evaluated these methods by using them to generate high quality rectangular cartograms. We also showed that every irreducible triangulation with n vertices has less than $O(4.6807^n)$ regular edge labelings and that there are irreducible triangulations with $\Omega(3.0426^n)$ regular edge labelings.

Regarding the experiments, we can conclude that our fully automated method to find optimal regular edge labelings for cartogram construction performs at least as well as semi-manual methods and frequently leads to better cartograms. Although this shows that false adjacencies are not necessary to generate cartograms with low cartographic error, using our approach to optimize the regular edge labeling in addition to allowing some adjacencies to slip a little could result in even better cartograms. Our recognizability measures also raise some interesting questions for future research focusing on user perception: Which measure matches user perception best, regarding the direction of adjacency of two regions? Are the cartograms produced using this measure indeed more recognizable? Are there other measures that produce even better results? Finally, while our choices of cooling schedule, acceptance probability function and neighbor generation seem to work well for our simulated annealing implementation, we did not perform any experiments that establish them as the best values. Tweaking these parameters to the specific measures and lattice structure could improve the chances of quickly finding high-quality results. Another option is to use other global optimization schemes such as genetic algorithms.

Of the two bounds we proved, we believe that the lower bound is closer to the true value, and improving the upper bound seems to be a promising open problem. This improvement could come from working more globally and using more specific properties of regular edge labelings. The lower bound can still be increased by applying our method for larger values of h', but when increasing it from 6 to 7, he bound increased by less than 0.02. We estimate that subsequent increases of h' will yield even smaller improvements of the bound, while using much more computing power and time.

There are many more related problems that are still unsolved. They can be divided in two groups: problems dealing with more theoretical aspects of regular edge labelings and problems dealing specifically with rectangular cartogram construction.

From an algorithmic point of view, it would be interesting to find a more efficient algorithm that produces a regular edge labeling consistent with a given partial labeling, or reports the non-existence of such a labeling. The algorithm presented by Eppstein and Mumford [12] can solve this problem, but they have a quadratic running time only if the lattice representation

is known and it is unclear how much time it takes to build this representation (although it is certainly polynomial). Working directly on the graph instead of via the lattice could give a much better running time. Some more fundamental decision problems have also not been fully solved yet. For example, it is unknown if it is possible to decide efficiently if a given irreducible triangulation has a sliceable dual. Further, although Eppstein *et al.* [13] give an algorithm that finds a one-sided dual if it exists, this algorithm is not fully polynomial. It is still unknown whether this problem is NP-complete.

Concerning the generation of rectangular cartograms, one of the few steps that has not been automated yet is dividing the sea into sea regions such that the resulting map has an irreducible triangulation as dual graph. There are always many sets of possible sea regions and this choice can have a very large influence on the quality of the resulting cartogram. A good choice of sea regions increases recognizability by preserving the shape of the coastline or border and decreases cartographic errors by shrinking or expanding to compensate for errors in adjacent land regions. For example, looking at the US population cartograms in Fig. 5.8, Texas is too small in each of them. Splitting the large sea region at the bottom into two and letting Texas be adjacent to both of these and the South border would allow more room for it to expand, while having little influence on other regions. This would also correspond better with the input map and make more sense geographically, since the right sea region would then represent the Gulf of Mexico, while the left would represent Mexico. There is also no algorithm yet that can decide whether an errorless rectangular cartogram can be made for a given graph and set of weights.

Bibliography

[1] E. Aarts, J. Korst, and W. Michiels. Simulated annealing. In *Search Methodologies*, pages 187–210. Springer, 2005.

[2] E. Ackerman, G. Barequet, and R. Y. Pinter. A bijection between permutations and floorplans, and its applications. *Discrete Applied Mathematics*, 154(12):1674–1684, 2006.

[3] O. Aichholzer, T. Hackl, B. Vogtenhuber, C. Huemer, F. Hurtado, and H. Krasser. On the number of plane graphs. In *Proceedings of the 17th Symposium on Discrete Algorithms (SODA)*, pages 504–513, 2006.

[4] K. Amano, S. Nakano, and K. Yamanaka. On the number of rectangular drawings: Exact counting and lower and upper bounds. Technical Report 2007-AL-115, IPSJ SIG, 2007.

[5] D. Avis and K. Fukuda. Reverse search for enumeration. *Discrete Applied Mathematics*, 65(1):21–46, 1996.

[6] J. Bhasker and S. Sahni. A linear algorithm to check for the existence of a rectangular dual of a planar triangulated graph. *Networks*, 7:307–317, 1987.

[7] G. Birkhoff. Rings of sets. *Duke Mathematical Journal*, 3(3):443–454, 1937.

[8] A. Björklund, T. Husfeldt, P. Kaski, and M. Koivisto. The travelling salesman problem in bounded degree graphs. In *Proceedings of the 35th International Colloquium on Automata, Languages and Programming (ICALP), Part I*, volume 4051 of *Lecture Notes in Computer Science*, pages 198–209. Springer, 2008.

[9] K. Buchin, C. Knauer, K. Kriegel, A. Schulz, and R. Seidel. On the number of cycles in planar graphs. In *Proceedings of the 13th Computing and Combinatorics Conference (COCOON)*, volume 4598 of *Lecture Notes in Computer Science*, pages 97–107. Springer, 2007.

[10] K. Buchin and A. Schulz. On the number of spanning trees a planar graph can have. In *Proceedings of the 18th European Symposium on Algorithms (ESA)*, 2010. To appear. arXiv/0912.0712.

[11] F. R. K. Chung, R. L. Graham, P. Frankl, and J. B. Shearer. Some intersection theorems for ordered sets and graphs. *Journal of Combinatorial Theory, Series A*, 43(1):23–37, 1986.

[12] D. Eppstein and E. Mumford. Orientation-constrained rectangular layouts. In *Proceedings of the 11th Algorithms and Data Structures Symposium (WADS)*, volume 5664 of *Lecture Notes in Computer Science*, pages 49–60, 2009.

[13] D. Eppstein, E. Mumford, B. Speckmann, and K. Verbeek. Area-universal rectangular layouts. In *Proceedings of the 25th ACM Symposium on Computational Geometry (SoCG)*, pages 267–276, 2009.

[14] S. Felsner and F. Zickfeld. On the number of planar orientations with prescribed degrees. *Electronic Journal of Combinatorics*, 15(1), 2008. Research paper R77, 41 p.

[15] R. Fujimaki, Y. Inoue, and T. Takahashi. An asymptotic estimate of the numbers of rectangular drawings or floorplans. In *Proceedings of the International Symposium on Circuits and Systems (ISCAS)*, pages 856 – 859, 2009.

[16] É. Fusy. Transversal structures on triangulations: A combinatorial study and straight-line drawings. *Discrete Mathematics*, 309(7):1870–1894, 2009.

[17] R. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, Cambridge, UK, 1985.

[18] G. Kant and X. He. Regular edge labeling of 4-connected plane graphs and its applications in graph drawing problems. *Theoretical Computer Science*, 172(1–2):175–193, 1997.

[19] S. Kirkpatrick. Optimization by simulated annealing: Quantitative studies. *Journal of Statistical Physics*, 34(5):975–986, 1984.

[20] K. Koźmiński and E. Kinnen. Rectangular dual of planar graphs. *Networks*, 5:145–157, 1985.

[21] H. Minc. *Nonnegative Matrices*. Wiley-Interscience, Hoboken, NJ, USA, 1988.

[22] D. Peuquet and Z. Ci-Xiang. An algorithm to determine the directional relationship between arbitrarily-shaped polygons in the plane. *Pattern Recognition*, 20(1):65–74, 1987.

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

[24] B. Speckmann, M. Kreveld, and S. Florisson. A linear programming approach to rectangular cartograms. In *Progress in Spatial Data Handling: Proceedings of the 12th International Symposium on Spatial Data Handling*, pages 529–546. Springer, 2006.

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

[26] G. K. H. Yeap and M. Sarrafzadeh. Sliceable floorplanning by graph dualization. *SIAM Journal on Discrete Mathematics*, 8(2):258–280, 1995.