Circular-Arc Cartograms

Jan-Hinrich Kämper¹, Stephen G. Kobourov², and Martin Nöllenburg³

¹ Universität Oldenburg, Germany
² University of Arizona, Tucson, Arizona, USA
³ Karlsruhe Institute of Technology (KIT), Germany

Abstract. We present a new circular-arc cartogram model in which countries are drawn with circular arcs instead of straight-line segments. Given a geographic map and values associated with each country in the map, the cartogram is a new map in which the areas of the countries represent the corresponding values. In the circular-arc cartogram model straight-line segments can be replaced with circular arcs in order to achieve the desired areas, while the corners of the polygons defining each country remain fixed. The countries in circular-arc cartograms have the aesthetically pleasing appearance of clouds or snowflakes, depending on whether their edges are bent outwards or inwards. This makes it easy to determine whether a country has grown or shrunk, just by its overall shape. We show that determining whether a given map and area-values can be realized with a circular-arc cartogram is an NP-hard problem. Next we describe a heuristic method for constructing circular-arc cartograms, which uses a max-flow computation on the dual of the map, along with a computation of the straight skeleton of the underlying polygonal decomposition. Our method is implemented and produces cartograms that, while not perfectly accurate, achieve the desired areas for many real-world examples.

1 Introduction

A cartogram, or a value-by-area diagram, is a visualization in which the areas of countries in a given map are modified in order to represent a given set of values, such as population, gross-domestic product, or other geo-referenced statistical data. Red-and-blue population cartograms of the United States were often used to illustrate the results in the 2000 and 2004 presidential elections. While geographically accurate maps seemed to show an overwhelming victory for George W. Bush, population cartograms effectively communicated the near 50-50 split, by deflating the rural and suburban central states.

The task in creating a cartogram is to shrink or grow certain regions in a map so that they reflect the set of pre-specified area values. In this paper we study a new circular-arc cartogram model, where circular arcs can be used in place of straight-line segments, and corners of the polygons defining each country remain fixed. Intuitively, a region that grows is inflated and becomes cloud-shaped, whereas a region that shrinks is deflated and becomes snowflake-shaped; see Fig. [1]

There are many design and implementation aspects that determine the effectiveness of a cartogram. Here we consider four of the main aesthetic and computational criteria:

1. It is important that the cartogram is readable, in that it is possible to find every country in the map. Moreover, a readable cartogram makes it possible to visually answer queries about the relative size of the shown countries.
2. It is important to ensure that the cartogram keeps the underlying map structure recognizable. This criterion can be expressed by insisting that the country adjacencies in the original and modified map remain unchanged. An even stronger version of this requirement is to ensure that the relative positions between pairs of countries (e.g., North-South, East-West) are not disturbed.

3. It is important that the cartogram faithfully represents the given weight function. This criterion is often expressed by the cartographic error, defined as the absolute value of the difference between the given weight and the area of a country.

4. The complexity of a cartogram also impacts its effectiveness. Here, the complexity is often measured by the maximum number of vertices (or edges) defining the boundary of any country in the cartogram. Highly schematized cartograms use as few as three or four vertices per country, while geographically accurate and recognizable cartograms may have arbitrarily high complexity.

Circular-arc cartograms ensure readability by keeping the corners of the countries undisturbed and easily convey the type of area changes by the cloud-shape and snowflake-shape of the countries. They are also recognizable as they leave the adjacencies undisturbed and also preserve the relative positions of countries. The complexity is exactly the same as that of the input map: a highly schematized input map directly results in low complexity of the resulting cartogram. These advantages come at a cost: it is possible that a given map with pre-specified areas cannot be realized as a circular-arc cartogram, and determining whether such a realization exists is NP-hard. However, if we are willing to tolerate cartographic errors we can use a heuristic algorithm, which while not perfectly accurate, achieves the desired areas for many real-world examples.

1.1 Related Work

The problem of representing additional information on top of a geographic map dates back to the 19th century, and highly schematized rectangular cartograms can be dated
to the 1930’s work of Raisz [21]. With rectangular cartograms it is not always possible to preserve all country adjacencies and realize all areas accurately [15]. Eppstein et al. studied area-universal rectangular layouts and characterize the class of rectangular layouts for which all area-assignments can be achieved with combinatorially equivalent layouts [13]. If the requirement that rectangles are used is relaxed to allow the use of rectilinear regions then adjacencies and areas can be maintained with 40-sided regions as shown by de Berg et al. [9]. A series of papers reduced the complexity of the regions to 34, 12, and 10 [4, 5, 17].

More general cartograms, without restrictions to rectangular or rectilinear shapes, have also been studied. 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 [11]. Dorling also uses a cell-based approach, where regions acquire new cells, or get rid of cells, until an equilibrium has been achieved, i.e., each region has attained the desired amount of cells [10]. This technique can result in significant distortions, thereby reducing readability and recognizability. Keim et al. define a distance between the original map and the cartogram with a metric based on Fourier transforms, and then use a scan-line algorithm to reposition the edges so as to optimize the metric [18]. Edelsbrunner and Waupotitsch generate cartograms using a sequence of homeomorphic deformations and measure the quality with local distance distortion metrics [12]. Kocmoud and House [16] describe a technique that combines the cell-based approach of Dorling with the homeomorphic deformations of Edelsbrunner et al.

A popular method by Gastner and Newman [14] projects the original map onto a distorted grid, calculated so that cell areas match the pre-defined values. This method relies on a physical model in which the desired areas are achieved via an iterative diffusion process. Flow moves from one country to another until a balanced distribution is reached, i.e., the density is the same everywhere. The cartograms produced this way are readable and have no cartographic error. However, some countries may be deformed into shapes very different from those in the original map, and the complexity of the polygons can increase significantly.

This brief review of related work is woefully incomplete; a fairly recent survey by Tobler [22] provides a more comprehensive overview.

1.2 Our Contributions

Our model combines aspects of existing cartogram types, but at the same time tries to avoid some of the common shortcomings. By pinning the vertices at their input positions and only modifying edge shapes, regions are not displaced and we avoid strong positional distortions that are common in the popular diffusion cartograms. On the other hand, the shapes of the regions are not as severely schematized as in rectangular cartograms and recognizability of characteristic shapes is preserved, at least for moderate area changes. The use of the inflation/deflation metaphor allows to immediately recognize regions with positive/negative area changes.

Our results in this paper are as follows. In Section 2 we formally introduce the circular-arc cartogram model and state the algorithmic problem. In Section 3 we show that the circular-arc cartogram problem is NP-hard. In Section 4 we describe a heuristic
algorithm to compute circular-arc cartograms with low cartographic error. In Section 5 we summarize our results and describe several open problems.

2 Model

Geometrically, a map of countries or administrative regions is a subdivision $S$ of the plane into a set of disjoint regions or faces $\mathcal{F} = \{f_1, \ldots, f_n\}$. In our model we assume that each face is a simple polygon. The topological structure of the map can be described by its face graph or dual graph $G$, which contains a vertex for each face and an edge between adjacent faces. In order to construct a cartogram of $S$, we additionally need to specify a weight vector $t = (t_1, \ldots, t_n)$, where for each $i = 1, \ldots, n$ the value $t_i$ is the target area of face $f_i$ in the cartogram. An accurate cartogram of the input pair $(S, t)$ is a subdivision $S'$ that is homeomorphic to $S$ and in which the area of every face $f_i$ equals its given weight $t_i$.

In this paper, we are interested in the special class of circular-arc cartograms, i.e., cartograms that can be obtained from the input $S$ by bending each polygon edge $e$ of $S$ into a circular arc whose endpoints coincide with the endpoints of $e$. No two circular arcs are allowed to cross, but we may allow that two arcs touch. Bending an edge between two faces $f_i$ and $f_j$ has the effect of transferring a certain area from one face to the other. This exchange of area between faces can be seen as a discrete diffusion process similar to the model of Gastner and Newman [14]. The algorithmic problem in creating a circular-arc cartogram is thus to compute a bending radius for each edge of the input subdivision so that the resulting circular-arc subdivision $S'$ remains topologically equivalent to the polygonal input subdivision $S$ and each face $f_i$ has area $t_i$. We define a bending configuration of $S$ to be an assignment of a bend radius (including radius $r = \infty$ to represent straight-line arcs) to each edge of $S$. A bending configuration is valid if no two circular arcs cross and the input topology of $S$ is preserved.

In real-world maps there are often regions (e.g., oceans or seas) whose target area in the cartogram is unspecified. Our model allows sea faces in $S$ with no specified target area. Note that if there is a single sea face then its target area change is implicitly given by the sum of the target area changes of the other faces.

The Circular-Arc Cartogram (CAC) decision problem then is:

**Problem 1 (CAC).** Given a planar polygonal subdivision $S$ and a weight vector $t$, is there a valid bending configuration so that the resulting subdivision $S'$ is an accurate cartogram, i.e., all face areas comply with $t$?

While the decision version is mainly of theoretical interest, there is also a corresponding optimization version of Circular-Arc Cartogram. Here the algorithmic problem is to compute a bending configuration that minimizes the cartographic error, i.e., the sum of the differences between the target areas and the actual areas of all faces. In Section 3 we show that CAC is NP-hard and in Section 4 we describe a heuristic algorithm that successfully minimizes the cartographic error in practice.
3 NP-hardness

First note that positive as well as negative CAC instances can be constructed easily: Only polygons whose vertices are cocircular can be made arbitrarily small by bending edges; all other polygons have some positive lower bound on their area in a circular-arc cartogram. Hence, for example, no simple non-convex polygon can attain area 0 by replacing straight edges with circular arcs. On the other hand, any subdivision with a target area vector that contains the exact initial face areas is a positive instance.

Theorem 1. Circular-arc cartogram is NP-hard.

Proof. Our reduction is from the NP-complete problem Planar Monotone 3-SAT [8]. This problem is a special variant of the Planar 3-SAT problem [19]: We are given a Boolean formula \( \varphi \), in which every clause consists of three literals. Each clause, however, must be monotone, i.e., it may contain either only positive or only negative literals. The planarity of the formula refers to the planarity of the associated bipartite variable-clause graph \( G_\varphi \) (with a vertex for every clause and variable of \( \varphi \) and an edge between a variable vertex and a clause vertex if and only if the variable appears in the clause). It is known that for every instance of Planar Monotone 3-SAT the graph \( G_\varphi \) can be drawn in a planar rectilinear fashion by placing the variable vertices on a horizontal line, the positive clauses above that line, and the negative clauses below; see Fig. 2.

Our reduction constructs a subdivision \( S_\varphi \) for the Boolean formula \( \varphi \) that resembles the general structure of the rectilinear drawing of \( G_\varphi \). The weight vector \( t_\varphi \) is chosen so that \( S_\varphi \) can be transformed into a valid circular-arc cartogram if and only if \( \varphi \) is satisfiable. The subdivision consists of three types of gadgets: the variable, literal, and clause gadgets, which we describe below.

A basic building block in all three gadgets is a triangle with target area 0. It is easy to verify that there are exactly three configurations that realize a 0-area circular-arc triangle, all of which consist of circular arcs of the unique circle defined by the three points; see Fig. 3. This is used to control the possible shapes of regions in the cartogram.

Variable gadget. The variable gadget consists of a horizontal row of rectangles with height 4 and width 2, except for some taller rectangles in between of height 5 that serve as connectors to the literal gadgets; see Fig. 4. With the exception of the connector rectangles, all rectangles are enclosed on their two short sides by skinny triangles with a base side of length 2. These triangles have target area 0. They are designed so that...
two of the three possibilities to achieve area 0 would require edges to become circular arcs that pass beyond some of the input vertices of the rectangles. Hence only a single configuration remains feasible. This immediately fixes the shape of the rectangles’ short edges by bending them slightly outward and extends the area of each rectangle by two circular segments. We define the area of the circular segment thus attached to each rectangle as $c_1$. We also need a scaled-down version of this triangle with base length 1 instead of 2 whose corresponding circular segment thus has area $c_2 = c_1 / 4$.

There is one special decision rectangle (purple) in the center of the gadget. The target area change of this rectangle is set to $2c_1 - 2\pi$, where $2\pi$ is the area of a half-circle with radius 2. All other rectangles of height 4 have a target area change of $2c_1$, i.e., they can be extended by the two circular segments at their short sides but otherwise want to keep their area constant. Finally, the taller connector rectangles (which actually consist of six vertices) are adjacent to a literal gadget on one of their short sides (indicated by dots in Fig. 4) and to a right triangle on the other short side. This triangle has target area 0, but other than the skinny triangles described before, all three possible 0-area configurations in Fig. 3 are feasible. The area change of the connector rectangles is $2c_2$, the area gained from the two small skinny triangles adjacent to the left and right sides of the short part that sticks out of the variable row.

Let us consider the purple decision rectangle in the center, with its two short edges fixed by the shape of the attached skinny triangles. If one of its long edges is bent inside the rectangle as exactly a half-circle and the opposite edge remains a straight-line segment, then the specified area constraint is satisfied. It is, however, geometrically impossible to achieve the given target area by bending both edges simultaneously inside the rectangle like in a concave lens. Hence we can use the two possible configurations of the decision rectangle to encode the two truth values of the variable; see Fig. 4(b) and (c). Since by pulling one long edge inside the rectangle the area of the adjacent rectangle in the gadget enlarges, that adjacent rectangle must in turn pull the opposite long edge inwards by the same amount. So the semi-circle arcs propagate, similar to negative air pressure in a physical model, on one side of the gadget, namely that side whose connecting literals evaluate to false in the current state. It remains to describe the behavior of the connector rectangles. Since the long edges are bent into half-circles and no two edges of the subdivision may cross, the right triangle attached to the connector rectangle must be in the state that forms a half-circle and thus increases the area of the connector rectangle. In order to balance this area increase, the opposite short edge must be bent inwards and form an identical half-circle. This gives us a means to transmit the negative pressure from the center of the variable towards all literals that evaluate to false.
There is no negative pressure on the positive side of the variable gadget, i.e., the side whose literals evaluate to true. Hence the long edges of the rectangles on this side can remain straight, and there are two possible configuration for the short edges of each connector rectangle, one of which pushes the half-circles towards the literal gadgets rather than pulling them away as it is the case on the negative side.

**Literal gadget.** The main task of the literal gadget is to maintain and transmit the truth state that is found at the variable gadget towards the clause gadget. The gadget can be seen as a pipe composed of chains of rectangles that connects variable and clause. In the pipe the truth value is transmitted by pulling or pushing the long edges of the rectangles into half-circles similarly as in the variable gadget. Three literal gadgets are depicted (together with a clause gadget) in Fig. 5.

There is one notable difference from the transmission of the truth value in the variable gadget since two of the incoming literals for each clause must make a turn of 90\(^\circ\). The turn is realized by a square of side length 2 with target area 4 and two right triangles with target area 0 (as those in the connector rectangles of the variable gadget). Since the two right triangles are placed on adjacent sides of the square, one of them must bend to the outside of the square while the other one must bend to the inside. If the literal is in state false (Fig. 5(b)) and the half-circles are pulled towards the variable gadget, then both the left and right edges of the square are bent inward while the top and bottom
edges are bent outward. This is exactly what is needed to transmit negative pressure to the horizontal part of the literal gadget. For a literal in state \textit{true} we observe exactly the opposite behavior. Fig. 5(c) shows a \textit{true} literal on the left and a \textit{false} literal on the right.

\textit{Clause gadget.} The clause gadget consists of a cross-shaped rectilinear polygon joining the three incoming literal gadgets; see Fig. 5. In its top part there are three right triangles with target area $0$. The target area increase of the clause polygon is $8c_2$, the area increase caused by the eight skinny triangles attached to some of its edges. Note that of the three right triangles at most two can simultaneously bend as half-circles inside the polygon, while they all can bend to the outside independently. As long as one of the incoming literals is \textit{true}, i.e., it pushes a half-circle inside the clause polygon, the three triangles in the top part can balance the area change of the clause polygon caused by any other combination of the remaining two literals; see Fig. 5(c). However, if all three literals are \textit{false}, the area of three half-circles is added to the clause region (indicated by dotted line segments). Consequently, the area of three half-circles must be removed from the clause region, but at most two half-circles can be removed by the right triangles; see Fig. 5(b). This shows that the area requirement of the clause polygon can be realized if and only if the clause evaluates to \textit{true} in the given truth assignment.

\textit{Reduction.} From the construction of the gadgets it follows that if the Boolean formula $\varphi$ has a satisfying variable assignment, then the subdivision $S_\varphi$ and the weights $t_\varphi$
are a positive instance of CIRCULAR-ARC CARTOGRAM. On the other hand, we can immediately obtain a satisfying truth assignment for the variables of \( \varphi \) from a valid circular-arc cartogram of \( S_\varphi \). The vertices of the subdivision \( S_\varphi \) all lie on a grid of polynomial size and the target weights are either 0 or can be encoded algebraically in polynomial space. This concludes the proof.

\[\square\]

4 Heuristic Method for Computing Circular-Arc Cartograms

Here we describe a heuristic method for generating circular-arc cartograms based on network flows and polygonal straight skeletons.

Recall that in our model a map is a subdivision \( S \) of the plane into a set of disjoint regions, or faces, \( F = \{f_1, \ldots, f_n\} \), where each face is a simple polygon. The topological structure of the map is described by its face graph, or dual graph \( G \), which contains a vertex \( v_i \) for each face \( f_i \) and an edge \( \{v_i, v_j\} \) between adjacent faces \( f_i \) and \( f_j \). Here we convert \( G \) into a directed graph: for any two adjacent countries in \( S \) the corresponding vertices in \( G \) are connected with two edges, one for each direction.

The initial face areas are described by the vector \( a = (a_1, \ldots, a_n) \) and the target areas are given by the vector \( t = (t_1, \ldots, t_n) \). Without loss of generality, we can assume that both vectors are normalized, i.e., \( \sum_{i=1}^{n} a_i = \sum_{i=1}^{n} t_i = 1 \). This means that the total area of the map remains the same. From \( a \) and \( t \) we can obtain the vector \( \Delta = (\Delta_1, \ldots, \Delta_n) \) of desired area changes, where \( \Delta_i = t_i - a_i \) for each \( i = 1, \ldots, n \).

The goal of our algorithm is to compute a valid bending configuration in which the resulting face areas \( b = (b_1, \ldots, b_n) \) are as close to the given target areas \( t \) as possible. More precisely, we aim to minimize the error \( \sum_{i=1}^{n} |b_i - t_i| \).

4.1 A Network Flow Model for Circular-Arc Cartograms

We use the directed face graph \( G \) to define a flow network in which the flow along an edge \( e = (u, v) \) corresponds to the area exchange from the face vertex \( u \) to the face vertex \( v \). We define the capacity \( c(e) \) to be equal to the area that can be safely transferred from \( u \) to \( v \). To compute valid capacities we use the geometry of the face polygons that specify the countries.

The straight skeleton of a simple \( m \)-edge polygon, \( P \), is made of straight-line segments and partitions the interior of \( P \) into \( m \) disjoint regions, each corresponding to exactly one edge of \( P \) \[3\]. The straight skeleton is similar to the medial axis but does not require parabolic curves and can be efficiently computed in subquadratic time \[7\]. Because the straight skeleton partitions a polygon into disjoint regions, we can define a “safe” bending limit for each edge of the polygon by requiring that the circular arcs remain inside their skeleton regions; see Fig. 6. This guarantees that no two circular arcs cross. For each edge \( e = (u, v) \) we can thus define the capacity \( c(e) \) as the maximally transferrable area from face \( u \) to face \( v \) subject to the constraint that every circular arc on the boundary between \( u \) and \( v \) remains inside its skeleton regions. We note that the capacities are by definition static and independent of each other.
Fig. 6: The straight skeleton of two adjacent polygons and the maximally realizable circular-arcs within the safe bending limits of each edge.

Now, we create a new vertex $v'_i$ for every vertex $v_i$ in $G$. If $\Delta_i > 0$ we make $v'_i$ a source vertex and add the edge $(v'_i, v_i)$ with capacity $c(v'_i, v_i) = \Delta_i$ to $G$; otherwise if $\Delta_i < 0$ we make $v'_i$ a sink vertex and add the edge $(v_i, v'_i)$ with capacity $c(v_i, v'_i) = -\Delta_i$ to $G$. Let $S$ be the set of sources and $T$ the set of sinks.

The quadruple $\mathcal{N} = (G, c, S, T)$ now forms a multiple-source multiple-sink flow network, which is planar since the original face graph of $S$ was planar. If a maximum flow in $\mathcal{N}$ with a value of $D = \sum_{\Delta_i > 0} \Delta_i$ can be found we know that all target areas can be achieved. Furthermore, even if the maximum flow has a value of less than $D$, it still minimizes the cartographic error $\sum_{i=1}^{n} |b_i - t_i|$ under the given safety constraints for the circular arcs.

The expected running time for computing the straight skeleton of a $k$-vertex polygon is $O(k \log^2 k)$ [7]. So if the input subdivision $S$ consists of $n$ faces with $N$ vertices in total, we can compute the straight skeletons in $O(N \log^2 N)$ expected time in total. To solve the multiple-source multiple-sink maximum-flow problem in our flow network based on the planar face graph of $S$ we can use the $O(n \log^3 n)$-time algorithm of Borradaile et al. [6].

4.2 Implementation and Results

We implemented our method in C++ using the CGAL library [2] for computing the straight skeletons and Boost [1] for solving the max-flow problem. In this section we present two circular-arc cartograms produced with our implementation and further examples are included in the appendix. For input subdivisions we used schematized maps, graciously provided by Wouter Meulemans and generated with his algorithm for area-preserving subdivision schematization [20].

Figure 7a shows a circular-arc cartogram for the 2010 gross domestic product (GDP) in Europe. The regions in the output are all recognizable. We can easily identify the different countries as the overall shapes are not very distorted. Even the aspect ratio of most regions remains mostly unchanged. The length of all borders is at least as big as in the input, so we obtained an improved readability of adjacencies.

4 2010 GDP data from [http://www.imf.org](http://www.imf.org)
On the other hand, the cartographic error for some regions is high. Groups of landlocked countries, such as Switzerland, Luxembourg and Belgium, that all need to increase (decrease) their sizes pose significant difficulties. While sea-adjacency helps, it does not always suffice to reach the target area. The major European economies, e.g., Germany, France and the UK, need to further increase their areas but eventually run into each other. This example suggest that for cartograms with large area changes it might be useful to allow some vertex movement in order to decrease cartographic error.

Figure 7b shows a cartogram for the population distribution in Italy. This is an example of a map where our algorithm performs much better. In this map most regions have access to the external sea face where the maximum size of circular arcs is less restricted. Moreover, the desired area changes are relatively moderate. With the removal of a few degree-2 vertices, i.e., a further simplification of the input subdivision, we could improve the cartogram even more.

In summary, both examples show that circular-arc cartograms in the current model yield visually appealing shapes but for certain configurations in the input further modifications to the model and the algorithm are required in order to lower the remaining cartographic error.

5 2010 population data from http://demo.istat.it/pop2010.
5 Conclusions and Future Work

In this paper we introduced circular-arc cartograms as a new model for value-by-area diagrams. We showed that the CIRCULAR-ARC CARTOGRAM problem is NP-hard and presented a heuristic algorithm to produce valid circular-arc cartograms with low cartographic error. The results from our implementation indicate that circular-arc cartograms are readable, recognizable, have low complexity and are visually appealing. While for many real-world examples the cartographic error is small, this cannot be guaranteed and in some cases the error can be large, due to input polygon shapes and their distance to the sea.

There are several natural directions for future work. First we note that the potential area change for each edge depends on its input length. Thus the fewer and longer the edges of a face boundary are, the larger is the range of realizable areas of the face. While we assumed that a fixed schematized subdivision is given as input, we can also allow further schematization (simplification) on demand, i.e., the larger the required area changes the more polygonal vertices may be discarded in order to create longer and fewer edges.

Second, while it is generally undesirable to displace regions, it is often possible to obtain lower cartographic errors by displacing only a few boundary vertices. It is natural to consider the trade-off between minimizing the overall cartographic error and minimizing overall vertex movement.

Third, one of the appealing features of circular-arc cartograms is the cloud-like shape of countries that have increased area and snowflake-shape of countries with decreased area. Generalizing circular arcs to other types of smooth curves, e.g., cubic splines, may result in visually similar cartograms which allow for more flexibility and lower error.

References

1. Boost C++ Libraries. http://www.boost.org.
2. CGAL, Computational Geometry Algorithms Library. http://www.cgal.org.
3. O. Aichholzer, F. Aurenhammer, D. Alberts, and B. Gärtner. A novel type of skeleton for polygons. J. Universal Computer Science, 1(12):752–761, 1995. doi:10.3217/jucs-001-12-0752.
4. M. J. Alam, T. Biedl, S. Felsner, A. Gerasch, M. Kaufmann, and S. G. Kobourov. Linear-time algorithms for proportional contact graph representations. In Proc. 22nd International Symposium on Algorithms and Computation (ISAAC’11), volume 7074 of Lecture Notes Comput. Sci., pages 281–291. Springer-Verlag, 2011. doi:10.1007/978-3-642-25591-5_30.
5. T. C. Biedl and L. E. R. Velázquez. Orthogonal cartograms with few corners per face. In Proc. 12th International Symposium on Algorithms and Data Structures (WADS’11), volume 6844 of Lecture Notes Comput. Sci., pages 98–109. Springer-Verlag, 2011. doi:10.1007/978-3-642-22300-6_9.
6. G. Borradaile, P. N. Klein, S. Mozes, Y. Nussbaum, and C. Wulff-Nilsen. Multiple-source multiple-sink maximum flow in directed planar graphs in near-linear time. CoRR, abs/1105.2228, 2011. To appear in FOCS 2011. arXiv:1105.2228.
7. S.-W. Cheng and A. Vigneron. Motorcycle graphs and straight skeletons. Algorithmica, 47(2):159–182, 2007. doi:10.1007/s00453-006-1229-7.
8. M. de Berg and A. Khosravi. Optimal binary space partitions in the plane. In Proc. 16th Int’l Conf. Computing and Combinatorics (COCOON’10), volume 6196 of Lecture Notes Comput. Sci., pages 216–225. Springer-Verlag, 2010. [doi:10.1007/978-3-642-14031-0_25]

9. M. de Berg, E. Mumford, and B. Speckmann. On rectilinear duals for vertex-weighted plane graphs. Discrete Mathematics, 309(7):1794–1812, 2009. [doi:10.1016/j.disc.2007.12.087]

10. D. Dorling. Area cartograms: their use and creation. Number 59 in Concepts and Techniques in Modern Geography. University of East Anglia: Environmental Publications, 1996. Available from: [http://www.dannydorling.org/?page_id=1448](http://www.dannydorling.org/?page_id=1448).

11. 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.

12. H. Edelsbrunner and R. Waupotitsch. A combinatorial approach to cartograms. Comput. Geom. Theory Appl., 7(5-6):343–360, 1997. [doi:10.1016/S0925-7721(96)00006-5]

13. D. Eppstein, E. Mumford, B. Speckmann, and K. Verbeek. Area-universal rectangular layouts. In Proc. 25th ACM Symposium on Computational Geometry (SoCG’09), pages 267–276. ACM, 2009. [doi:10.1145/1542362.1542411]

14. M. T. Gastner and M. E. J. Newman. Diffusion-based method for producing density-equalizing maps. Proceedings of the National Academy of Sciences of the United States of America, 101(20):7499–7504, 2004. [doi:10.1073/pnas.0400280101]

15. R. Heilmann, D. A. Keim, C. Panse, and M. Sips. Recmap: Rectangular map approximations. In Proc. IEEE Symposium on Information Visualization (InfoVis ’04), pages 33–40. IEEE Computer Society, 2004. [doi:10.1109/INFVIS.2004.57]

16. D. H. House and C. J. Kocmoud. Continuous cartogram construction. In Proc. Conf. Vis. (VIS ’98), pages 197–204, 1998. [doi:10.1109/VISUAL.1998.745303]

17. A. Kawaguchi and H. Nagamochi. Drawing slicing graphs with face areas. Theor. Comput. Sci., 410(11):1061–1072, 2009. [doi:10.1016/j.tcs.2008.10.038]

18. D. A. Keim, S. C. North, and C. Panse. Cartodraw: A fast algorithm for generating contiguous cartograms. IEEE Trans. Vis. Comput. Graph., 10(1):95–110, 2004. [doi:10.1109/TVCG.2004.1260761]

19. D. Lichtenstein. Planar formulae and their uses. SIAM J. Computing, 11(2):329–343, 1982. [doi:10.1137/0211025]

20. W. Meulemans, A. van Renssen, and B. Speckmann. Area-preserving subdivision schema- tization. In Proc. Geographic Information Science (GIScience ’10), volume 6292 of Lecture Notes Comput. Sci., pages 160–174. Springer-Verlag, 2010. [doi:10.1007/978-3-642-15300-6_12]

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

22. W. Tobler. Thirty five years of computer cartograms. Annals Assoc. American Geographers, 94(1):58–73, 2004. [doi:10.1111/j.1467-8306.2004.09401004.x]

23. M. van Kreveld and B. Speckmann. On rectangular cartograms. Comput. Geom. Theory Appl., 37(3):175–187, 2007. [doi:10.1016/j.comgeo.2006.06.002]
Appendix: Further Examples

Figure 8 shows a cartogram for the population distribution in the Netherlands in 2004. The Netherlands are quite unevenly populated: for example the three provinces of Noord-Brabant, Zuid-Holland and Noord-Holland (containing all important urban areas), contribute more than half of the Dutch population. This imbalance between south and north can be seen well in the cartogram. The regions of the metropolitan south are cloud-shaped, while the northern rural areas are more snowflake-shaped.

The type of input schematized map plays a significant role in the computation of circular cartograms. The example in Figure 7 shows a cartogram of Europe by GDP starting with a rectilinear schematization, while the somewhat similar map in Figure 7a showed a cartogram of Europe by GDP, starting with an octilinear schematization. Figure 10 shows a cartogram of the US based on agricultural exports and Figure 10 shows a cartogram of the US based on GDP.
Fig. 9: Europe GDP
Fig. 10: USA agricultural exports

Fig. 11: USA GDP