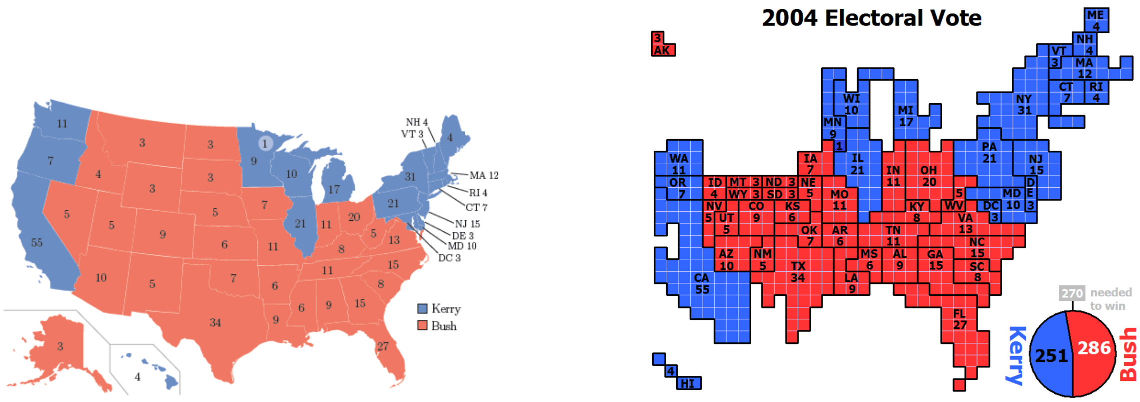
An Experimental Study of Algorithms for Cartogram Generation

Md. J. Alam, S. Kobourov, M. Schneider, S. Veeramoni

Department of Computer Science
University of Arizona

Abstract

Cartograms are used for visualizing geographically distributed data by scaling the regions of a map (e.g., countries in Europe) such that their areas are proportional to the data associated with them (e.g., GDP). Thus the cartogram computation problem can be thought of as a map deformation problem where the input is a planar polygonal map M and an assignment of some positive weight for each region in M. The goal is to create a deformed map M' where the area of each region in M realizes the weight assigned to it (no cartographic error) while the overall map remains recognizable (e.g., the topology, the relative positions and the shapes of the regions remain the same). Since achieving no cartographic error and preserving map readability are impossible to achieve simultaneously, all the cartogram generation algorithms tolerate some error in one or both of these criteria. Here we first define some quantitative measures that can be used to evaluate how faithfully a cartogram represents the weights, as well as several measures for evaluating the readability of the final representation. We then study several early algorithms for computing cartograms and two new ones and compare them in terms of our quantitative measures.



(a) (b)

Figure 1: (a) Geographically accurate map, (b) rectilinear value-by-area cartogram.

1 Introduction

A *cartogram*, or value-by-area diagram, is a thematic visualization of a planar map, where geographic regions such as countries or provinces are modified in order to realize a given set of values by their areas. This kind of visualization have been used for many years to represent census data like population, gross-domestic-product and to visualize election returns, disease incidence and other geo-referenced statistical data. Red-and-blue population cartograms of the United States are often used to illustrate the results in presidential elections starting in the year 2000. For example, in the 2004 elections, geographically accurate maps seemed to show an overwhelming victory for George W. Bush; see Fig. 1(a). On the other hand, the population cartograms effectively communicate the near even split, by deflating the rural and suburban central states; see Fig. 1(b). Incorporating vastly different scaling factors for different countries forces topological or geometrical distortions in the input map, resulting in poor readability, or even recognizability for the map. This is also undesirable since for effective visualization of the given data, the cartogram should enable the viewer to quickly relate the displayed data to the original map. This recognizability depends on preserving basic properties such as shapes and relative positions or orientations for the regions as well as the basic topology of the map. All of these goals are difficult to achieve simultaneously, and in general, it is impossible to retain even the original maps topology, while realizing the given geo-referential data perfectly [16].

For example, the rectilinear cartogram in Fig. 1(b) shows the correct distribution of red and blue squares, each representing one vote in the electoral college, but many characteristic shapes and adjacencies are compromised. Idaho and Washington are no longer neighbors, and the mirror-image shapes of New Hampshire and Vermont are lost. It also perturbs the relative north-south and east-west placement for some pairs of states. Thus none of the existing algorithms and techniques to generate cartograms for a map is "perfect"; each of them produces a "good" cartogram with respect to some criteria while this might be "bad" with respect to others. Moreover, we lack a set of performance measures which would capture the quality of a cartogram. We propose several such measures and evaluate several early and two new algorithms for cartogram generation.

1.1 Related Work

Realizing additional information on top of a geographic map dates back to the 19'th century and the highly schematized cartograms of Raisz [19], where each country is represented by an axis-aligned rectangle. Several more recent methods for computing rectangular cartogram have also been proposed [5, 12, 24]. The main advantage of such rectangular representations is that it is usually easy to realize all desired weights by area and it is usually easy to compare the resulting areas, unless the rectangles have poor aspect ratio. One disadvantage is that with rectangular cartograms, it is not always possible to maintain the topology of the map, i.e., not all given pairwise adjacencies between countries can be maintained. Thus in all the existing methods for rectangular cartogram, some adjacencies are compromised, or a few countries are allowed to have non-rectangular shapes, or some errors in the representation of weights by area are allowed [24]. Relaxing the requirement of rectangular shape, some algorithms use more general rectilinear shapes (still of constant polygonal complexity) to produced error-free cartograms. The upper bound on the complexity of the polygonal shapes used in these cartograms has been reduced from the initial 40 [6] to 34 [15], 12 [4], 10 [1] and finally to 8 [2], which also matches the lower bound [26].

All the cartogram algorithms mentioned so far have one feature in common: they extract the topology of the map, in the form of the dual graph, and then by processing these dual graphs they generate a schematized layout for the map. As a result, most cartograms produced by these methods result in maps where it is difficult to recognize which country is which by their shape, or even by their location in the map.

A different class of algorithms for generating cartograms gradually deforms the input map in order to realize different weights for the regions. One popular such method is the diffusion-based algorithm of Gastner and Newman [11], where the original input map is projected onto a distorted grid, computed in such a way that the areas of the countries match the pre-defined values. Dougenik *et al.* introduce a method based on force fields where the map is divided into cells and every cell has a force related to its data value which affects the other cells [8]. Dorling uses a cellular automaton approach, where regions exchange cells until an equilibrium has been achieved, i.e., each region has attained the desired amount of cells [7]. It is worth noting that this technique can result in significant distortions, thereby reducing readability and recognizability, which is usually one of the main advantages of this type of cartograms. Welzl *et al.* generate cartograms using a sequence of homeomorphic deformations and measure the quality with local distance distortion metrics [25]. Kocmoud and House [13] describe a technique that combines the cell-based approach of Dorling [7] with the homeomorphic deformations of Welzl *et al.* [25]. As cartogram generation is a popular research topic, there are many more algorithms that we are are aware of [16, 18, 23], and probably many more that we are not aware of; a great survey by Tobler provides more information [22].

Of particular interest in this paper is a recent method by Kämper *et al.* [14] which deforms an existing map so that countries are drawn as polygons with circular arcs instead of straight-line segments. Here the straight-line segments of the map are replaced by circular arcs so that the countries with less area in the original map than required inflate (and become cloud-shaped), while those with more area than required deflate (and become snowflake shaped). Thus in such a cartogram, it is easy to determine whether a country has grown or shrunk, just by its overall shape.

1.2 Our Contribution

In this paper we compare the performance of several existing algorithms for generating cartograms. For this purpose we first fix a number of quantitative measures for evaluating cartograms that captures both the degree of faithfulness in the representation of the geo-referential data and the extent to which the original map properties are maintained. We also designed and implemented two new algorithms, based on earlier methods, both of which offer significant improvements. In summary

- We study various quantitative measures that have been used in the literature for analysis of cartogram algorithms. We then compare how well these metrics capture different properties of cartograms. Based on this analysis, we propose a set of metrics that we use in the comparison of cartogram algorithms in this paper, and which might be useful in future evaluations.

- We develop two new heuristics for the circular-arc cartogram algorithm by Kämper *et al.* and compare this new algorithm with the one described in [14]. Our analysis shows significant improvement in performance.

- We also designed another new heuristic for obtaining a rectangular cartogram starting from the optimal rectilinear representation in [2]. This new algorithm also exhibits significant performance improvements.

- We analyze the performance of several existing algorithms as well as our new algorithms, using the proposed cartogram metrics. Although there is no clear winner in all the metrics, different algorithms exhibit better performance with respect to different metrics.

2 Performance Metrics for a Cartogram

The challenge in creating a good cartogram is to shrink or grow the regions in a map so that they reflect the set of pre-specified area values, while still retaining their characteristic shapes, relative positions, and adjacencies.In general, there are trade-offs between faithful realization of the weights and faithful representation of the initial map. We have considered several natural quantitative measures that captures these different quality criteria for a map. Often there are a number of metrics that have been used to measure the same aesthetic in different previous papers. In this section we analyze these metrics to compare how faithfully they represent the intuitive notions for the criteria they measure. Based on this analysis we select standard metrics that we use in this paper, and that will hopefully be used in future evaluations.

For the following discussion we assume that we are given an input map M that is partitioned into n countries $V = \{v_1, v_2, \ldots, v_n\}$ with polygonal boundaries. Assume that for each country v, $a(v)$ denotes the area of v in M. For each country v we are also given a weight $w(v)$ which represents the desired area for the country. Assume that both the functions a and w have been normalized to the same total area, i.e., $\sum_{v \in V} a(v) = \sum_{v \in V} w(v)$. An algorithm then constructs a cartogram M', that is a deformation of M where each country v is realized by a polygonal area of $o(v)$. We now define some quantitative metrics that measure the quality of M'.

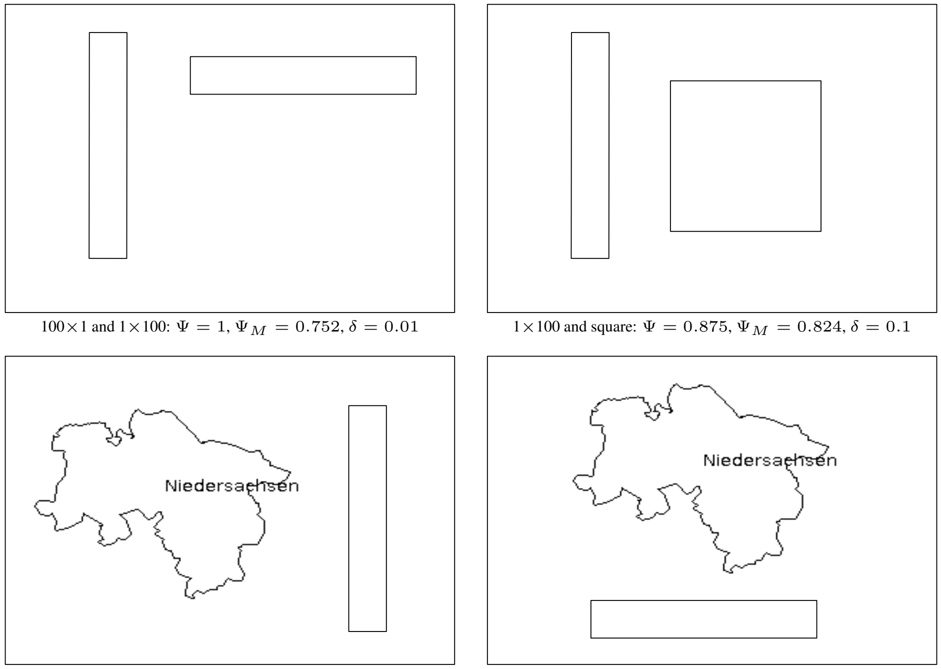
2.1 Parameters for Faithful Realization of Weights

The most common measure for weight realization is the *cartographic error*. Even though intuitively clear, this has been defined in different ways, which often make sense for the specific algorithm under consideration.

Cartographic Error: The *cartographic error* for each country v is defined as the value of $|o(v) - w(v)|$ where $o(v)$ and $w(v)$ are the obtained and required areas for the country v [17]. These error measures for the countries are then combined to obtain an estimate of the error for the whole cartogram. In some past analysis this error has been normalized by dividing the result for v by $w(v)$ [5, 12]. However if we normalize the error in this way, then the cartographic error penalizes a country that needs to grow and a country that needs to shrink in non-symmetric ways. In particular for a country v for which $w(v) \geq o(v)$ the error lies in the range $[0, 1]$, while in case $w(v) \leq o(v)$, the error theoretically can range inside $[0, \infty)$. Thus when the error for different countries are combined by taking an average or in some other way, the resulting error depends on the number of countries that need to grow or shrink, rather than the actual amount by which they need to grow or shrink. It is preferable to normalize the error for each country v by some function $f(o(v), w(v))$ that is symmetric with respect to $a(v)$ and $w(v)$ such as $\frac{|o(v)-w(v)|}{o(v)+w(v)}$ as in [16]. We considered adopting this function, but chose to normalized the error by the maximum of $o(v)$ and $w(v)$. Thus for each country v we compute the *normalized cartographic error* as $\frac{|(o(v)-w(v))|}{max\{o(v),w(v)\}}$. The reason we normalize the error in such a way is that our experiments indicate a more uniform and distribution of the normalized error.

In order to combine the cartographic error for each country to obtain an overall value for the map, previous papers have generally taken either the maximum, the average, or the root-mean-square (RMS) value of the errors over all the countries. Thus we consider three different measurements of cartographic errors for the cartograms: (i) *maximum cartographic error*, ε_m, (ii) *average cartographic error*, $\overline{\varepsilon}$ and (iii) *RMS cartographic error*, $\sqrt{\overline{\varepsilon^2}}$. Our experiment show that these measures are positively correlated with each other; in particular from a sample of 200 cartograms that we computed on different maps and data by different algorithms, we find that the correlation coefficient between ε_m, $\overline{\varepsilon}$ and between $\sqrt{\overline{\varepsilon^2}}$, $\overline{\varepsilon}$ are 0.83 and 0.97, respectively. Hence in our analysis, we use only one of these measurements: $\overline{\varepsilon}$.

Success Rate: Kämper *et al.* [14] use another way to evaluate the realization of weights, called the *success rate*. This measurement takes the obtained and required areas into account, but also uses the original area of the countries in the given geographic map. The goal is to evaluate the achieved area change, relative to the required area change. In particular, the success rate for a country v is computed by $|\frac{o(v)-a(v)}{w(v)-a(v)}|$, where $a(v)$ is the actual area of v in the input map. This metric is natural for cartograms that are generated by gradually deforming the input map to realize the weights. However, for the algorithms that do not deform the map, but rather use the topology or the dual graph of the map to generate a new layout, the actual area of a country in the original map



100×1 and 1×100: $\Psi = 1$, $\Psi_M = 0.752$, $\delta = 0.01$

1×100 and square: $\Psi = 0.875$, $\Psi_M = 0.824$, $\delta = 0.1$

1×10 and state: $\Psi = 0.824$, $\Psi_M = 0.78$, $\delta = 0.262$

10×1 and state: $\Psi = 0.824$, $\Psi_M = 0.822$, $\delta = 0.508$

Figure 2: Of the three shape comparison metrics, δ matches best with the underlying notion of map distortion.

does not make sense in comparing the cartogram performance.

2.2 Parameters for Faithful Representation of a Map

In order for a cartogram to effectively visualize some given data, such as population or GDP, it is important that the cartogram is *readable*, in that one can find and identify every country, and *recognizable*, in that one can see the same structure and topology as in the input map. We use the following quantitative metrics for evaluating the readability and recognizability for a cartogram.

Topology/Adjacency Error: The topology error τ is an estimation of how the adjacency relationships between pairs of neighboring countries have been affected in the cartogram, compared to the original map. Similar to [12] we measure this by the fraction of the adjacencies that the cartogram fails to preserve; i.e., $\tau = 1 - \frac{|E_c \cap E_m|}{|E_c \cup E_m|}$, where E_c and E_m are respectively the adjacencies between countries in the cartogram and the map.

Relative Position/Orientation: We define a metric, called the *angular orientation error*, θ, to estimate how distorted a cartogram is from its input map in terms of relative position of the pairs of countries. We compute θ by computing how much, on average, the slope of the line between the centroids of pairs of countries has changed. For applications where only orthogonal relative position (north-south, east-west) is important, sometimes this metric θ can be approximated by calculating the fraction of pairs of countries for which the relative north-south and east-west orientations have changed. We call this metric *orthogonal orientation error*, ρ. In this paper we use the metric θ to estimate the distortion in orientation.

Shape Distortions for the Countries: We need to measure how the shape a country in the generated cartogram compares with its original shape in the input map. The shape of two polygons can be compared in several ways. We want to use a metric that is translation-invariant, scale-invariant, but not rotation-invariant. Arkin *et al.* [3] compute the deviation between two polygons through an approximation of the curvature of each polygon by normalizing the polygons by perimeter and then measuring a turning function, which captures turning angle and edge length. We call this metric the *turning-angle distortion*, Ψ. This metric is translation-invariant, scale-invariant, and rotation-invariant. Thus two rectangles which are the same, up to rotation, are considered identical with this metric; see Fig. 2. However, the aspect ratio for a country is an important criterion in recognizing the shape of a country in a map. On the other extreme, Heilmann *et al.* [12] considered a metric that takes into account only the aspect ratios of the axis-aligned bounding boxes while comparing the shape of two polygons. We believe that this is also not a faithful approximation of the polygonal shape.

We have taken into consideration three different metrics for comparing the shapes of two polygons. The first metric is the turning function proposed by Arkin *et al.* [3] to compare the similarity of polygons. The second metric is a modification of the metric of Arkin *et al.* [3], where we have removed its rotation-invariance. We call this metric the *modified turning-angle distortion*, Ψ_M. The other metric is based on the idea of Hamming distance between two polygons [21]. We superimpose two polygons one top of each other and we take the percentage of area for the two polygons which are not common to both. In order to make the comparison scale-invariant, we normalize the area of polygons to unit area. Then to make it translation-invariant we consider all possible values of translation and use the one that gives the smallest error. We call this metric the *Hamming distance*, δ.

We compared the three metrics for various simple examples as well as some real world examples. The result shows that among the three metrics the *Hamming distance*, δ captures the similarity better than the other two metrics; see Fig. 2. We therefore use the metric δ in comparing the shapes of the countries in a cartogram with their original shapes.

Complexity: It is desirable for aesthetic, practical and cognitive reason to limit the polygonal complexity of the countries. We compare our implemented algorithms in terms of the worst-case polygonal complexity for the countries in the generated cartograms.

Metric Name	Notation	Definition
Average Cartographic Error	$\bar{\varepsilon}$	$\frac{1}{\lvert V\rvert}\sum_{v\in V}\frac{\lvert o(v)-w(v)\rvert}{max\{o(v),w(v)\}}$
Topology Error	τ	$1-\frac{\lvert E_c\cap E_m\rvert}{\lvert E_c\cup E_m\rvert}$
Angular Orientation Error	θ	
Hamming distance	δ	

Table 1: Definition of Performance Metrics

3 Cartogram Algorithms Implemented

In this section we give a brief description of several known cartogram algorithms (rectangular, rectilinear, diffusion-based) that we have implemented, as well as describe the new heuristics that we have developed.

Diffusion Method (*DIF*): Gastner and Newman use a diffusion method for creating cartograms [11], where the original input map is projected onto a distorted grid, computed in such a way that the areas of the countries match exactly the pre-defined values. This method uses a physical model in which the desired areas are achieved via an iterative diffusion process where flows move from one country to another until a balanced distribution is reached. After each iteration the new coordinates for points on the map are computed by interpolation of the distorted grid points. Note that this method depends on the size of the chosen grid. We have used a 1024×1024 grid that generally is sufficient for the geographic maps under consideration.

Rectangular Cartogram Algorithms (*RECT*): Not all planar maps can be drawn so that all the countries are rectangles. But if we allow some of the adjacencies to be absent, it is possible that one may find a rectangular cartogram. Such cartograms have been studied by van Kreveld and Speckmann [24], where they used three different heuristics to find a rectangular cartogram for a given map. Here we have implemented their "segment-moving heuristic" to generate rectangular cartograms. This heuristic gives two different cartogram algorithms: in one the adjacencies might be disturbed in order to realize the weights perfectly; while in the second one the adjacencies are maintained perfectly while the cartograms would contain some errors. We call these two variants *RECT* and *RECT-R*, respectively.

Evolution Algorithm for Rectangular Cartograms (*RECT-E*): Recently Buchin *et al.* have described an evolution algorithm to generate the "fittest" rectangular cartogram for a planar map. At each step they consider a number of different rectangular layouts for the map and keep only the ones that give the least error or the best "score" for a given scoring function. Then they generate a number of new rectangular layouts by combining the "fittest" old ones. The final layout is obtained after running this iterative process for several generations. We implemented this algorithm for the variant where we are required to maintain the adjacencies. Experimental evidences shows significant improvement for this algorithm over the naive segment moving heuristic of *RECT-R* (Section 4). We did not implement similar evolution algorithm for the other variant since the simple segment moving heuristic (*RECT*) gives almost zero cartographic error.

T-Shape Cartogram Algorithm (*COMB-T*): Using a Schnyder realizer [20] and the area-universality of one-sided rectangular duals [10], one can compute rectilinear cartograms with optimal complexity [2]. Each country in the resulting cartogram is drawn by a T-shape with at most 8 corners per polygon and the desired areas are obtained via an iterative process that mimics the natural phenomenon of air-pressure. This method of producing cartograms guarantees convergence to an error-free cartogram and converges very quickly in practice.

Degenerate T-Shape Cartogram Algorithm (*COMB-R*): The T-shape cartogram algorithm [10] also provides an option to generate rectangular cartograms. Since every T-shape is made of four rectangles and the layout is area-universal, one can conveniently distribute the weights arbitrarily among the four rectangles. A rectangular cartogram can be generated by randomly choosing one of these rectangle to realize all the weights while the other three have zero weights. We have modified this approach to obtain two new heuristics. In the first variant, called *COMB-R*, we run the algorithm for about 20-30 iterations, each time randomly choosing one of the four rectangles for each T-shape polygon to realize the entire weight and we return the one with minimum average cartographic error. In the second variant, called *COMB-I*, we again run the algorithm for 20-30 iterations, but at each iteration we compute the weight assignment of the rectangles as follows. We randomly pick one rectangle from each T-shape to realize the entire weight (with 50% probability), or we take the weight assignment from the previous iteration (with 25% probability), or the weight assignment from the past iteration that generated the cartogram with least cartographic error so far (with 25% probability) and modify the weight assignment in the T-shapes for 10% of the countries that gave the worst cartographic error.

Our experimental results (Section 4) show that both our *COMB-R* and *COMB-I* heuristics outperform the original algorithm [2]. However, the more complicated iterative variant does not provide significant improvement over the random variant. We therefore use the *COMB-R* variant to produce rectanglar cartograms for comparison with other algorithms.

Circular-Arc Cartogram Algorithm (*CIRC*): The method of Kämper *et al.* [14] deforms an existing map where countries are drawn as polygons with circular arcs instead of straight-line segments so that the areas for the countries closely match the predefined values. First a flow network is computed from the dual graph of the map, where bidrectional edges are created between pairs of adjacent countries. The flow on such edge (a, b) represents the area that the polygon for a transfers to the polygon for b. The capacity of this edge is assigned as the maximum "safe" area that can be transfered from polygon for a to polygon for b, without creating any crossing or overlapping polygons. Each country that needs to shrink (grow) is connected to a source (sink) node and the capacity on these edges corresponds to how much these countries need to grow or shrink. The output cartogram is the one that maximizes the flow in the network. Kämper *et al.* [14] defined *strong* and *weak* variants of circular-arc cartograms. The strong variant is more appealing from an information visualization point of view as if a country needs to grow (shrink) then none of its edges are bent inward (outward). This way all enlarged countries are cloud-shaped and all shrunk countries are snowflake-shaped, making it easy to see at a glance the type of change. The weak variant can achieve lower cartographic error. We have implemented both variants and identified some shortcomings. We also designed, implemented and tested new heuristics for both variants which address these shortcomings and result in significantly better performance.

The first shortcoming of the algorithm in [14] is that it does not allow the area of the sea to change. As a result a country cannot grow into the sea unless there is some other country that can compensate for it by

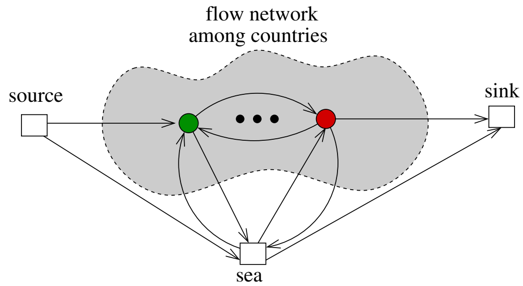


Figure 3: Modified flow formulation to allow the area of the sea to change.

shrinking near the sea. We overcome this restriction by making a small modification to the flow network; see Fig. 3. In addition to the edges in the flow network defined in [14], we add high capacity edges from the source to the sea and from the sea to the sink. This allows the flow to satisfy the need for some country to grow or shrink by changing the sea area (in other words the total land area). However we need to take care of an artifact caused by the modification: we have created a high capacity path from the source to the sink that goes only through the sea and no other country. We forbid this direct flow with a weighted flow, where the weight assigned to an edge adjacent to at least one country is twice as much as the weight on the edges from source to sea and from sea to sink. This also ensures that a flow between two countries takes precedence over a flow between a country and the sea. Because of the flexible sea, we call this variant *CIRC-F*.

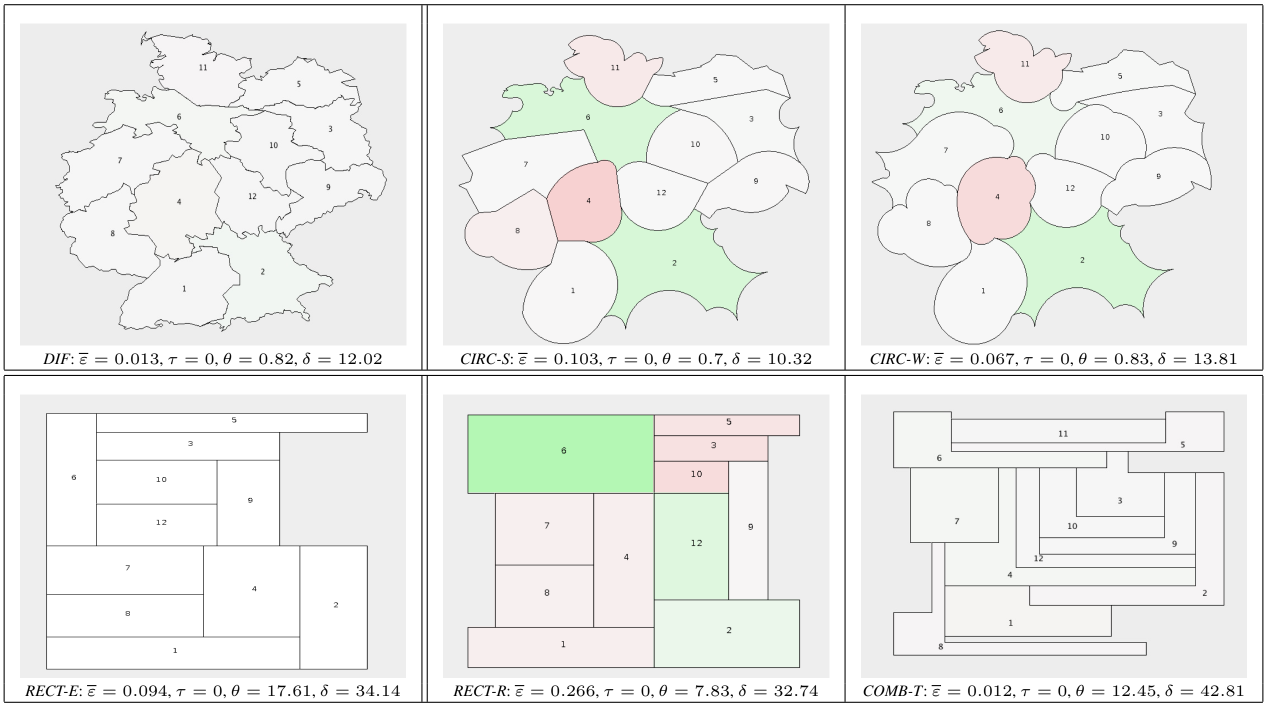
The second shortcoming of the algorithm is that if the boundary of a country consists of only edges of small length, then it is impossible to achieve significant change in the area by replacing straight-line segments by circular arcs. We propose to carefully delete intermediate degree-2 points on the boundary of a country so as to make it possible to achieve significant change in the area, while the overall shape of the country remains almost unchanged. We accomplish this with a modified version of the poly-line simplification algorithm of Douglas and Peucker [9]. At each iteration i of our algorithm, we select the border between two countries for which the difference between the signed cartographic error at the $(i-1)$-th iteration is maximum. Let c_1 and c_2 be two adjacent countries with cartographic errors of e_1 and e_2. The sign of e_1 (resp. e_2) is positive if c_1 (resp. c_2) has more area in the cartogram at the $(i-1)$-th iteration than desired. We select a pair (c_1, c_2) such that $|e_1 - e_2|$ is maximized, and then simplify the border between them by removing half of the degree-2 points provided that it improves the cartographic error. We call this variant of the heuristic *CIRC-A*.

In Section 4 we show that *CIRC-A* (which uses both the *CIRC-F* and *CIRC-A* optimizations) performs significantly better than the original circular-arc cartogram algorithm [14] in terms of cartographic error. We therefore use this variant of the heuristic to produce circular-arc cartograms for comparison with other algorithms. For simplicity in the remaining parts of the paper, we use *CIRC* to denote this variant of the heuristic and we use *CIRC-S* and *CIRC-W* to denote the strong and weak model of this heuristic.

Table 2 summarizes the high-level features of the cartograms generated by our different algorithms: *adjacency-preserving* refers to the topology of the map, $O(1)$-complexity refers to the number of corners per country, *zero-error* refers to the cartographic error, recognizability refers to angular orientation error and Hamming distance. The last two columns are discussed in the next section.

Algorithm	Adjacency Preserving	$O(1)$-Complexity	Zero-Error	Recognizability	Effect of Schematization	Exhibit Area-Change
DIF	Y	N	Y	M	Y	N
CIRC-S	Y	N	N	H	Y	Y
CIRC-W	Y	N	N	H	Y	N
COMB-T	Y	Y	Y	L	N	N
COMB-R	N	Y	N	L	N	N
RECT	N	Y	Y	L	N	N
RECT-R	Y	Y	N	L	N	N
RECT-E	Y	Y	N	L	N	N

Table 2: High-level cartogram features; Yes (Y), No (N), or High (H), Medium (M), and Low (L).



DIF: $\overline{\varepsilon} = 0.013$, $\tau = 0$, $\theta = 0.82$, $\delta = 12.02$

CIRC-S: $\overline{\varepsilon} = 0.103$, $\tau = 0$, $\theta = 0.7$, $\delta = 10.32$

CIRC-W: $\overline{\varepsilon} = 0.067$, $\tau = 0$, $\theta = 0.83$, $\delta = 13.81$

RECT-E: $\overline{\varepsilon} = 0.094$, $\tau = 0$, $\theta = 17.61$, $\delta = 34.14$

RECT-R: $\overline{\varepsilon} = 0.266$, $\tau = 0$, $\theta = 7.83$, $\delta = 32.74$

COMB-T: $\overline{\varepsilon} = 0.012$, $\tau = 0$, $\theta = 12.45$, $\delta = 42.81$

Table 3: Germany GDP cartograms by different algorithms along with the values for $\overline{\varepsilon}$, τ, θ and δ.

4 Experiments and Results

We have used three different maps, USA, Italy and Germany, to compare all the cartogram algorithms. For each of them we used several different schematizations of the map. For each of these maps we used two geo-referenced datasets: GDP and population in 2010. More details about the maps, geo-referenced data. are in Appendix A. All experiments were performed on on an Intel Core i5 1.8GHz machine with 8 GB RAM. We run each algorithm for the weight functions of GDP and population on the maps of USA, Germany, Italy. After each algorithm is run on each map and data, we compute the value of the 4 performance metrics $\overline{\varepsilon}$, τ, θ, δ. Table 3 shows GDP cartograms of Germany generated by six algorithms. The white regions have as much area as required; the redder a region is the more it needs to grow, the greener a region is, the more it needs to shrink. Appendix B contains more examples.

4.1 Circular-Arc Cartograms (*CIRC*): Old Algorithm vs New Heuristic

We compare the cartographic errors for the original weak and strong models and the four new variants; see Fig. 4. The new heuristic *CIRC-A* significantly reduces the errors in both the strong and weak model In

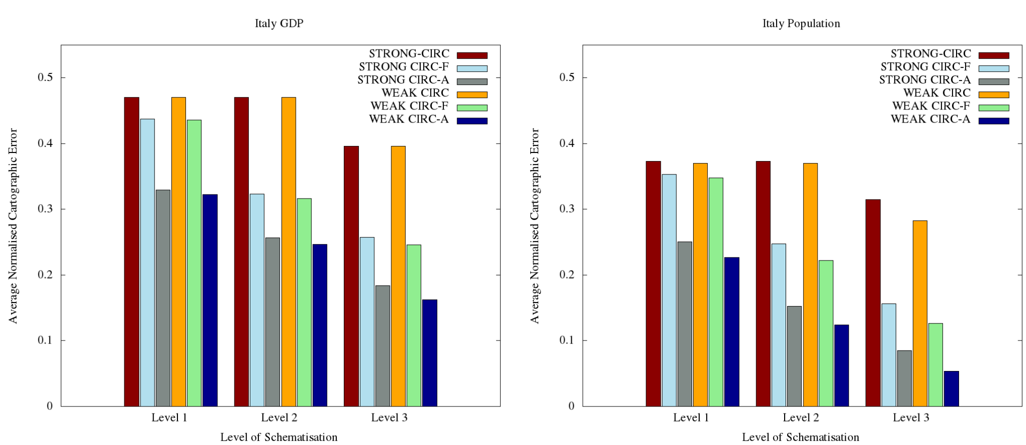


Figure 4: Cartographic Error for different variants of *CIRC*.

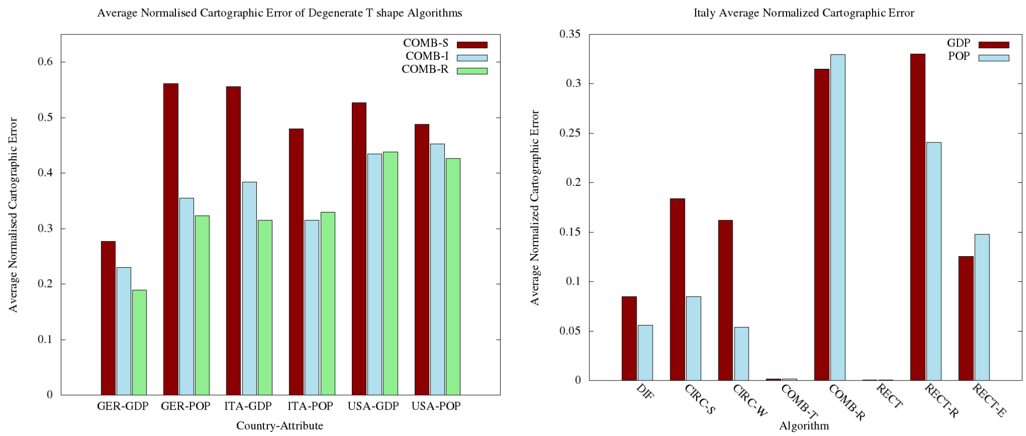


Figure 5: Average normalized cartographic error ($\bar{\varepsilon}$) for (a) different *COMB* variants and (b) for all algorithms.

particular, the combination of flexible sea and the ability to simplify borders leads to 50-100% improvement in cartographic error. We use this variant in our subsequent comparisons.

4.2 Degenerate T-shape Cartogram Algorithms : *COMB-S* vs *COMB-R* vs *COMB-I*

We implemented three different variants for the Degenerate T-shape Cartogram algorithm. Here we compare these three variants with respect to the cartographic error they produce; see Fig. 5(a). Clearly both the iterative variants outperforms the naive approach significantly. While the error generated by the two iterative variant is comparable, we will take the *COMB-R* heuristics in the subsequent analysis due to their relative simplicity.

4.3 Performance Comparison for Different Algorithms

Here we compare all the cartogram algorithms that we implemented with respect to all the four performance metrics. Figures 5(b), 6(a) and 6(b) illustrate the values of the three metrics $\bar{\varepsilon}$, θ and δ, for the cartograms generated by all algorithms. Before we analyze these results we note that the algorithms under consideration can be partitioned into two types. The first type includes the diffusion method and the circular-arc cartogram algorithm, both of which modify the input map by either moving vertices and deforming edges. The second type contains all remaining algorithms which use the map topology (i.e., the planar dual) and construct cartograms using different combinatorial properties.

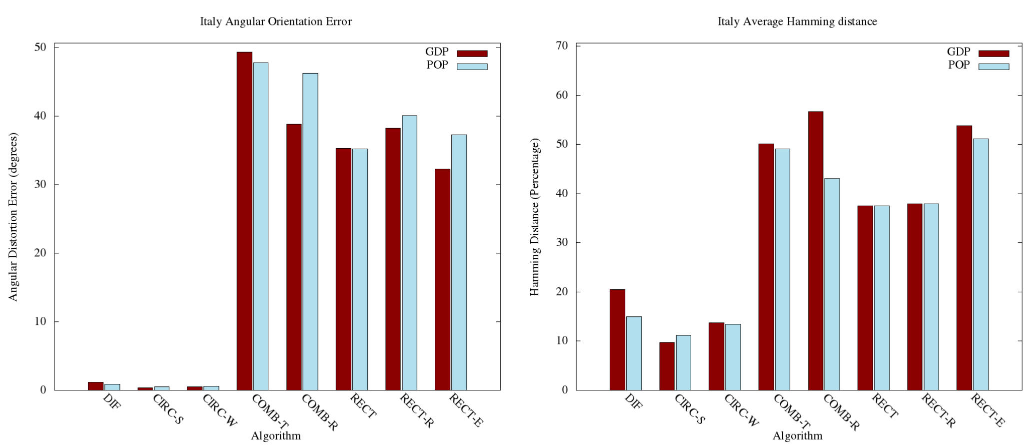


Figure 6: (a) Angular orientation error (θ); (b) Hamming distance (δ).

Since Type-1 algorithms works on the map itself rather than the dual, they produce cartograms with better readability and recongnizability than Type-2 algorithms; see Fig. 6. However, Type-2 algorithms have the advantage of having a constant polygonal complexity for the countries, while for Type-1 algorithms the polygonal complexity is the same as the input map; see Table 2.

From the Type-1 algorithms, the circular-arc cartograms performs slightly better than the diffusion method in terms of readability metrics (θ, δ); see Fig. 6. This implies that circular-arc cartograms preserve the shapes or the polygons and their relative positions better than any other algorithm. On the other hand the diffusion method generates cartograms with lower cartographic error. While the error for the circular-arc cartograms are comparable with *RECT-R* and *COMB-R*, it is much worse than *COMB-T* and *RECT* and slightly worse than *RECT-E*. A unique advantage of strong circular-arc cartograms is that one can estimate whether a country has grown or shrunk just by looking at the country shapes (cloud/snowflake shapes for expanding/shrinking countries); see Table 2.

Among the Type-2 algorithms, the *COMB-T* algorithm and the *RECT* algorithm achieve almost zero cartographic error, but the *RECT* algorithm might not preserve adjacencies; see Fig. 7. *RECT*, *RECT-R*, and *RECT-E* often require that one more more countries be deleted, in order to make the map suitable for rectangular representation. In particular, these three algorithms delete one state from Italy, two states from Germany and four states from the USA to make the graph 4-connected, which guarantees the existence of such a rectangular drawing. *RECT-R*, *RECT-E*, and *COMB-R* generate rectangular cartograms with much higher cartographic error; see Fig. 5. *RECT-R* and *RECT-E* preserve adjacencies in the map while *COMB-R* does not; see Fig. 7. Unlike *RECT-R* and *RECT-E*, *COMB-R* does not need to delete any state to compute a rectangular layout. A disadvantage of type-2 algorithms is that they do not preserve country shapes and positions compared with their type-1 counterparts; see Fig. 6.

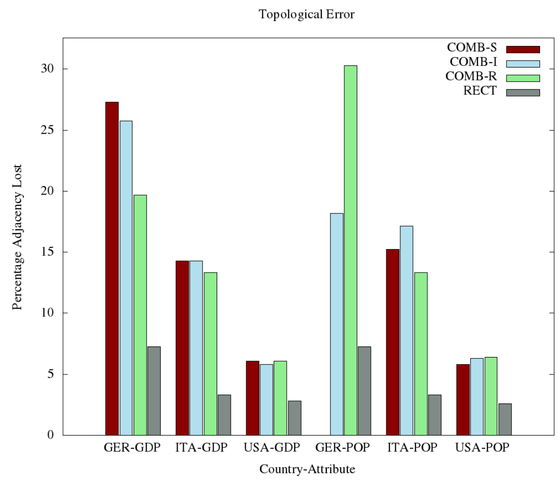


Figure 7: Topological errors (τ).

4.4 Level of Schematization

The level of schematization of the map affects the performance of Type-1 algorithms: *CIRC* and *DIF*. For circular-arc cartograms there is a general trend of reduction of error as the level of schematization increases. This is because, the greater the schematization, the farther we can bend an edge inward or outward. Schematizing too much can be detrimental as the readability and recognizability decrease. For the *DIF* cartograms there is a general trend of increase in error as the level of schematization increases. This is due to the effect of interpolation used to compute point positions. Type-2 algorithms are not affected by schematization since they only use the topology of the map.

4.5 Time-Analysis

Fig. 8 (Appendix C) shows the time required for different algorithms. The *DIF* algorithm require 30-50 seconds. The runtime for all the iterative algorithms (*CIRC-A*, *COMB-I*, *COMB-R*, *RECT-E*) depends on the preset number of iterations. The remaining algorithms are fast at 10-20 seconds. The *COMB-T* and the *RECT* algorithms guarantee convergence to zero cartographic error; we let them run for only 10 seconds, which suffices to obtain near-zero cartographic error.

5 Conclusion and Future Work

We compared a number of cartogram algorithms of different types in terms of various metrics to quantitatively evaluate how faithfully the generated cartograms represent the underlying map and the given data. We plan to link data sources (e.g., US Census) with our implementations in order to generate many different cartograms online. We also plan to perform a user study to test whether our metrics (cartographic error, topology, orientation, distortion) accurately capture the underlying notions of faithful representation. Several cartogram generation algorithms are time-consuming, making them less suitable for real-time interaction; running time improvements would be valuable.

References

[1] M. J. Alam, T. C. Biedl, S. Felsner, A. Gerasch, M. Kaufmann, and S. G. Kobourov. Linear-time algorithms for hole-free rectilinear proportional contact graph representations. *Algorithmica*, 2013 (to appear).

[2] M. J. Alam, T. C. Biedl, S. Felsner, M. Kaufmann, S. G. Kobourov, and T. Ueckerdt. Computing cartograms with optimal complexity. In *Symposium on Computational Geometry (SoCG'12)*, pages 21–30, 2012.

[3] E. M. Arkin, L. P. Chew, D. P. Huttenlocher, K. Kedem, and J. S. Mitchell. An efficiently computable metric for comparing polygonal shapes. In *Proceedings of the first annual ACM-SIAM symposium on Discrete algorithms*, pages 129–137. Society for Industrial and Applied Mathematics, 1990.

[4] T. C. Biedl and L. E. R. Velázquez. Orthogonal cartograms with few corners per face. In *Workshop on Algorithms and Data Structures(WADS'11)*, volume 6844 of *Lecture Notes in Computer Science*, pages 98–109. Springer, 2011.

[5] K. Buchin, B. Speckmann, and S. Verdonschot. Evolution strategies for optimizing rectangular cartograms. In *Geographic Information Science (GIS'12)*, volume 7478, pages 29–42, 2012.

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

[7] D. Dorling. *Area Cartograms: Their Use and Creation*, volume 59 of *Concepts and Techniques in Modern Geography*. University of East Anglia: Environmental Publications, 1996.

[8] J. A. Dougenik, N. R. Chrisman, and D. R. Niemeyer. An algorithm to construct continuous area cartograms. *The Professional Geographer*, 37(1):75–81, 1985.

[9] D. H. Douglas and T. K. Peucker. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: The International Journal for Geographic Information and Geovisualization*, 10(2):112–122, 1973.

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

[11] M. T. Gastner and M. E. J. Newman. Diffusion-based method for producing density-equalizing maps. In *Proceedings of the National Academy of Sciences of the United States of America*, volume 101, pages 7499–7504, 2004.

[12] R. Heilmann, D. A. Keim, C. Panse, and M. Sips. Recmap: Rectangular map approximations. In *IEEE Symp. Information Visualization (INFOVIS'04)*, pages 33–40, 2004.

[13] D. H. House and C. J. Kocmoud. Continuous cartogram construction. In *IEEE Visualization*, pages 197–204, 1998.

[14] J.-H. Kämper, S. G. Kobourov, and M. Nöllenburg. Circular-arc cartograms. In *IEEE Pacific Visualization Symposium (PacificVis)*, 2013 (to appear).

[15] A. Kawaguchi and H. Nagamochi. Orthogonal drawings for plane graphs with specified face areas. In *International Conf. Theory and Applications of Models of Computation (TAMC'07)*, volume 4484 of *Lecture Notes in Computer Science*, pages 584–594, 2007.

[16] D. A. Keim, S. C. North, and C. Panse. Cartodraw: A fast algorithm for generating contiguous cartograms. *IEEE Transactions on Visualization and Computer Graphics*, 10(1):95–110, 2004.

[17] D. A. Keim, S. C. North, C. Panse, and J. Schneidewind. Visualizing geographic information: Visualpoints vs. cartodraw. *Information Visualization*, 2(1):58–67, 2003.

[18] D. A. Keim, C. Panse, and S. C. North. Medial-axis-based cartograms. *IEEE Computer Graphics and Applications*, 25(3):60–68, 2005.

[19] E. Raisz. The rectangular statistical cartogram. *Geographical Review*, 24(3):292–296, 1934.

[20] W. Schnyder. Embedding planar graphs on the grid. In *ACM-SIAM Symp. Discrete Algorithms (SODA'90)*, pages 138–148, 1990.

[21] S. Skiena. *The algorithm design manual*. Springer, 1997.

[22] W. Tobler. Thirty five years of computer cartograms. *Annals of Association of American Geographers*, 94:58–73, 2004.

[23] W. R. Tobler. Pseudo-Cartograms. *Cartography and Geographic Information Science*, pages 43–50, 1986.

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

[25] E. Welzl, H. Edelsbrunner, and R. Waupotitsch. A combinatorial approach to cartograms. *Compuational Geometry*, 7:343–360, 1997.

[26] K.-H. Yeap and M. Sarrafzadeh. Floor-planning by graph dualization: 2-concave rectilinear modules. *SIAM Journal on Computing*, 22:500–526, 1993.

Appendix A

Here we give some important statistics about the three maps (USA, Germany and Italy) and the geo-referential data (GDP and Population for each map) that we have used in our experiments.

Datasets

We have used three different maps, USA, Italy and Germany, to compare all the cartogram algorithms. For each of them we used three or four different schematizations of the map. In particular we use four levels of schematization for Germany and three levels of schematization for USA and Italy. We collected the initial source files for these maps from `http://download.geofabrik.de`. We then ran a modified version of the algorithm in [9] to compute different levels of schematization for each map. For each of these maps we used two geo-referenced data: GDP and population in 2010. We collected the datasets from `http://europa.eu` and `http://quickfacts.census.gov/qfd/index.html`. All the maps and data are also available in `http://www.cs.arizona.edu/~mjalam/optocart/data.html`. The properties of these maps and the data are illustrated in Tables 4 and 5.

Country	Level of Schematization	Number of States	Number of Dual Edges	Average Polygon Complexity
USA	1	46	117	52.4
	2			27.2
	3			15.1
Germany	1	12	28	159.5
	2			66.3
	3			29.1
	4			14.4
Italy	1	15	30	32
	2			16
	3			9.3

Table 4: Properties of the input maps.

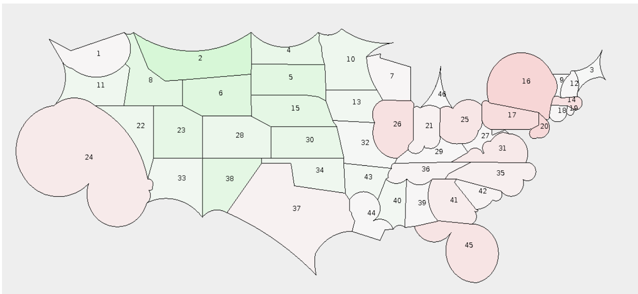
Country	Data	Maximum Value	Minimum Value	Average Value
USA	GDP	1936400	26400	284846.71
	Population	38041430	576412	5926901.69
Germany	GDP	37509	21404	27822.92
	Population	17837000	1639000	6232500
Italy	GDP	321627	6067	82289.53
	Population	9642000	320000	3377196.6

Table 5: Properties of the input data.

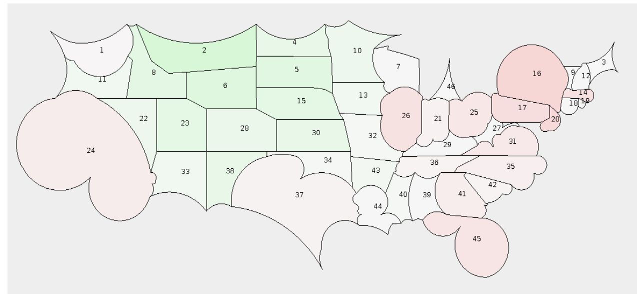
Appendix B

Here we show cartograms for USA, Germany and Italy generated by different algorithms.

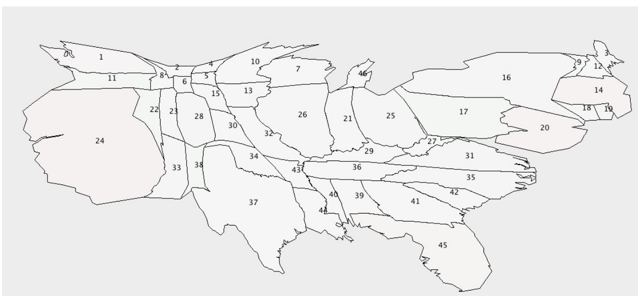
USA GDP Cartograms



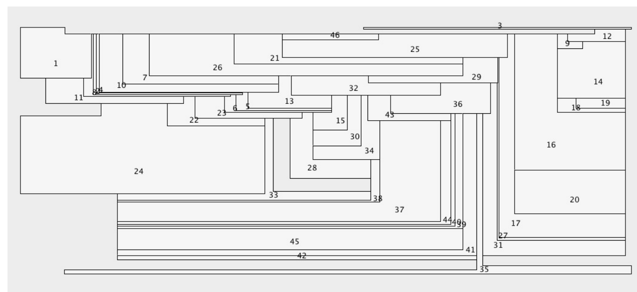
CIRC-S: $\overline{\varepsilon} = 0.407$, $\tau = 0$, $\theta = 1.86$, $\delta = 9.72$



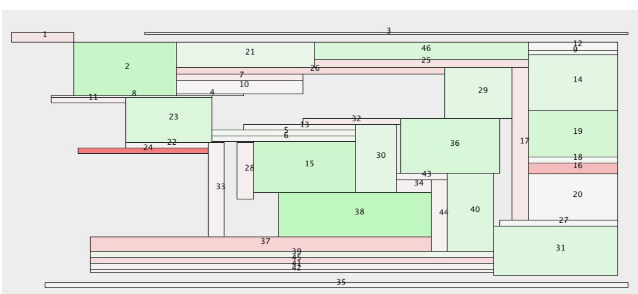
CIRC-W: $\overline{\varepsilon} = 0.395$, $\tau = 0$, $\theta = 1.91$, $\delta = 10.86$



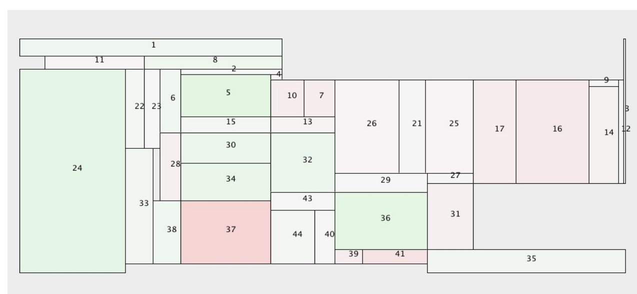
DIF: $\overline{\varepsilon} = 0.092$, $\tau = 0$, $\theta = 7.91$, $\delta = 25.81$



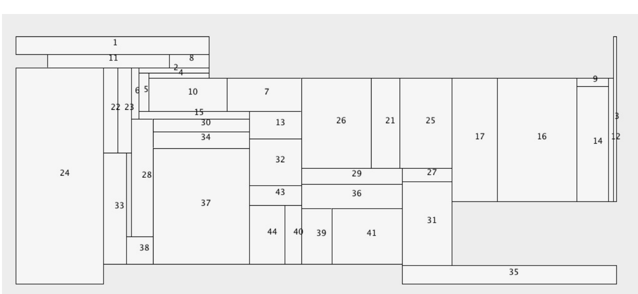
COMB-T: $\overline{\varepsilon} = 0.001$, $\tau = 0$, $\theta = 33.88$, $\delta = 53.14$



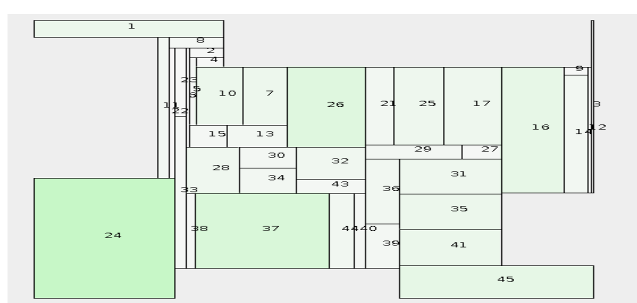
COMB-R: $\overline{\varepsilon} = 0.438$, $\tau = 6.087$, $\theta = 30.57$, $\delta = 53.81$



RECT-R: $\overline{\varepsilon} = 0.345$, $\tau = 0$, $\theta = 13.08$, $\delta = 37.48$



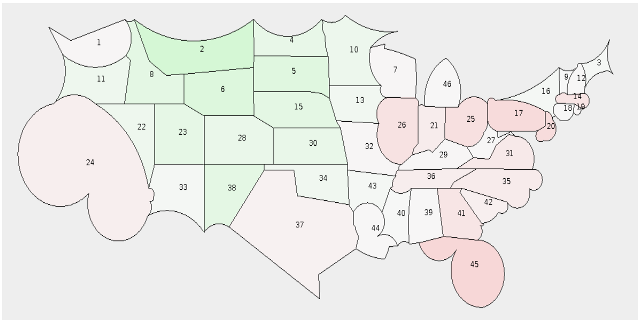
RECT: $\overline{\varepsilon} = 0.0003$, $\tau = 2.82$, $\theta = 16.56$, $\delta = 40.67$



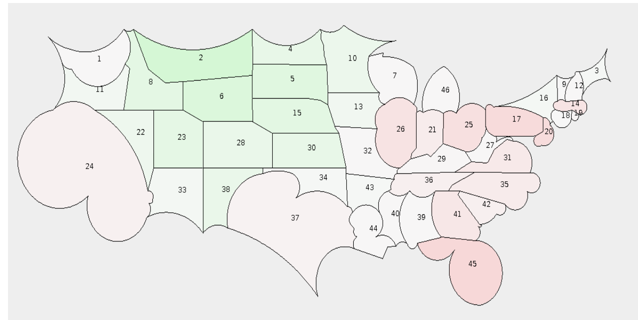
RECT-E: $\overline{\varepsilon} = 0.191$, $\tau = 0$, $\theta = 27.73$, $\delta = 33.88$

Table 6: USA GDP cartograms by different algorithms along with the values for $\overline{\varepsilon}$, τ, θ and δ.

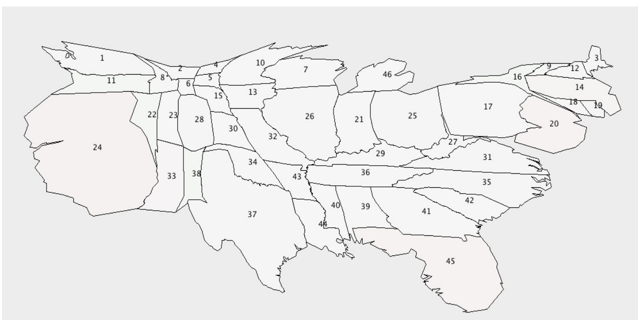
USA Population Cartograms



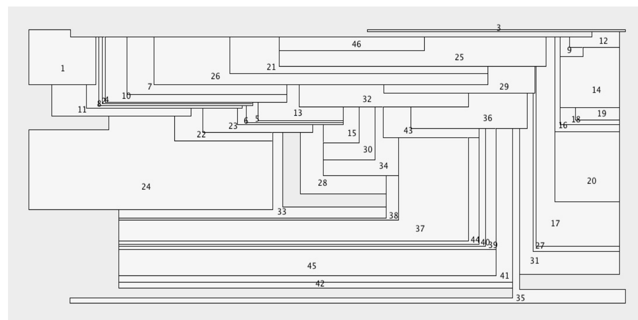
CIRC-S: $\overline{\varepsilon} = 0.405$, $\tau = 0$, $\theta = 1.22$, $\delta = 8.12$



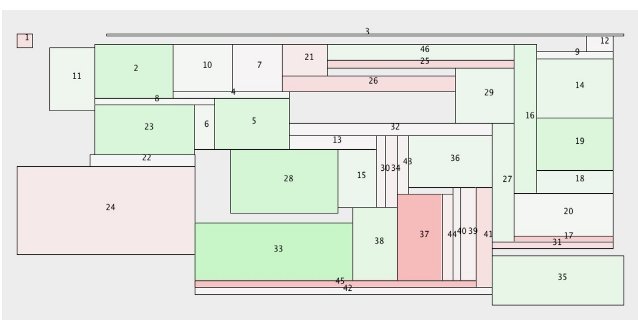
CIRC-W: $\overline{\varepsilon} = 0.386$, $\tau = 0$, $\theta = 1.3$, $\delta = 9.96$



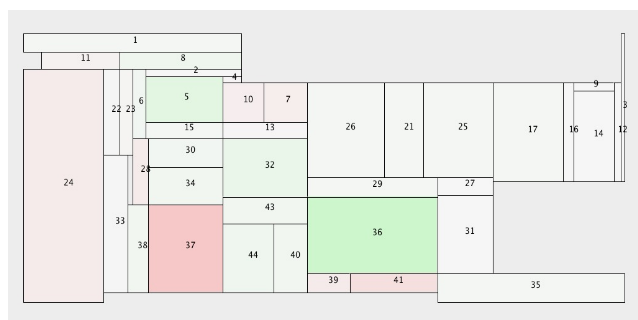
DIF: $\overline{\varepsilon} = 0.081$, $\tau = 0$, $\theta = 6.88$, $\delta = 25.57$



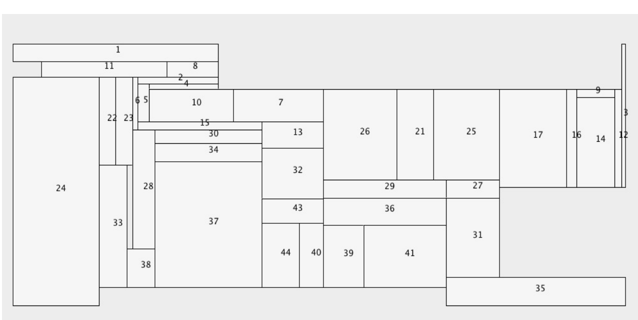
COMB-T: $\overline{\varepsilon} = 0.001$, $\tau = 0$, $\theta = 33.57$, $\delta = 52.16$



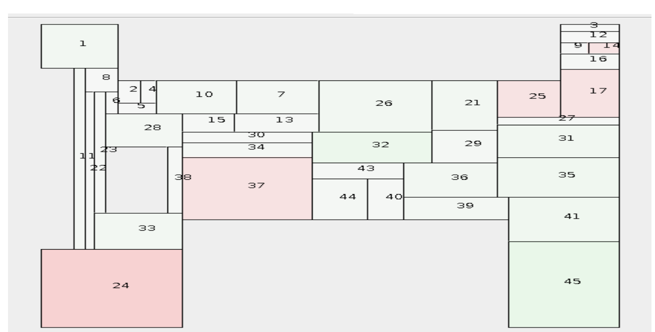
COMB-R: $\overline{\varepsilon} = 0.427$, $\tau = 6.38$, $\theta = 30.51$, $\delta = 51.65$



RECT-R: $\overline{\varepsilon} = 0.277$, $\tau = 0$, $\theta = 14.23$, $\delta = 37.76$



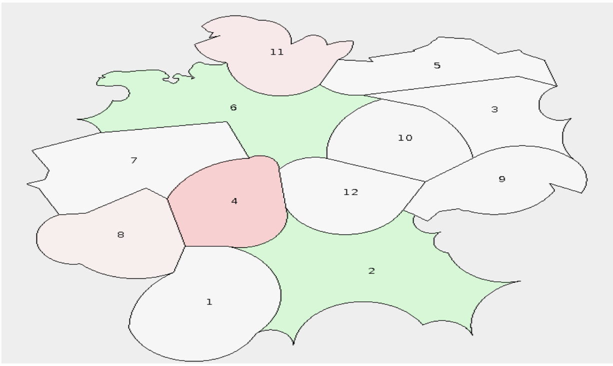
RECT: $\overline{\varepsilon} = 0.0002$, $\tau = 2.56$, $\theta = 16.17$, $\delta = 40$



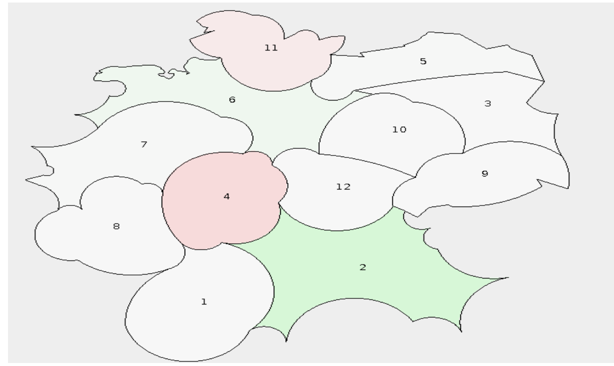
RECT-E: $\overline{\varepsilon} = 0.163$, $\tau = 0$, $\theta = 29.75$, $\delta = 48.79$

Table 7: USA Population cartograms by different algorithms along with the values for $\overline{\varepsilon}$, τ, θ and δ.

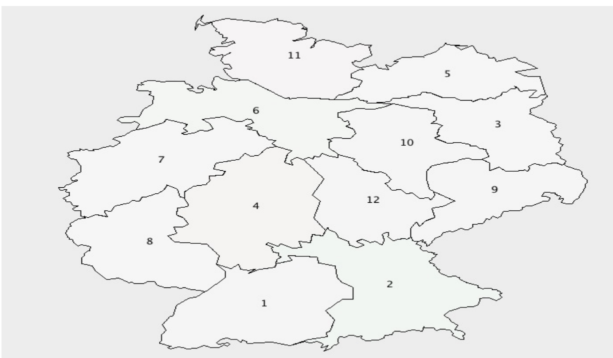
Germany GDP Cartograms



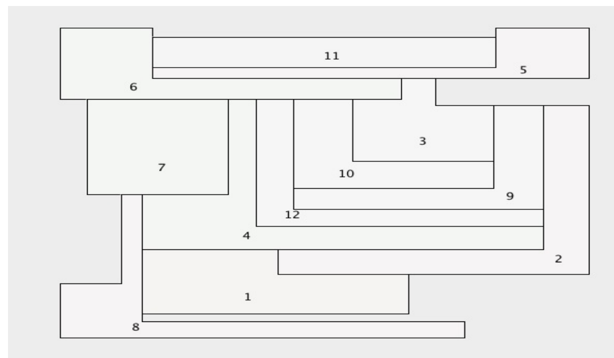
CIRC-S: $\overline{\varepsilon} = 0.103$, $\tau = 0$, $\theta = 0.7$, $\delta = 10.32$



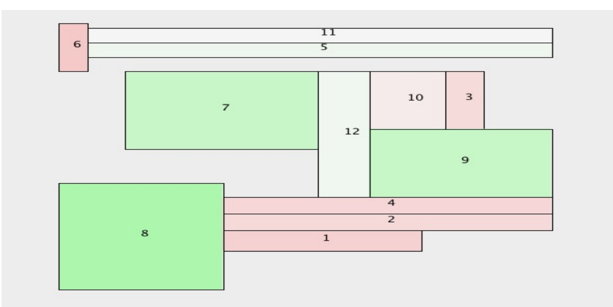
CIRC-W: $\overline{\varepsilon} = 0.067$, $\tau = 0$, $\theta = 0.83$, $\delta = 13.81$



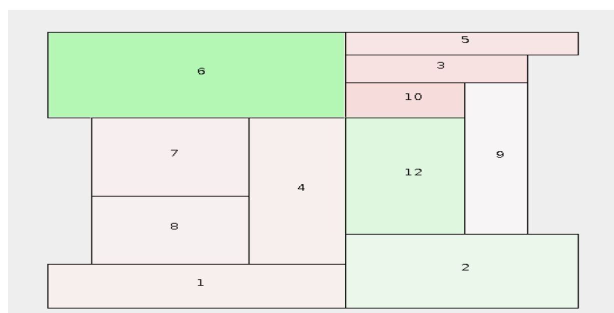
DIF: $\overline{\varepsilon} = 0.013$, $\tau = 0$, $\theta = 0.82$, $\delta = 12.02$



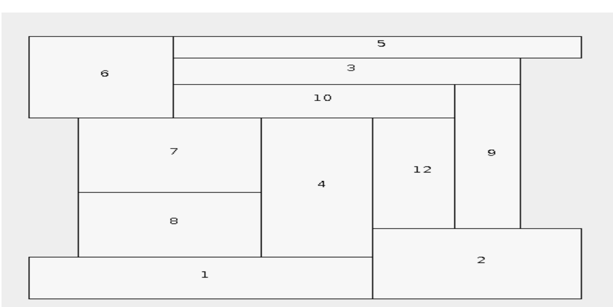
COMB-T: $\overline{\varepsilon} = 0.012$, $\tau = 0$, $\theta = 12.45$, $\delta = 42.81$



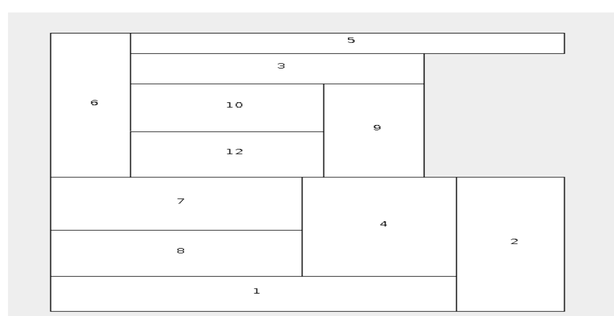
COMB-R: $\overline{\varepsilon} = 0.189$, $\tau = 19.7$, $\theta = 15.42$, $\delta = 42.88$



RECT-R: $\overline{\varepsilon} = 0.266$, $\tau = 0$, $\theta = 7.83$, $\delta = 32.74$



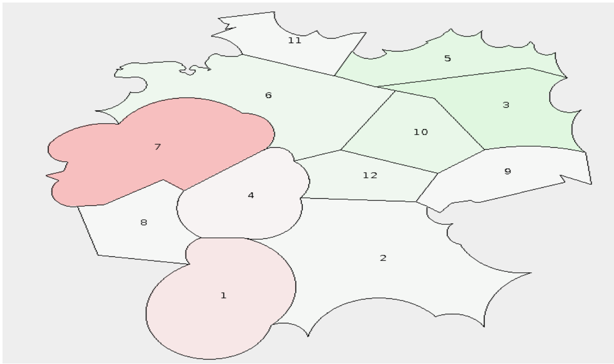
RECT: $\overline{\varepsilon} = 0$, $\tau = 7.27$, $\theta = 11.79$, $\delta = 38.25$



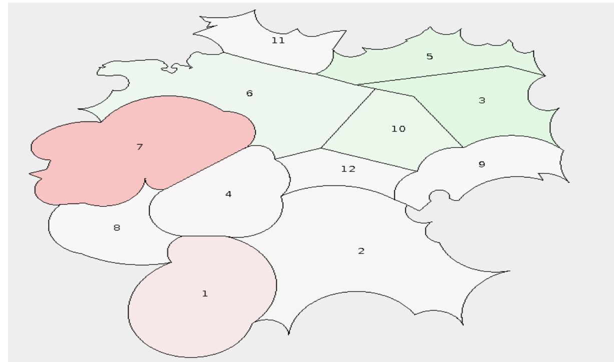
RECT-E: $\overline{\varepsilon} = 0.094$, $\tau = 0$, $\theta = 17.61$, $\delta = 34.14$

Table 8: Germany GDP cartograms by different algorithms along with the values for $\overline{\varepsilon}$, τ, θ and δ.

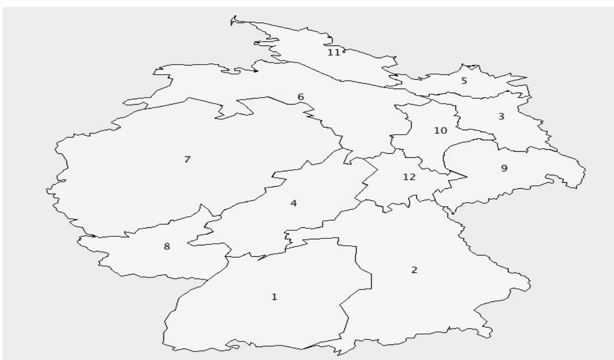
Germany Population Cartograms



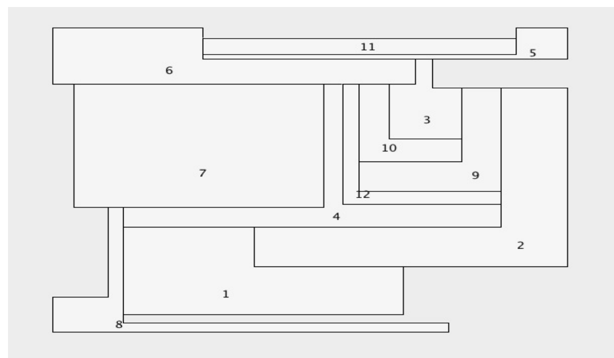
CIRC-S: $\overline{\varepsilon} = 0.238$, $\tau = 0$, $\theta = 0.43$, $\delta = 7.45$



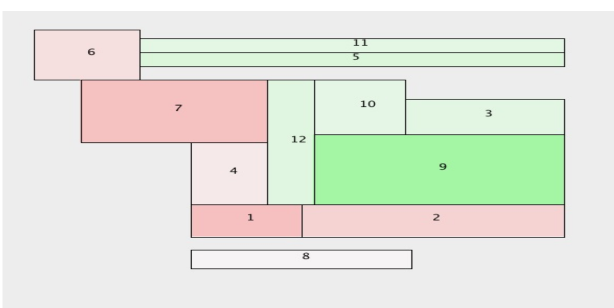
CIRC-W: $\overline{\varepsilon} = 0.209$, $\tau = 0$, $\theta = 0.57$, $\delta = 10.25$



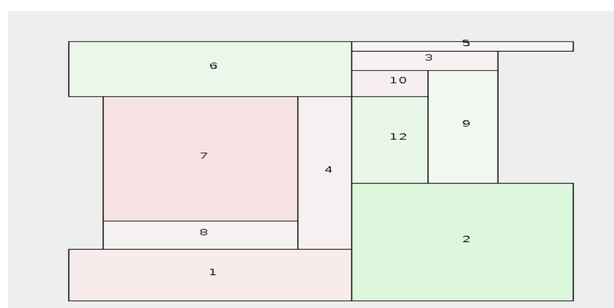
DIF: $\overline{\varepsilon} = 0.02$, $\tau = 0$, $\theta = 1.66$, $\delta = 15.35$



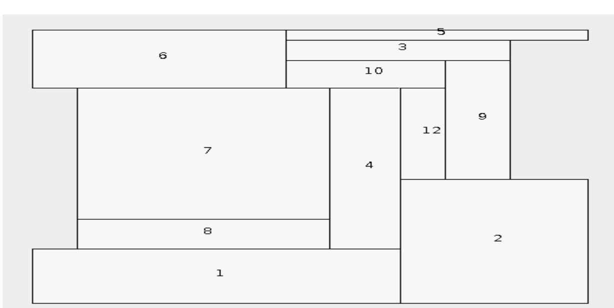
COMB-T: $\overline{\varepsilon} = 0.01$, $\tau = 0$, $\theta = 9.49$, $\delta = 42.7$



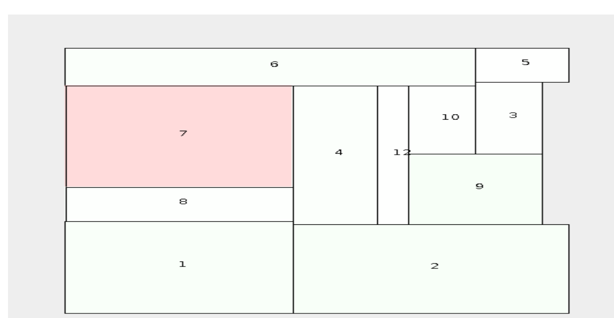
COMB-R: $\overline{\varepsilon} = 0.323$, $\tau = 30.3$, $\theta = 17.19$, $\delta = 42.34$



RECT-R: $\overline{\varepsilon} = 0.235$, $\tau = 0$, $\theta = 16.67$, $\delta = 37.71$



RECT: $\overline{\varepsilon} = 0$, $\tau = 7.27$, $\theta = 18.97$, $\delta = 41.47$



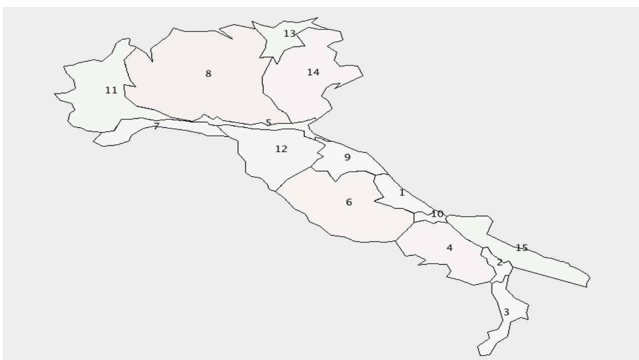
RECT-E: $\overline{\varepsilon} = 0.052$, $\tau = 0$, $\theta = 10.79$, $\delta = 30.21$

Table 9: Germany Population cartograms by different algorithms along with the values for $\overline{\varepsilon}$, τ, θ and δ.

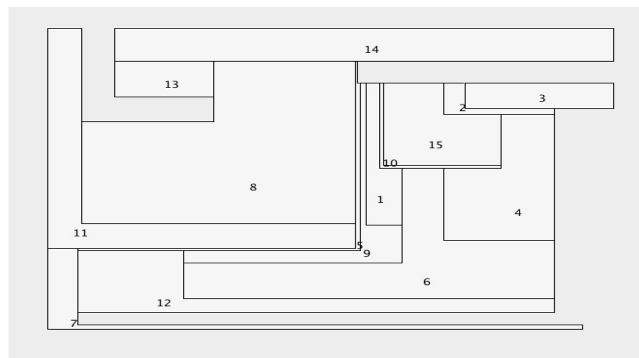
Italy GDP Cartograms

CIRC-S: $\bar{\varepsilon} = 0.184$, $\tau = 0$, $\theta = 0.34$, $\delta = 9.7$

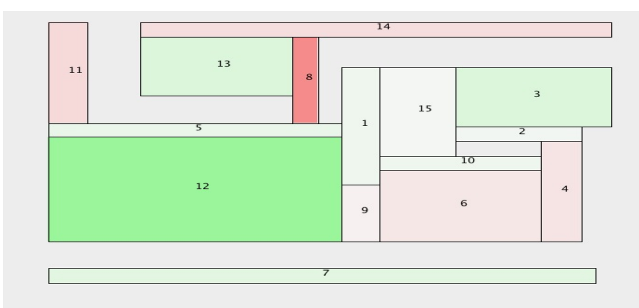
CIRC-W: $\bar{\varepsilon} = 0.162$, $\tau = 0$, $\theta = 0.48$, $\delta = 13.78$



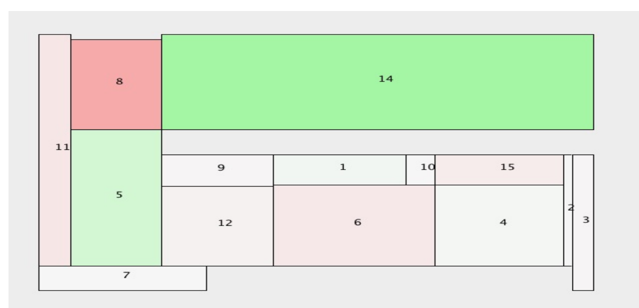
DIF: $\bar{\varepsilon} = 0.085$, $\tau = 0$, $\theta = 1.19$, $\delta = 20.5$



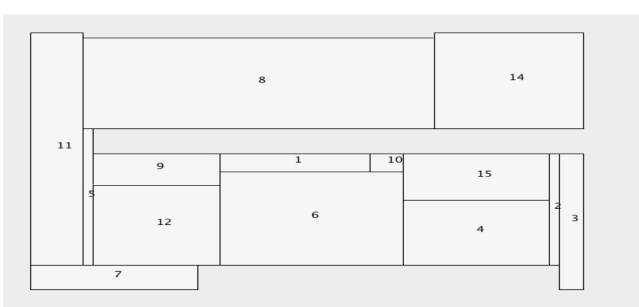
COMB-T: $\bar{\varepsilon} = 0.001$, $\tau = 0$, $\theta = 49.29$, $\delta = 50.18$



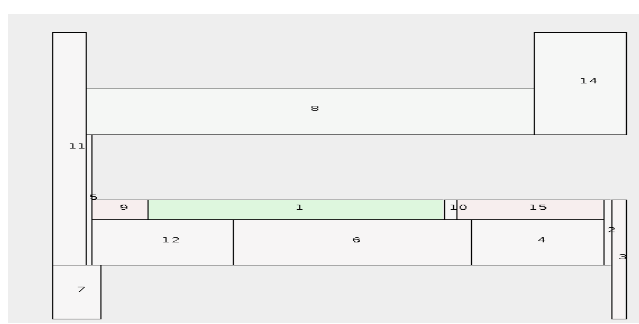
COMB-R: $\bar{\varepsilon} = 0.315$, $\tau = 13.33$, $\theta = 38.81$, $\delta = 56.71$



RECT-R: $\bar{\varepsilon} = 0.33$, $\tau = 0$, $\theta = 38.26$, $\delta = 37.97$



RECT: $\bar{\varepsilon} = 0.0003$, $\tau = 3.3$, $\theta = 35.3$, $\delta = 37.48$

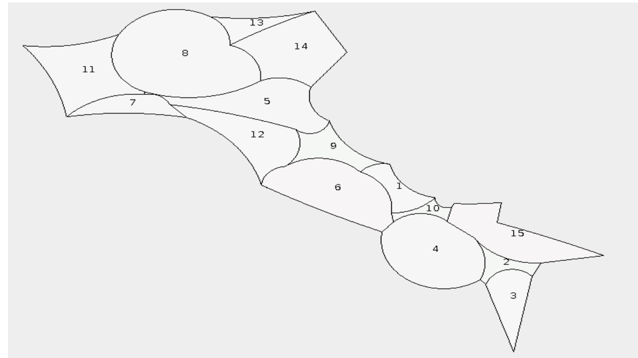


RECT-E: $\bar{\varepsilon} = 0.126$, $\tau = 0$, $\theta = 32.3$, $\delta = 53.88$

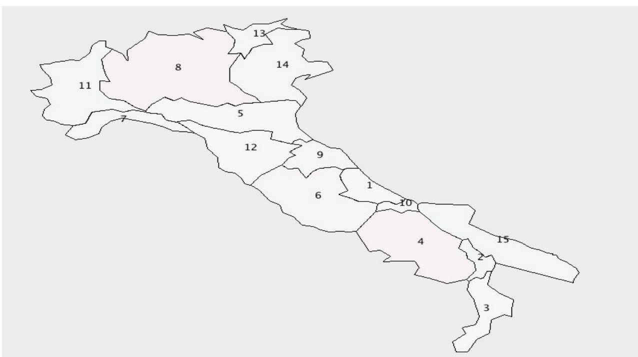
Table 10: Italy GDP cartograms by different algorithms along with the values for $\bar{\varepsilon}$, τ, θ and δ.

Italy Population Cartograms

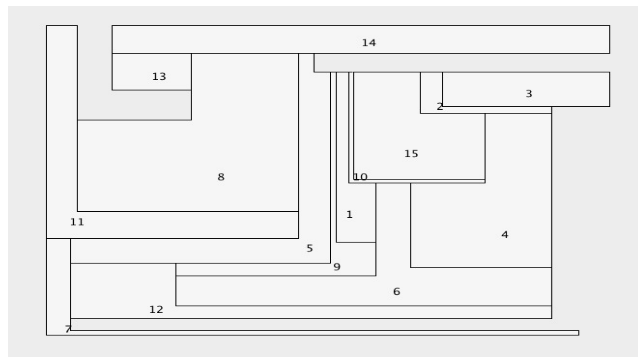
CIRC-S: $\overline{\varepsilon} = 0.085$, $\tau = 0$, $\theta = 0.54$, $\delta = 11.18$



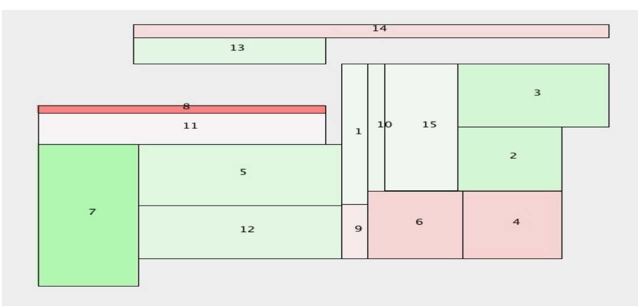
CIRC-W: $\overline{\varepsilon} = 0.054$, $\tau = 0$, $\theta = 0.6$, $\delta = 13.47$



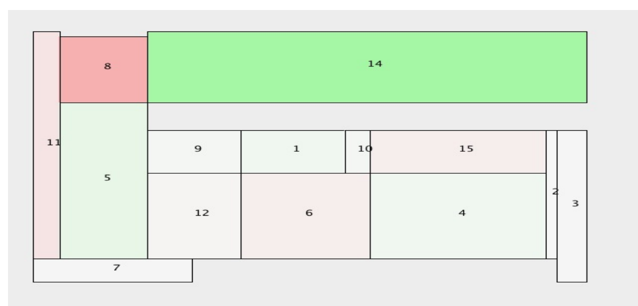
DIF: $\overline{\varepsilon} = 0.056$, $\tau = 0$, $\theta = 0.85$, $\delta = 14.95$



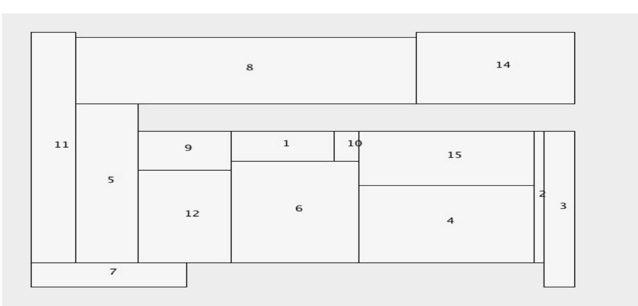
COMB-T: $\overline{\varepsilon} = 0.001$, $\tau = 0$, $\theta = 47.77$, $\delta = 49.11$



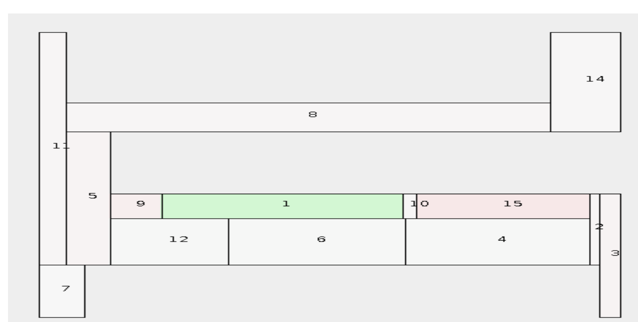
COMB-R: $\overline{\varepsilon} = 0.33$, $\tau = 13.33$, $\theta = 46.25$, $\delta = 43.09$



RECT-R: $\overline{\varepsilon} = 0.241$, $\tau = 0$, $\theta = 40.09$, $\delta = 37.96$



RECT: $\overline{\varepsilon} = 0.0003$, $\tau = 3.3$, $\theta = 35.23$, $\delta = 37.52$



RECT-E: $\overline{\varepsilon} = 0.148$, $\tau = 0$, $\theta = 37.29$, $\delta = 51.19$

Table 11: Italy Population cartograms by different algorithms along with the values for $\overline{\varepsilon}$, τ, θ and δ.

Appendix C

Here we report the running time for different cartogram generation algorithms, using GDP and population on the maps of USA, Germany and Italy.

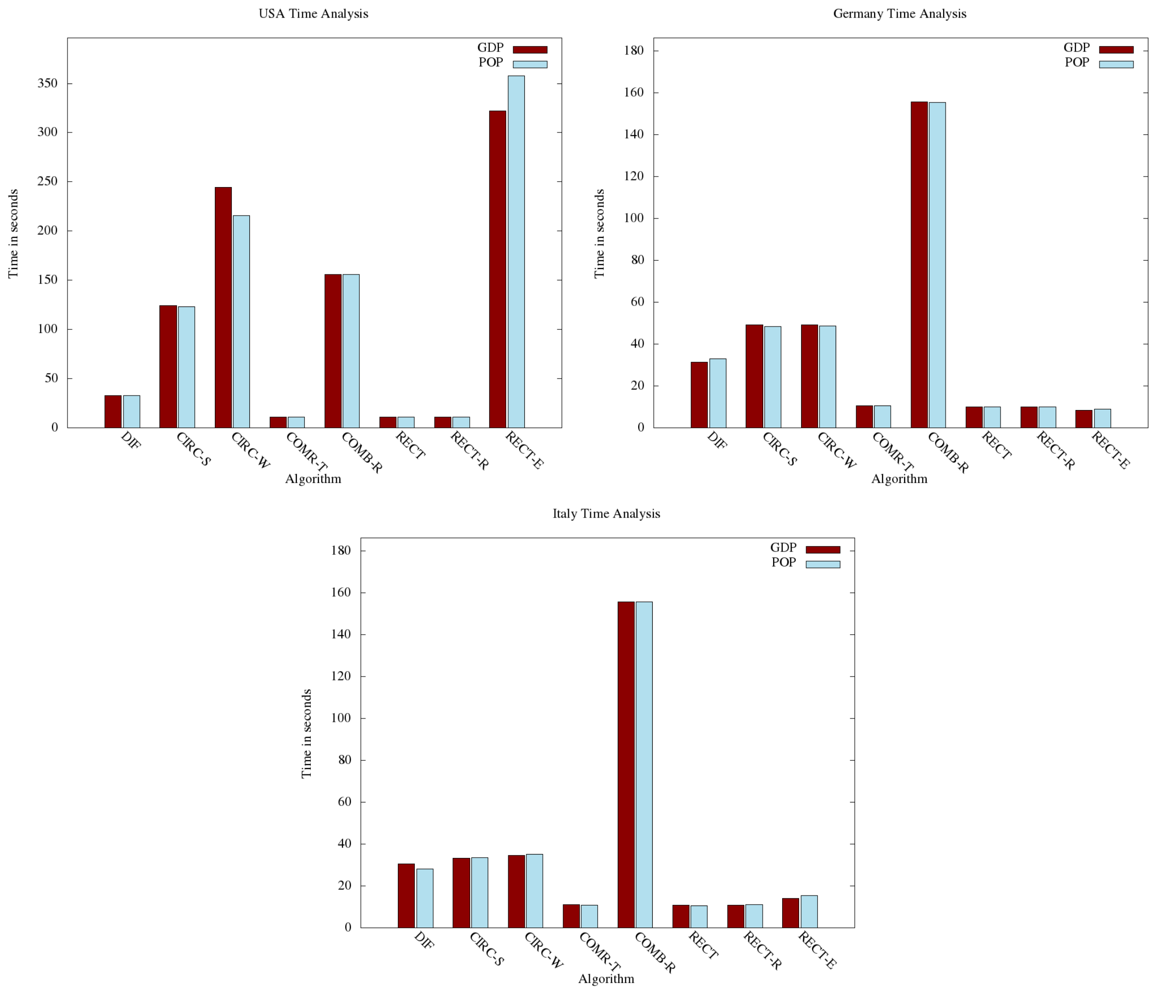


Figure 8: Running time for different algorithms in seconds. For the iterative algorithms, *COMB-R*, *RECT-E*, *CIRC-S* and *CIRC-W*, the running time depends on the number of iteration. The three algorithms *COMB-T*, *RECT* and *RECT-R* were let to run for 10 seconds. For *DIF*, *CIRC-S* and *CIRC-W* algorithms, running time increases as the level of schematization decreases.

Acknowledgments

We thank Jan-Hinrich Kämper and Martin Nöllenburg for helpful discussions and suggestions. We also thank Wouter Meulemans for several schematized maps of Europe, and Bettina Speckmann, Marc van Kreveld, and Mark de Berg for discussions about rectangular cartograms. Finally we acknowledge Arkin, Chew, Huttenlocher, Kedem, and Mitchell for the turning function metric and its implementation in C by E. Ressler.