Karlsruhe Institute of Technology

Institut für Theoretische Informatik
Lehrstuhl für Algorithmik I

Cartograms and Circular-Arc Simplification of Polygonal Subdivisions

Diplomarbeit
von

Jan-Hinrich Kämper

an der Fakultät für Informatik

Erstgutachter:	Prof. Dr. Dorothea Wagner
Zweitgutachter:	Prof. Dr. Peter Sanders
Betreuende Mitarbeiter:	Dr. Martin Nöllenburg Prof. Stephen Kobourov

Bearbeitungszeit: 14. März 2011 – 13. September 2011

KIT – Universität des Landes Baden-Württemberg und
nationales Forschungszentrum in der Helmholtz-Gemeinschaft

www.kit.edu

Hiermit versichere ich, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Hilfsmittel verwendet habe.

Karlsruhe, den 13. September 2011

.....

Jan-Hinrich Kämper

Deutsche Zusammenfassung

Kartogramme sind zu einem Gewichtsvektor proportionale Zeichnungen von Landkarten bei denen die Flächeninhalte der Länder genau den vorgegebenen Gewichten entsprechen. Neben der Proportionalität spielt bei der Generierung von Kartogrammen die Qualität der Darstellung – und damit die Lesbarkeit – eine entscheidende Rolle. Die Länder dürfen nicht zu stark deformiert oder verschoben werden, damit die ursprünglichen Konturen sichtbar bleiben und die einzelnen Länder aufgrund ihrer Form schnell wiedererkannt werden können.

In dieser Arbeit untersuchen wir einen Ansatz zur Erstellung von Kartogrammen durch Krümmung von Ländergrenzen. Genauer erlauben wir die Ersetzung von geraden Kanten durch Kreisbögen, was offensichtlich einen Flächentransfer zwischen den beteiligten Ländern bewirkt. Die Regionen in den auf diese Weise generierten Kartogrammen bekommen durch die geschwungenen Konturen dann eine wolkenähnliche beziehungsweise flockenähnliche Form je nachdem, ob sie ihre Fläche vergrößern oder verringern. Neben einer formalen Definition mehrerer Problemvarianten und dem Nachweis der NP-Schwere all dieser Varianten schlagen wir verschiedene Relaxierungen der Bedingungen an ein korrektes Kartogramm vor mit dem Ziel, das Problem unter diesen vereinfachten Bedingungen effizient lösen zu können. Hauptansatzpunkt hierbei sind die Lockerung der strikten Flächenbedingung sowie die Lockerung der Knotenbedingungen. Abschließend stellen wir eine Umsetzung dieser Relaxierungen als Heuristik vor, die auf der Berechnung von maximalen Netzwerkflüssen basiert. Diese Heuristik wurde implementiert und eine Bewertung anhand der resultierenden Kartogramme vorgenommen.

Abstract

Cartograms are proportional drawings of maps where the areas of all countries correspond to prescribed weights. Besides the property of being a proportional representation it is essential for the quality of a cartogram that the countries are not too badly deformed. A good clarity can only be achieved if all countries can easily be identified by the reader due to the fact that their principal topological properties have been carried over from the original map to the cartogram.

In this thesis we investigate an approach to generate cartograms based on a deformation of borders between different countries. In particular, we allow to replace straight line segments with circular arcs in order to transfer area from one country to another. The countries in the resulting cartograms look thus a bit like clouds or flakes, depending on whether the edges were bent outwards or inwards. We introduce several problem variants and show that they are all NP-hard. Then we propose some relaxations aiming to simplify the problem so that it can be resolved efficiently. First a relaxation of the area constraint is considered where a deviation of the attained area from the prescribed area can be tolerated. Then we suggest relaxations of the vertex-related constraints. A heuristic based on the area relaxation and using a max-flow algorithm is then presented. This heuristic has been implemented and an evaluation has been conducted by showcasing some visual output from the implementation.

Contents

1. Introduction

Automated computational approaches for the visualization of geographic data are becoming increasingly important as modern information systems and the Internet evolve. Cartographers have made a lot of effort in order to represent and manipulate geo-information digitally. Digital maps can now be found anywhere where a few decades ago people used paper maps or atlases. It is not only the fact that they are digital and therefore more easily handleable which accounts for their success but also the possibility to provide them with features which make our lives more convenient. Algorithms that run on digital maps or other geographic data are today crucial in many fields, e.g. communications, route-planing, meteorology, aviation, sea-navigation, logistics, astrology etc. Digital maps and their manipulation have become indispensable and a huge amount of applications rely on elaborated algorithms which operate in real-time and provide the user with any kind of information one could think of.

In this context the schematization and redrawing of maps plays a significant role when regarding various fields of application. Maps are redrawn in order to simplify the original map or in order to augment the geographic information available in the original map, i.e. contours of countries and borders, with related data. One example of such an augmentation is the insertion of labels in a map in order to tag certain places. Also the plotting of isolines is such an augmentation used in meteorology to represent some data (e.g. air pressure) in a simple digital map. The same holds for relief shading performed by geologists aiming to represent altitudes in a map.

All these redrawings have in common that an abstraction from the precise geographic map is made and the focus is shifted onto associated data which is in some way incorporated into the map. This abstraction however always preserves the basic characteristics of the original map so that the latter can still easily be recognized. When looking at the map the user can concentrate on the added data which has been visualized without spending any time on reading or understanding the basic contours of the map.

One specific kind of these redrawings are cartograms. A cartogram is a map in which the sizes of all countries are proportional to given quantities associated with every country, most often statistical data. Given a map and a vector which supplies a weight for every country in the map we look for a transformation of the map where all areas are equal to the weights and by applying this transformation we want to deform the map as little as possible. A good cartogram will give a very quick overview over the represented quantities and at the same time it will resemble the initial map sufficiently well so that one can

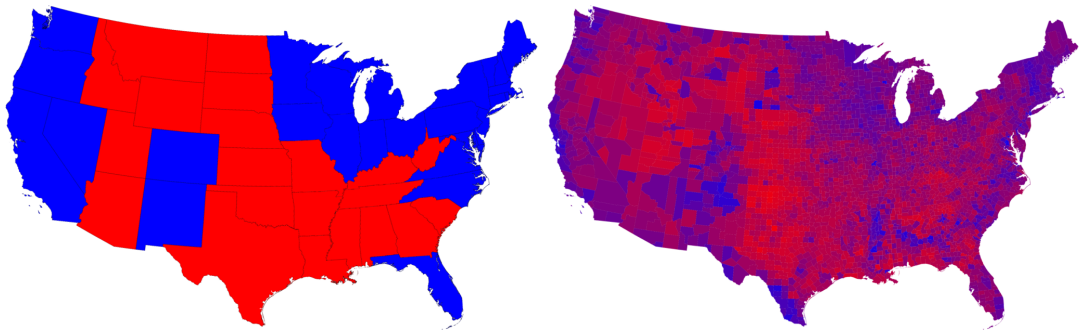
identify the different regions without difficulty. The convenience of cartograms lies thus in the fact that the distribution of the illustrated data can be studied comfortably with a quick glance. For instance one can compare the values of two countries by simply comparing their sizes in the map. Or one can find the country with the biggest value by roughly scanning over the map. The order of the countries according to the given data and rough proportions between the countries can be deduced easily. Nevertheless, it is obvious that a cartogram alone will never be as accurate as the set of numeric values itself. A good extension of a cartogram is thus to print the set of values next to it or to incorporate the values directly into the map, i.e. to print them as labels in the map. The fundamental advantage of cartograms over most other cartographic quantity illustration techniques is their simplicity and compactness.

The most primitive related quantity illustration technique is a simple table with two columns where every region is mapped to a value. The main advantage of tables over cartograms is that one can look up exact values and compare regions on a more precise level. In all other respects mentioned in the last paragraph a cartogram is superior to a table. Tables do not attain a comparable clarity because they do not make use of the potential that visual illustration has with respect to perceptibility by the human eye as opposed to numeric illustration.

A first approach to visualize the data from a table is to use a choropleth map. In a choropleth map the set of values is divided into intervals of a certain granularity and then regions are shaded with a color according to the interval which their value belongs to. The topology of the map is not modified. This method is suitable for certain purposes since it groups the regions in the map and one can quickly view the zones in the map where regions belonging to the same group are clustered. However, depending on the granularity this method is not very precise and besides it has the major drawback that a fast comparison between the values of different regions is hardly possible since the intervals might span quite a large range of values.

This problem of choropleth maps can be remedied partly by relating the color tone directly to the given values with a code, e.g. by providing some appropriate function. Then the coloring of the map can be adapted to reveal more nuance as opposed to the discrete shading performed in choropleth maps. That way the proportions are better and more quickly perceivable because one can estimate the value of a point or region by evaluating the tone of its color. This kind of drawing where regions in the map are colored with smoother transitions between them is called a heat map. Nevertheless, a heat map still does not provide a true proportional representation of the given quantities and it can be quite cumbersome to compare two given points in the map by retrieving the values associated to their color via the shading function.

An illustration of the difference between a choropleth map and a heat map is depicted in Figure 1.1. Both graphics are part of a study of Mark Newman visualizing the results of the 2008 US presidential elections [new]. The left Figure 1.1a is a simple choropleth map where the range of values between 0% and 100% has been divided into two intervals signifying whether Democrats (blue) or Republicans (red) have won the vote in that state. The information provided in this image is quite poor because one can not see whether the vote has been close or not in the different states. The heat map in Figure 1.1b is richer in information since the color tone per county is related immediately to the precise percentage of the vote. For instance we can now conclude that in Arizona which is pretty much violet in its overall color tone the win of Republicans was a lot closer (McCain 8.8% in front of Obama) than in Utah which is dominated by strong red (McCain 27.6% in front of Obama). One principal deficiency of both shading methods is that colors are not very appropriate to visualize quantities. The relation between colors and quantities is always



(a) Choropleth Map (b) Heat Map

Figure 1.1: Shading methods (graphics reproduced from [new])

an artificial one and a calculation is necessary in order to link a color to a value and vice versa. Also the two images in Figure 1.1 can be very misleading because most large US states in the west are sparsely populated while the states that are smaller in area in the east are densely populated but in the shading methods this distribution is not taken into account. The maps would be more meaningful if states were proportional in size to their population.

A more sophisticated method which takes quantities as sizes of the modeled objects into account is proportional contact representation, where regions of the map are represented as geometric objects which touch if the corresponding regions are adjacent in the map. These objects can be line segments, circles, triangles or other simple geometric objects. Hence, the only structure from the map that is preserved in the representation is the adjacency information given by the dual graph. Different classes of contact representation exist, for instance contact may be defined as point contact or as contact along an actual border or one could ask for a hole-free contact representation. Now, if the objects have sizes which comply with prescribed weights, then the contact representation is called proportional.

There are several results for contact representation and for proportional contact representation: In 1936 Koebe showed that any planar graph has a contact representation with circles if not considering weights [Koe36]. De Fraysseix et al. showed that the same can be done with triangles [dFdMR94]. Note that in both cases the restriction was made on point contact. Many applications however do require side contact, i.e. the objects must touch along a non-trivial border, since this requirement leads to more compelling results. Gansner et al. showed that any planar graph has a side contact representation with 6-sided polygons and that this is sometimes necessary [GHKK10].

Proportional contact representations which also take the quantities into account and model them as areas of the contact objects include Alam et al., see [ABF$^+$11]. They give upper bounds on the complexity of the objects, i.e. the number of bends per object, in a hole-free proportional contact representation with rectilinear polygons for certain graph types. For maximal planar graphs they describe a linear-time algorithm using 10-sided rectilinear polygons and for planar 3-trees they give an algorithm that cuts the complexity of the polygons down to 8 and still runs in $O(n)$ time. In the latter case they even show that the achieved complexity is optimal. Furthermore, they prove that only 6-sided polygons are necessary for maximal outer-planar graphs. Note that rectilinear representations are particularly interesting because they play a role in VLSI design.

Proportional contact representation is a good way to show quantities over a map if the visualization of the quantities is more important than the shape of the countries in the initial map of which only adjacencies are kept by performing the contact representation

on the map's dual graph. In many applications, especially in statistical illustrations in newspapers or books, this lack of resemblance between map and representation can not be tolerated. A method is needed that represents the weights proportionally and at the same time preserves the main topographic properties of the original map.

A way to rectify this deficiency of not even providing a roughly correct visualization of the map which all methods discussed so far have is to use proportional symbol maps. In a proportional symbol map geometric objects – in the majority of cases disks or squares – which are scaled to a size proportional to the prescribed weights are placed on the map while the map itself remains unchanged in its boundaries. In general, objects can be placed at any point in the map where data was collected (true point data), but if the objects are located in such a way that they aggregate the data for one entire region than we talk of conceptual point data and this comes close to the idea of a value-by-area map. The fact that in general objects are allowed to overlap in proportional symbol maps raises the question what the best positioning of the objects is such that their sizes can be estimated as good as possible. Clearly, a long visible border of an object benefits its recognizability and thereby the suitability for quick estimates. Cabello et al. address the problem of how to arrange the objects in order to maximize the amount of visible border [CHvKS10]. Most of the problems that they consider turn out to be NP-hard. Proportional symbol maps are a good technique to enhance a map with weights per country, especially if the layout of the map and structure of the weights allow for an easily readable positioning of the symbols with a small amount of overlap. However, this method suffers from a quickly growing complexity, for instance if many small countries occur in a zone of the map which implies that many symbols have to be arranged on a limited space.

It can thus be said that a well drawn cartogram is an advantageous way to visualize quantities in a geographic map when the user or application demands for an easily readable, aesthetic representation of the quantities in the map. Cartograms take both topology of the map and given quantities into account whereas the other methods which were discussed above all attempt to provide correct visualization of at most one of the two criteria (tables, choropleth maps, heat maps and proportional contact representation) or work with objects that must separately be added into the map (proportional symbol maps).

However, obviously there is a trade-off to be made in cartograms: If the data varies a lot from the actual geographic proportions, then we can not establish the correct areas while keeping the shapes of all regions close to their initial shapes. These two criteria mutually affect each other. It depends on the purpose of the visualization whether one or the other criterion is prioritized. Only a method that neatly balances between distortions of the two factors can lead to a satisfying cartogram. If a cartogram algorithm does not at all preserve the topology of the initial map then it is no more than proportional contact representation and if on the contrary it requires the output to be too close to the input than it might be unable to find cartograms for certain unfavorable weight distributions.

1.1 Related work

Different cartogram models have been proposed. A well studied one is the rectangular cartogram. In a rectangular cartogram all countries and their values are represented by rectangles of the right size. The pioneer in this field was Erwin Raisz who published his work in 1934, see [Rai34]. At this early stage Raisz was already conscious about what distinguishes a proportional map from a proportional (contact) representation:

> "It should be emphasized that the statistical cartogram is not a map. Although it has roughly the proportions of the country and retains as far as possible the relative locations of the various regions, the cartogram is purely a geometrical

design to visualize certain statistical facts and to work out certain problems of distribution." ([Rai34])

So, Raisz argues that his cartograms are not really maps, since the boundaries of regions are approximated by rectangles which are too primitive geometrical objects for the purpose of illustrating a geographic map. Nevertheless – and this is essential – Raisz' cartograms despite their radical abstraction are more than proportional contact representation as they take the layout of countries into account, i.e. where a country lies with respect to its neighbors. This is stronger than simply producing a representation from the dual graph. This fact of considering the original map's properties beyond the dual graph is the step that brings us from proportional contact representation to cartograms.

Van Kreveld and Speckmann were the first to give automated methods to produce rectangular cartograms [vKS07]. However there is no guarantee for correct areas and correct adjacencies. Further work in the field of rectangular cartograms was done by Heilmann et al. where correct areas are achieved but the strict constraint on the adjacencies is dropped as well [HKPS04]. Eppstein et al. studied area-universal rectangular layouts [EMSV09]. They answer the questions for which kind of rectangular layouts all area-assignments can be achieved with combinatorially equivalent layouts. De Berg et al. investigated cartograms for maps with an internally triangulated plane dual graph by using rectilinear polygons having a complexity of at most 40 [dBMS09]. Biedl and Vélazquez improved this bound to 12 [BV11].

More general cartograms without any limitation to a rectangular or rectilinear shape of the regions have also been studied in the cartography community. Tobler presented an early automated method to draw cartograms, see [Tob86]. He describes the problem mathematically as a minimization problem which is subject to boundary constraints and partial differential equations. Based on an iterative approach he then obtains approximations which can have big cartographic errors, thus the name "pseudo cartogram". Also in Tobler's approach there is the issue of very costly computations which slow down the algorithm.

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 [DCN85].

Keim et al. describe the distance of the cartogram from the original map with a metric based on Fourier transformation and then perform a scanline algorithm on the map which repositions the edges according to this metric [KNP04]. They obtain pretty appealing results with a significant cartographic error for only very few regions in their experiments.

Welzl, Edelsbrunner and Waupotitsch focus on combinatorial resolution approaches for cartograms [WEW97]. They define the generation of a sequence of homeomorphic deformations which produces the cartogram and the effect on the area distortion of which can be prescribed. They also discuss how quality of cartograms can be assessed with local distance distortion measures.

Dorling presents a grid and cell based approach where regions try to 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, see [Dor96]. In some cases this technique leads to big distortions of the regions' shapes which reduces the readability significantly.

Kocmoud and House propose another method which is a mix of Dorling's method and the approach of Edelsbrunner et al. [HK98].

Gastner and Newman present a cartogram model which relies on elementary physics [GN04]. They express the desired balancing of the regions' weights as an iterative diffusion process where quantities are flowing from one country to another until a balanced

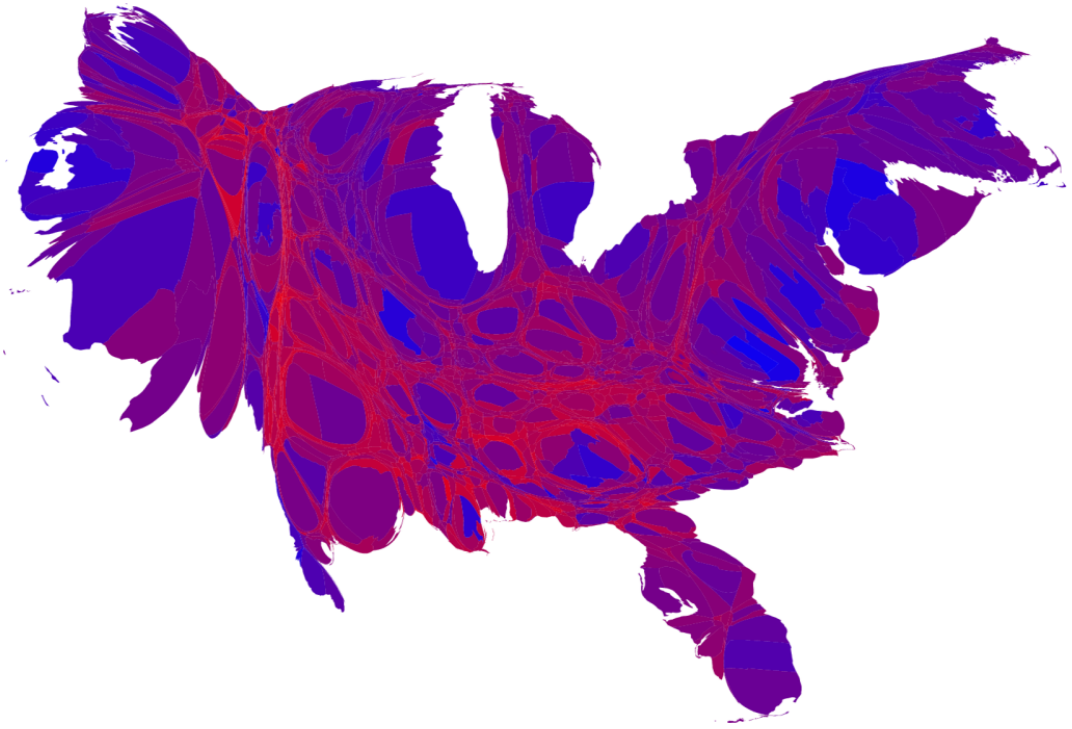


Figure 1.2: A population cartogram augmented with a color code (graphic reproduced from [new])

distribution is reached, i.e. the density is the same everywhere. The cartograms produced this way are well readable, compact and have no cartographic error. Plus the computation is not very costly. Nevertheless, the method does allow to control the deformation of single regions. If there is a big discrepancy between the weight and the initial area of a region, no trade-off will be made and the region will probably not at all look alike in the resulting cartogram. In many cases it could be desirable though to accept a certain cartographic in favor of a more accurate geographic representation.

Recall that earlier we identified one major deficiency of methods that do not model quantities as areas in the map which was the fact that the resulting maps can be very delusive since human visual perception is best adapted to deducing quantities from sizes and not from colors for example. The two maps for the 2008 US presidential elections in Figure 1.1 were therefore not satisfactory as the amount of red and blue in the map does not at all match the actual outcome of the election. Figure 1.2 illustrates the power of cartograms. A population cartogram generated with Gastner and Newman's algorithm has been shaded with the same code as the heat map in Figure 1.1b and the outcome of the election is therefore represented proportionally to the population. Practically we can now see that blue and red are quite balanced in the map with blue predominating slightly and we conclude that the vote must have been close but Democrats prevailed (end result: Democrats 52.9% – Republicans 45.7%).

Frequent applications of cartograms are illustrations of population distribution or statistical variables like GDP. Cartograms are discovered as a useful cartographic tool by more and more users, especially by newspapers and journals, but also by public authorities for instance. New computational cartogram generation methods which are efficient and produce high-quality maps are therefore much in demand. All of the methods presented here have significant weaknesses and therefore researchers continue to explore new approaches.

1.2 Overview

In this thesis we study a new cartogram generation approach which is based on an action of bending the border between different countries in order to transfer area. Borders consisting of straight line segments are thus replaced with curved segments. Here we limit ourselves to circular arcs because this is a kind of curve which is easy to control and still provides a good ability to shift a big amount of area from one region to another. The overall goal is to obtain maps with smoothly curved interior and exterior borders where the countries possibly look a bit like clouds if the edges are bent outwards and like flakes if the edges are bent inwards. Such a cartogram allows to quickly identify the countries which increased their area and those which decreased their area due to the distinctive shape of inflated (clouds) and deflated (flakes) regions. This holds in particular when imposing the additional requi rement that increasing regions can only use outwards bent circular arcs and vice versa. This requirement is called *normal-bending* and will be explained more in detail later in this thesis.

A manually drawn exemplary cartogram is shown in Figure 1.3. Inflated and deflated regions can easily be distinguished and the fact of having longer borders as opposed to a drawing with straight segments emphasizes the existing adjacencies in the map.

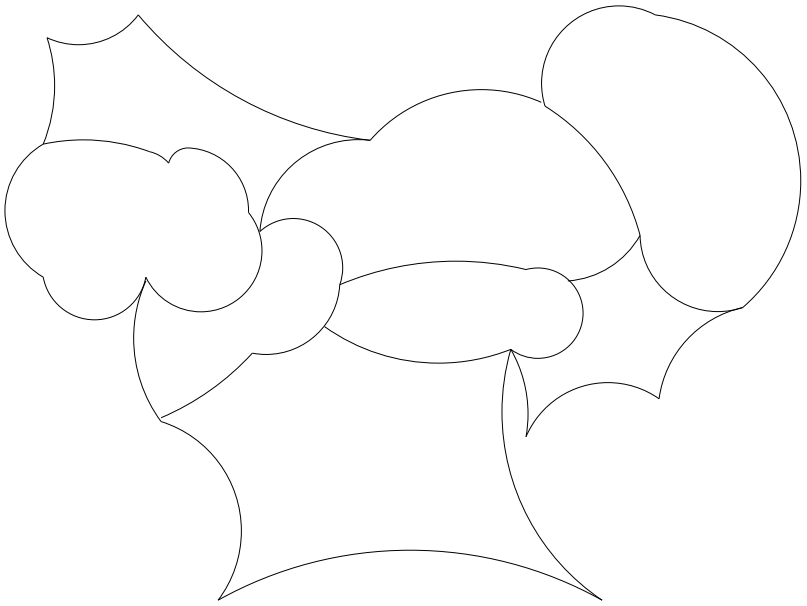


Figure 1.3: A hand drawn circular-arc cartogram

After the introduction to the topic of our work in this chapter the next Chapter 2 will acquaint the reader with notations, algorithms and concepts from graph theory and cartography which will be relevant in this thesis.

In Chapter 3 the problem at hand will be stated formally and a short overview over positive and negative problem instances in the case of single polygons will be given. We investigate the questions for which polygons any positive weight can be attained by replacing straight segments with circular arcs and which combinations of polygons and weights are not feasible. Then in the main Section 3.3 of this chapter complexity results for different variants of the problem will be derived. All studied variants are proven to be NP-hard. The reductions are based upon each other, so it is recommended to read them one after another.

Some possible relaxations are discussed in Chapter 4. These relaxations drop certain requirements which are imposed on a solution of an instance with the aim to render the

problem efficiently solvable. We treat possible relaxations of the area constraint as well as relaxations of constraints that are related to vertex positions in the map.

A heuristic which uses the relaxations from Chapter 4 and which is based on a max-flow algorithm is then presented in Chapter 5. The chapter ends with an evaluation of this heuristic through a case study which discusses the visual output of an implementation.

2. Preliminaries

In this chapter a short overview will be given over the theoretic fundamentals and algorithms from graph theory and computational geometry which are used in this thesis.

Graph A *graph* $G = (V, E)$ consists of a set of *vertices* V and a set of *edges* E. Edges are unordered pairs of vertices $\{u, v\}$ representing a link between u and v. Two vertices that have a link are said to be *adjacent*. The *degree* of a vertex is the number of vertices adjacent to it. A graph is *simple* if no loops (edges $\{v, v\}$) and no multiple edges occur. Graphs will always be assumed to be simple in this thesis. In a graph the cardinalities of V and E are denoted n and m respectively.

Topological embedding of a graph A *topological embedding* or a *plane drawing* of a graph is a mapping of its vertices to points in the Euclidean plane and of its edges to curves such that none of the curves intersect in their interior and the endpoints of a curve are exactly the two points associated with the vertices of the corresponding edge. A curve does not contain any points associated with vertices other than its two endpoints.

Combinatorial embedding of a graph A *combinatorial embedding* of a graph consists of the definition of a cyclic order of all incident edges around any vertex of the graph. Such a cyclic order is sometimes also called rotation system. A combinatorial embedding can be viewed as an equivalence class of topological embeddings.

Planar graph A *planar graph* is a graph that can be topologically embedded, i.e. a graph that can be drawn in the plane such that none of its edges cross.

Plane graph A *plane graph* is a planar graph with a given topological embedding (a given drawing of the graph). The connected pieces of the plane that are bounded by the edges of a plane graph are called *interior faces* (or simply *faces*). The piece of the plane unbounded by a plane graph is called the *outer-face*.

Planar straight-line graph A *planar straight-line graph* is an embedding of a planar graph such that the edges of the graph are mapped to straight line segments. Fáry showed that every planar graph has an embedding as planar straight-line graph [Fár48].

Polygonal subdivision of the plane A planar straight-line graph without vertices of degree 1 is called a *polygonal subdivision* as it subdivides the plane into disjoint regions or faces. Polygonal subdivisions are a common term in computational geometry used to represent geographical maps. The *edges* E of a polygonal subdivision are the straight line segments of the graph and the *vertices* V are all endpoints of edges. The cardinalities of V and E are denoted by n and m. The bounded pieces of a polygonal subdivision are its *regions* or sometimes *countries*. Every region of a polygonal subdivision is thus a polygon. Regions can also be labeled as *lakes* which means that they are not true administrative entities like countries, states or counties. The piece of the plane unbounded by a polygonal subdivision is called the *sea*.

Dual graph The *dual graph* G_D of a plane graph G has a vertex associated to each face of G and an edge if the two faces corresponding to the vertices have a common border in G. If two faces share several common borders that are disjoint, then one edge in the dual graph represents all of these borders. A dual graph does thus not contain multiple edges. Plane graphs with the same combinatorial embedding have the same dual graph. The dual graph can be defined likewise for polygonal subdivisions.

Directed graph Like a graph a *directed graph* G is defined as a pair of vertices and edges (V, E), except that in a directed graph an edge e is an ordered pair of two vertices $e = (u, v)$, meaning that e is an edge directed from u to v. For an edge (u, v) we say that (v, u) is its *reverse edge*.

Directed path A *directed path* p in a directed graph is a sequence of vertices $(v_1, \ldots, v_l)$ with $(v_i, v_{i+1}) \in E$ for $1 \leq i \leq l-1$. We say that p is a path from v_1 to v_l and its length is $l - 1$.

Flow network A *flow network* is a directed graph G with two designated vertices s and t, called the *source* and the *sink* of the network, and a capacity function $c : E \mapsto \mathbb{R}_0^+$ which assigns non-negative capacities to the edges of G. For any edge (u, v) in E the reverse edge (v, u) must be in E as well. A *valid flow* in such a network is an assignment of real numbers to the edges $f : E \mapsto \mathbb{R}$ which has the following properties:

$$f(u,v) \leq c(u,v) \quad \forall (u,v) \in E \qquad \text{(capacity constraint)}$$

$$f(u,v) = -f(v,u) \quad \forall (u,v) \in E \qquad \text{(skew symmetry)}$$

$$\sum_{(u,v)\in E} f(u,v) = 0 \quad \forall u \in V/\{s,t\} \qquad \text{(flow conservation)}$$

The value of a flow is $|f| = \sum_{(s,v)\in E} f(s,v)$. Given a flow f the *residual capacity* c_f of an edge in a flow network G is defined as $c_f(u,v) = c(u,v) - f(u,v)$. The residual network of a flow network G is $G_f = (G, s, t, c_f)$. An edge is *saturated* if its residual capacity is 0. We call an *augmenting path* in G a directed path from s to t with all edges on the path non-saturated.

Maximum flow problem The *maximum flow problem* in a flow network G is to find a valid flow f of maximum value. There are several algorithms for the maximum flow problem. The first one is the algorithm of Ford-Fulkerson which finds augmenting paths and pushes more flow over the unsaturated edges, see [FF56]. Their method can however

not be guaranteed to terminate unless all capacities are integer. Numerous variations have been proposed that improve the method of Ford-Fulkerson and the runtime of which can be polynomially bounded, for instance the algorithm of Edmonds-Karp which can be implemented with a runtime of $O(nm^2)$, see [CLRS09]. A different approach is the Push-and-Relabel algorithm by Goldberg and Tarjan which runs in $O(n^2m)$ time, see [GT86].

One particular variation of augmenting-path algorithms is the one of Boykov and Kolmogorov [BK04]. While their method does not have a polynomial runtime bound ($O(mn^2|C|)$) because it depends on the cost of a minimum cut C (similar to Ford-Fulkerson), they show that in practice it performs very fast and in fact better than other standard algorithms.

The algorithm of Boykov and Kolmogorov builds up one source search tree S and one sink search tree T. Vertices are labeled if they belong to either one of these two search trees. Vertices that do not belong to S or T are said to be free. In S and T all edges directed towards the sink are non-saturated. It always hold that S and T do not overlap: $S \cap T = \emptyset$. Vertices are tagged as active or passive throughout the execution of the algorithm. Only vertices belonging to a search tree can be active (free vertices are always passive). During execution the algorithm iterates over three phases:

1. **grow-phase**
   Active vertices can acquire new free vertices to their search three that are adjacent via a non-saturated edge. After exploration of all neighbors the current vertex becomes passive. The set of active vertices is treated iteratively as a whole independent of the two search trees. If no more active vertices are left, this phase terminates and the current flow is optimal. The phase also terminates if acquisition of a vertex from the opposite search tree would have been possible. In that case an augmenting path from source to sink was found.

2. **augment-phase**
   In this phase the bottleneck capacity on the found augmenting path from phase 1 is added to the flow on this path. The search trees possibly collapse into forests because edges become saturated.

3. **adopt-phase**
   The search trees are restored in this phase: The vertices that were cut from their search trees because of the flow augmentation can be reconnected to the search tree by non-saturated edges (they are thus assigned a new parent). If such an edge does not exist the disconnected vertex can be marked as free in order to be newly assigned in the next phase.

The advantage of this algorithm is that it is adapted to work with positive flows on an edge and its reverse edge at the same time. The skew-symmetry constraint from the definition of valid flows can thus be dropped when using this method.

Transshipment problem The *transshipment problem* is also a network flow problem. The particularity here is that any vertex can be a source or sink whereas in a normal network flow all vertices are just transferring flow except the two designated source and the sink vertices which can have a surplus or loss of flow. More formally we define the problem as follows: G is a directed graph with a capacity function $c : E \mapsto \mathbb{R}_0^+$ assigning non-negative real numbers to the edges. Additionally, costs $p : E \mapsto \mathbb{R}_0^+$ are assigned to the edges. Vertices are weighted with a function $b : V \mapsto \mathbb{R}$ which describes whether a vertex has a demand (b negative) or supply (b positive) or whether it is a transfer node (b equal to 0). In a valid flow f for the transshipment problem the two following conditions hold:

$$f(u,v) \leq c(u,v) \quad \forall (u,v) \in E \qquad \text{(capacity constraint)}$$

$$\sum_{(u,v)\in E} f(u,v) = b(u) \quad \forall u \in V \qquad \text{(demand-supply balance)}$$

So, the first condition from the definition of Network flows holds (capacity constraint), the second does not (skew symmetry) and instead of the third condition a demand-supply balance constraint is introduced which allows for several sinks and sources. All sources and sinks have to satisfy their demand, or reduce their supply to 0 respectively. The goal is to find a valid flow which minimizes the total cost

$$\sum_{(u,v)\in E} f(u,v)p(u,v)$$

This kind of problem is closely related to other network flow problems.

Straight Skeleton Evolution The *straight skeleton* of a simple, closed polygon consisting of straight line segments exclusively is constructed by inwards offsetting the edges of the polygon simultaneously at a constant speed. During this action of offsetting the vertices of the input polygon are traced until two traces intersect. Upon an intersection the polygon is split into two which means that the bisector between the two intersecting traces is the new trace to be followed. At the end of this tracing process a partitioning of the original polygon into several regions will be obtained. Each of these regions corresponds to exactly one contour edge of the initial polygon. Methods exist for the computation of straight skeletons which run in $O(n^2 \log(n))$ time, see Aichholzer et al. [AAH+07].

Bipartition *Bipartition* is a NP-complete decision problem (see [GJ79]): Given a set of positive integers $S = \{x_1, \ldots, x_n\}$ is there a subset S' of S such that all integers in S' sum up to the same value as all integer not in S'?

Subset sum *Subset sum* is a NP-complete decision problem (see [GJ79]): Given a set of positive integers $S = \{x_1, \ldots, x_n\}$ and a positive integer K is there a subset S' of S such that all integers in S' sum up to K?

3. Circular-arc Cartograms

Starting from a map given as a polygonal subdivision we want to find a transformation such that the transformed map is a cartogram with respect to a given weight vector. The only operation allowed in this transformation is the substitution of straight segments with circular arcs.

In this chapter we will define the problem at hand formally and give some results concerning possible bending configurations for the case where the underlying map is a single polygon. We then treat several variants of the problem which differ in the constraints they impose on a resulting cartogram in order to be valid.

Note that the words 'map' and 'subdivision' will be used interchangeably in the remainder of this thesis. When talking about geometric properties and operations the expression 'subdivision' will be used preferably whereas the same object might be called 'map' (or 'cartogram' which is the transformed map) in a context where for example visual properties of the output are pointed out. In Section 4.2.1 we also identify a map or subdivision with a plane graph which is equivalent.

3.1 Problem Statement

Given a polygonal subdivision of the plane S which can be viewed as the set of straight edges that it consists of we define a circular arc configuration as an assignment of a point C_e lying on the perpendicular line through e's midpoint and of a directed normal vector d_e being perpendicular to e to each edge e of S. This assignment uniquely defines a circular arc through e's endpoints for each e where C_e is the center of the circular arc and d_e gives the direction into which e is bent. If all edges in S have been associated with a circular arc in this way we talk of a circular-arc configuration (sometimes also called bending configuration in this thesis). If additionally none of the circular arcs in such a configuration intersect or introduce new adjacencies we say that the configuration is valid. Obviously there is an infinite number of valid circular-arc configurations for any subdivision. See Figure 3.1 for an illustration of an assignment of a center C_e and direction d_e to edge e.

With this definition of a circular-arc configuration we can then specify the problem which underlies the rest of this chapter:

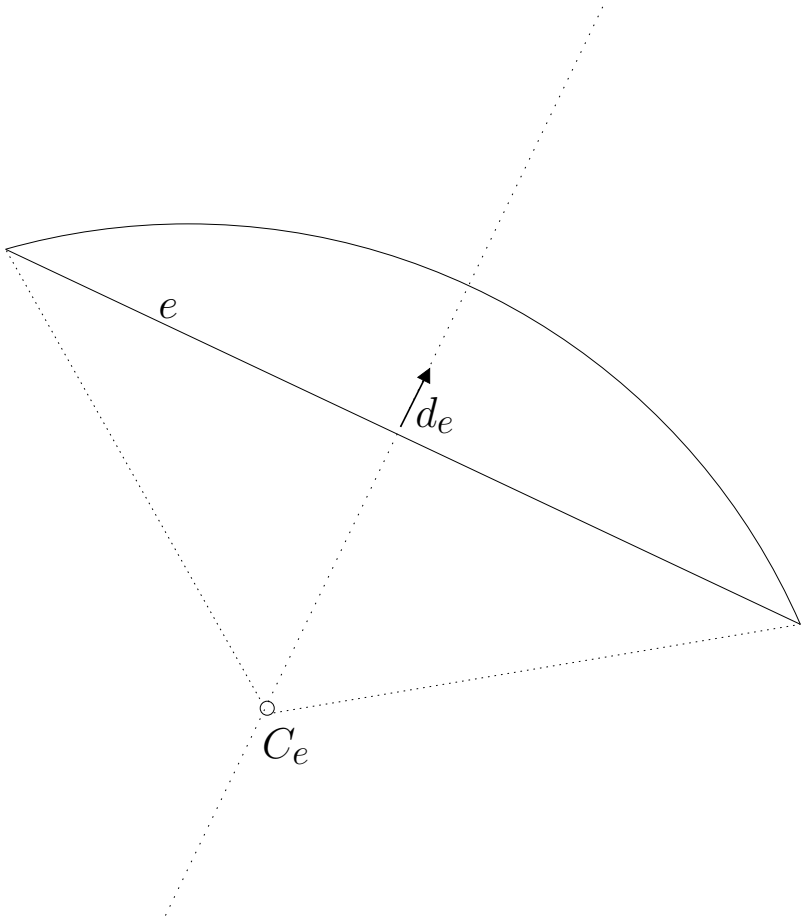


Figure 3.1: Circular arc configuration at e by choosing C_e and d_e

Problem 3.1.1 (Circular-arc Cartogram) *Given a polygonal subdivision of the plane S and weights t that are assigned to the regions of S. Does a valid circular-arc configuration S' exist where all areas of the regions comply with the vector t?*

This is the Circular-arc Cartogram problem (CAC) in the general setting. Several variants will be defined later by introducing further going restrictions on what we consider a valid configuration.

3.2 Single Polygon Bending Configurations

This section contains results for possible bending configurations of single polygons which give detailed evidence that positive as well as negative instances exist for the problem at hand. We will see that it is not even necessary to consider more complex subdivisions with more than one region in order to give weights that can not be realized as areas in a cartogram. Interesting questions for single polygon bending configurations include:

- Are there polygons for which any target area can be achieved with valid bending configurations?
- Is there a classification of these polygons?
- Which polygons admit target values (weights) that cannot be realized?

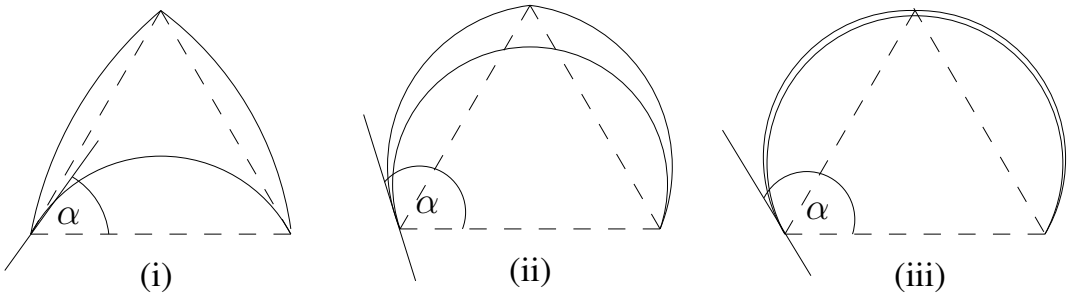


Figure 3.2: Configuration with $120° \geq \alpha > 60°$

3.2.1 Feasible Instances for any Weight

Theorem 3.2.1 *For polygons that admit a circumscribing circle any target area can be achieved with valid bending configurations.*

Proof In order to give an intuition consider an equilateral triangle T with side length l. Its area is A_T and its height is h_T. We will first show that we can shrink this region to an arbitrarily small sickle by bending the edges.

We are allowed to bend all the edges, but must observe the rule that no intersections can be introduced. Let α be the angle between the tangent of the arc replacing the lower side segment and the original straight segment at one of the two endvertices. In the following $A(s, \alpha)$ denotes the area under a circular arc with segment length s and angle α between the tangent of the arc at one of the vertices and the segment. Consider what happens if $\alpha > 60°$, so the lower edge is bent inwards and the created arc exceeds the original boundaries of the triangle. The two other edges need to be bent outwards in order to avoid intersections. As we want the decrease to be maximal, the two other edges will be bent as little as possible. See Figure 3.2 for an illustration of several bending configurations where this idea is applied.

Note that the height of the circular arc created from the lower edge is bounded by h_T as this is the point where the arc would touch the upper point of T which is immovable. We deduce a bound for the angle of $\alpha \leq 120°$. The angles at the tangents for the arcs of the two other edges are $\alpha - 60°$. So they have the same tangent as the lower arc at their common bottom vertex. That way intersections are avoided. The area of the object that we obtain is $A_T + 2A(l, \alpha - 60°) - A(l, \alpha)$. This area function tends towards its minimum at $\alpha = 120°$ where the value is 0. In fact for this angle the circular arc is part of a circumscribing circle of the triangle and we can approach this circle as close as we want to with circular arcs emerging from the other two edges. So we can get an arbitrarily small target area with this configuration.

Making the area arbitrarily big is very easy for this polygon: We simply take one edge and bend it outwards as far as required (no intersections will occur).

This result can be extended to all convex polygons with cocircular points. The proof is in brief words: For these Polygons we can obtain a circumscribing circle C which can be approached with circular arcs arbitrarily close making the area 0. □

3.2.2 Effect of Imposing Normal Bendings

Theorem 3.2.1 only holds if we allow bending edges outwards for a region that wants to decrease its area which is rather unnatural. If we disallow this we can prove that for the polygons used above not any target area can be attained.

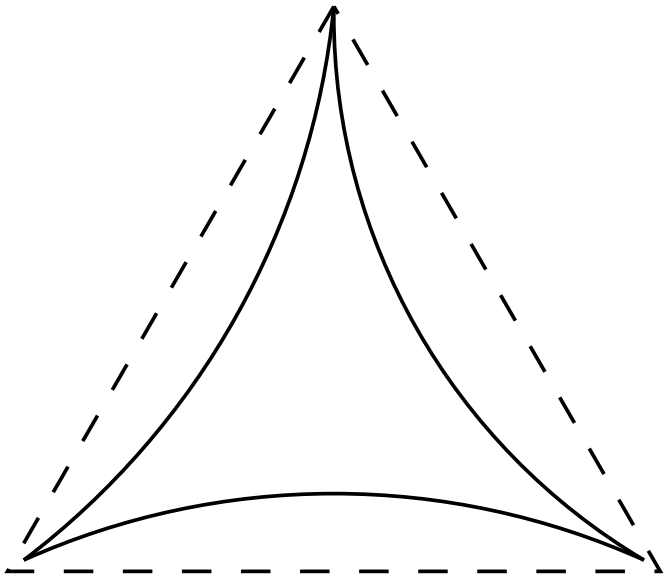


Figure 3.3: Circular arcs always leave an empty space when only allowed to be bent inwards

Normal bending We say that a valid bending configuration respects the *normal bending condition* if edges are bent outwards exclusively for regions which want to increase their area and if edges are bent inwards exclusively for regions which want to decrease their area.

Theorem 3.2.2 *Under the normal bending condition certain target areas can not be achieved for polygons that admit a circumscribing circle.*

Proof Consider again an equilateral triangle with side length l. If we let its target area tend towards 0 no possible bending configuration of its edges that respects the normal bending condition can attain this value. See Figure 3.3 for an illustration. The ciruclar-arcs are all bent into the polygon maximally but there is no way to subtract the remaining are in the middle. □

3.2.3 Infeasible Instances

Theorem 3.2.3 *Polygons exist for which certain target values cannot be realized with circular arcs in the general setting (ignoring the normal bending condition).*

Proof A necessary condition for arbitrarily big area modifications of single polygons with edge bending is that there is at least one edge that separates the plane into two half-planes. One half-plane that contains all of the polygon's vertices and another one that is empty, i.e. that does not contain any vertices. If this is the case a circular arc can be expanded in the empty half-plane as far as required to attain any area. We can modify the equilateral triangle very easily in order to violate this condition: We insert small spikes in the middle of each of the three triangle edges. Every edge now has a blocking vertex that prevents it from admitting arbitrarily big circular arcs. See Figure 3.4, we distinguish shorter edges (type f) and longer edges (type e). Edges of type e get blocked by the upper spike vertex and edges of type f get blocked by one of the corners of the initial triangle.

So we have found a class of polygons with an upper bound on their area-increase. □

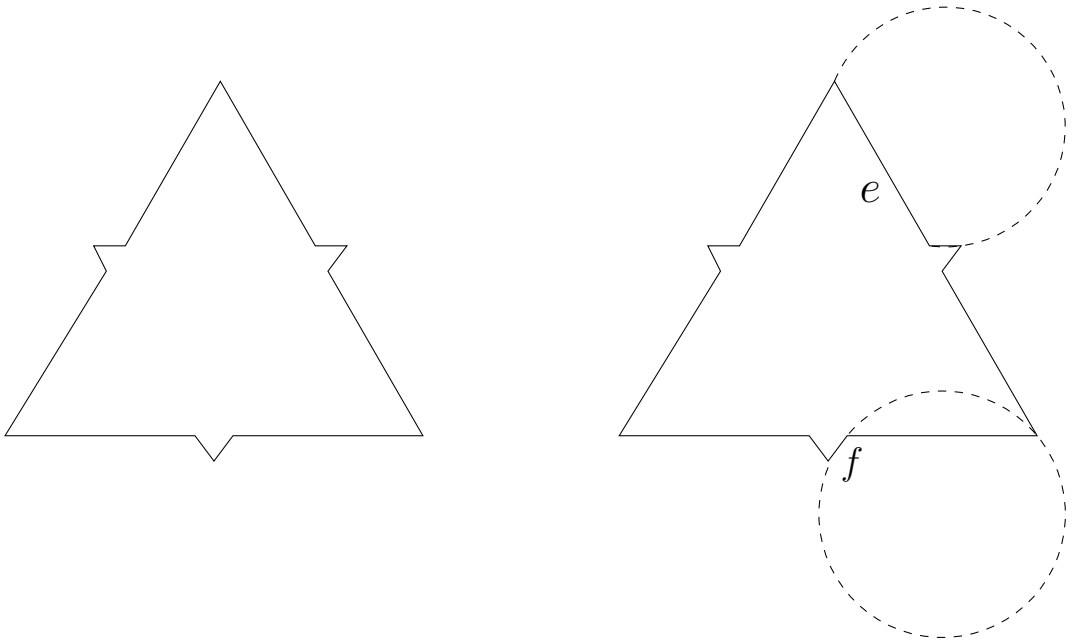


Figure 3.4: No edge has an empty half-plane on one of its sides

The overall result from this section can be summed up as follows:

Theorem 3.2.4 *The polygons for which any area in the range* $[0, \infty)$ *can be achieved with a valid bending configuration are exactly the polygons that admit a circumscribing circle.*

The first implication was proven in Theorem 3.2.1. It is trivial to see that if a polygon does not admit a circumscribing circle then an arbitrarily small area can not be achieved with circular arcs.

3.3 Complexity of Several CAC Variants

As defined in Definition 3.1.1 the problem Circular-arc Cartogram (CAC) deals with the question whether a cartogram can be obtained from a polygonal subdivision by replacing straight line edges with circular arcs in order to achieve desired areas in each region. In the preceding Section 3.2 positive and negative instances for this problem were presented and a classification of single polygons with respect to their resolvability was given.

In the following we present complexity results for different CAC decision problems. We will presume the real-RAM model of computation for this chapter which was introduced by Shamos and Preparata in 1985, see [PS85]. This model is used above all by computational geometers, for instance when dealing with algorithms or computability results that involve real numbers which need to be handled as continuous entities or that involve operations like square roots or trigonometric functions. Brattka and Hertling presented a real-RAM model which allows to use the computational power of Turing machines on real numbers, see [BH98].

3.3.1 Parameterized Version

In Circular-Arc Cartogram with Parameter ($CACP$) we consider the following variant of CAC:

Problem 3.3.1 (Circular-arc Cartogram with Parameter) *For a given polygonal subdivision S with target values t_i (positive real numbers) for each region R_i of S and an integer k decide if S admits a valid bending configuration where all areas of the deformed regions comply with the vector t. In addition this bending configuration can bend at most k edges.*

For certain subdivisions and target vectors the answer is 'no' and such a cartogram does not exist (for any k, even for $k = m$). Evidence of this is given in Section 3.2.

In the remainder of this thesis the target area vector t will often be given implicitly by providing the target area change vector Δt. Let A_i denote the initial area of the i-th region in the subdivision. Then t can be obtained as follows: $t_i = A_i + \Delta t_i$.

Theorem 3.3.1 *CACP is NP-hard*

Proof Consider the following reduction of Bipartition to the problem in question.

Let $I = (x_1, \ldots, x_n)$ with all x_i integer be an instance of Bipartition. From I we construct an instance I_{CACP} of CACP as follows:

- subdivision: One rectangle R_i per x_i (height $h_i = \sqrt{\frac{2x_i}{\pi} + \varepsilon}$ and width $w_i = 2\sqrt{\frac{2x_i}{\pi}}$). All these rectangles touch alternately at an upper or lower vertex and are enclosed from above by one region (named R_0) and one from below (R_{n+1}). We will specify the exact value of ε later.
- target values (implied by desired area changes Δt per region):

  $\Delta t_i = -x_i$, for $i = 1 \ldots n$

  $\Delta t_0 = \Delta t_{n+1} = \frac{1}{2} \sum_{1 \leq i \leq n} x_i$
- parameter $k = n$

The construction of an instance of $CACP$ is illustrated in Figure 3.5 along with a valid bending configuration (dashed circular arcs). The total area of the two shaded circular segments equals the total area of all four non-shaded circular segments.

We show that I has a solution if and only if I_{CACP} has a solution.

$\Leftarrow$ Let first I_{CACP} be a positive instance of $CACP$. Hence, a valid bending configuration C exists where all regions R_i in the subdivision attain their desired area increase or decrease. We analyze the configurations C and construct a solution for I from it.

All rectangles R_i with $i = 1 \ldots n$ want to decrease their area, so we need at least one circular arc bent inwards for each one of those. As we can only use n arcs for these n regions, we are obliged to use exactly one arc per region (none of the rectangles share any edges). Bending the right or left edge of R_i until it gets blocked leads to an area decrease of $\frac{\left(\frac{h_i}{2}\right)^2 \pi}{2}$ as the resulting circular segment is a half circle with radius $\frac{h_i}{2}$. We can now set $\varepsilon = \frac{2}{\pi}$ and obtain $\frac{\left(\frac{h_i}{2}\right)^2 \pi}{2} = \frac{x_i}{4} + \frac{\varepsilon\pi}{8} = \frac{x_i+1}{4} < x_i$, for $x_i \geq 1$. Using the upper or lower edge of R_i results in a half-circle of radius $\frac{w_i}{2}$ and therefore leads to an area decrease of exactly x_i being subtracted from region R_i and being added either to R_0 or to R_{n+1}. The height of this circular arc is always $\sqrt{\frac{2x_i}{\pi}} < h_i$, so

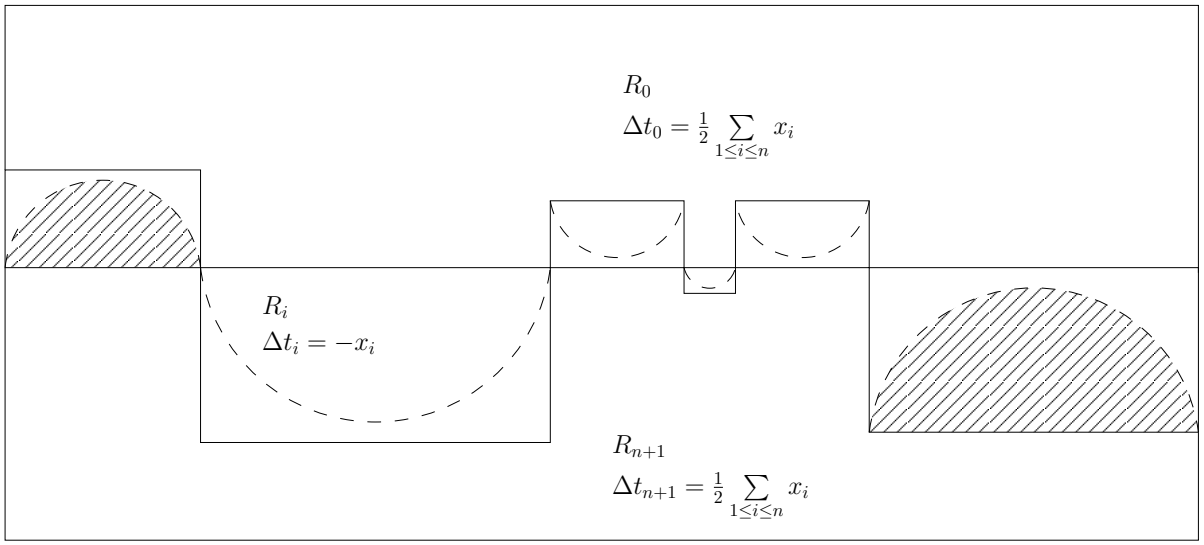


Figure 3.5: Construction of the subdivision for CACP

the circular arc does not touch the opposite edge and the region remains contiguous. So, for all R_i either the upper or lower edge is bent until it gets blocked by the left and right edge, always leaving a gap that keeps the region contiguous. If the target area of R_0 can be met we can deduce a subset of the x_i where x_i is in the subset if and only if the top edge of R_i is bent. All x_i in this subset sum up to $\frac{1}{2} \sum_{1 \le i \le n} x_i$ as no additional circular arcs can change the area of R_0 since we already used the maximum of n arcs. Equally all x_i not in this subset sum up to $\frac{1}{2} \sum_{1 \le i \le n} x_i$. The bending configuration C implies thus a solution for I.

$\Rightarrow$ A solution for I, i.e. a subset of $\{x_1 \dots x_n\}$ with total value $\frac{1}{2} \sum_{1 \le i \le n} x_i$, can be translated into a solution for I_{CACP} where the lower edge of R_i is fully bent inwards if x_i is in the subset and the upper edge of R_i is fully bent inwards otherwise. It can easily be checked that this configuration satisfies all requirements. □

Remark: In this construction only horizontal and vertical edges have been used for the construction of the subdivision. That means that already the special case of $CACP$ where all edges need to be rectilinear is NP-hard.

3.3.2 Requiring the Stable Sea Property

We have shown that the general $CACP$ is NP-hard. One case of $CACP$ with particular interest is the case were k is fixed to the total number of edges in the original polygonal subdivision. This allows us to bend any edge in the subdivision whereas for example in the reduction above with the right choice of k we only could bend one edge per interior region. Abandoning the parameter k leads thus to the initial CAC problem and comes closer to the natural question of constructing the cartogram. So let us study CAC with a specific restriction on valid bendings now. The problem formulation underlying Circular-Arc Cartogram with Stable Sea ($CACSS$) is the following:

Problem 3.3.2 (Circular-Arc Cartogram with Stable Sea) *For a given polygonal subdivision S with target values t_i (positive real numbers) for each region of S decide if S admits a valid bending configuration where all areas of the deformed regions comply with the vector t. In addition in a valid bending configuration edges neighboring the sea or lakes cannot at all be bent. We call this requirement the stable sea property.*

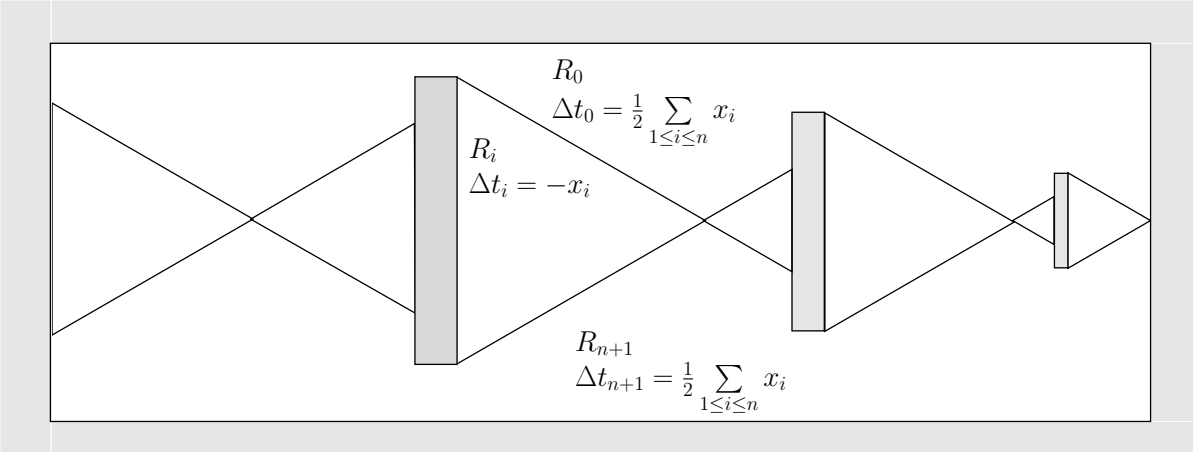


Figure 3.6: Construction of the subdivision for $CACSS$

The stable sea property can be advantageous in situations where one needs to obtain cartograms with no deformations of the global shape in order to guarantee quick recognizability of the map as a whole. Many recent cartogram generation approaches do not consider this a crucial aspect of quality and the maps they produce may have an outline that does not at all resemble the original one.

Theorem 3.3.2 *CACSS is NP-hard*

Proof In the case where we require the stable sea property instead of a parameter k we will have to use a different reduction. The problem with the previous reduction here is that since any edge might be bent, infinitely many configurations exist that lead to the desired area decrease. We will use the same principle for the proof as before but simplify the argumentation by going back to even less complex objects than rectangles, namely equilateral triangles.

Let $I = (x_1, \ldots, x_n)$ with all x_i integer be an instance of Bipartition. We construct an instance I_{CACSS} of $CACSS$ from I consisting of a polygonal subdivision and a target weight vector as follows:

In this construction again we will have one region R_i for each x_i, this time with the form of an equilateral triangle with side length $s_i = \sqrt{\frac{x_i}{c}}$ where c is a constant the value of which we will determine later. These triangles are arranged in a horizontal chain in pairs of two triangles that touch at one vertex. Additional lakes with suitable dimensions that separate the pairs of triangles and always border one entire edge of their right and left neighboring triangle will be added. Again the region on top is called R_0 and its target area increase is $\Delta t_0 = \frac{\sum x_i}{2}$, equally for R_{n+1}. The target area decrease of triangle R_i is $\Delta t_i = -x_i$.

Figure 3.6 depicts an example of a subdivision created with this transformation. The sea and the lakes in the subdivision are marked as shaded areas in the image.

It remains to show that the given weights are feasible in the subdivision if and only if the instance of Bipartition has a solution:

$\Leftarrow$ Let I_{CACSS} be positive and let C be a solution, i.e. a valid bending configuration. We show how a solution for I can be obtained from C.

In C all areas match the target areas and additionally the stable sea property is respected which means that no sea or lake edges are curved (the radius of the circular

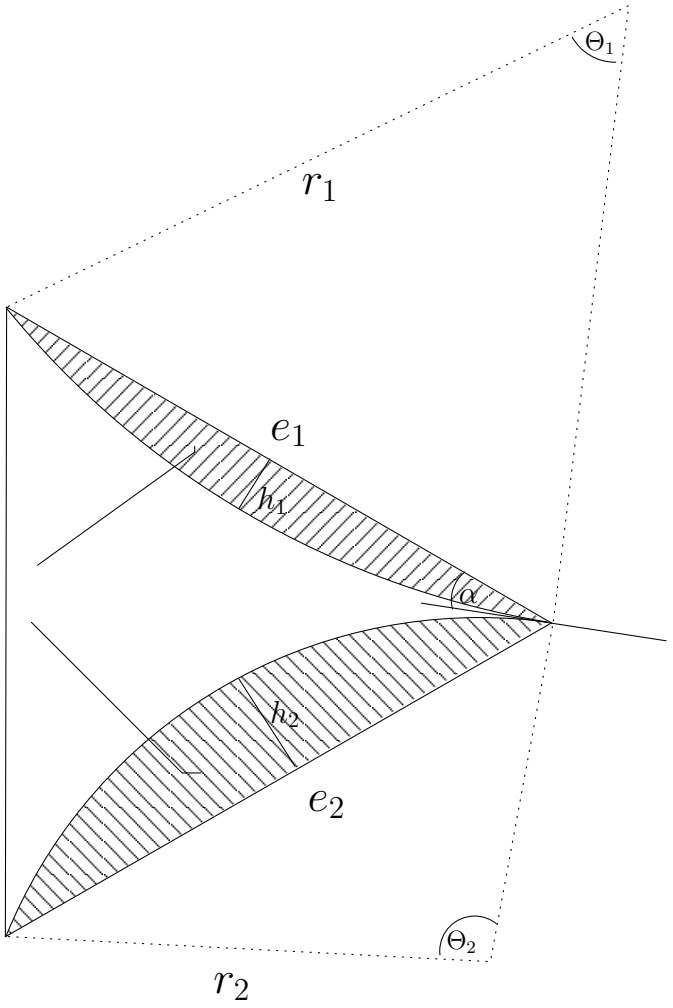


Figure 3.7: Bending of the two edges at triangle R_i

arc is $+\infty$). Hence, we know that for R_i the decrease has been achieved exclusively by using two edges, one bordering R_0 and one bordering R_{n+1}. This is because of our construction which places a lake or sea region next to one edge of each triangle which then cannot be bent because of the stable sea property. The angle between the two edges which can still be bent, let us call them e_1 and e_2, is initially 60°. We denote $A_{e_1+e_2}(\alpha)$ the area that is subtracted from R_i when e_1 is bent to a circular arc whose tangent at the common vertex of e_1 and e_2 encloses an angle of α with the straight line e_1 and e_2 is bent as far as possible until becoming tangent to the arc from e_1. The implied angle of the tangent of e_2's circular arc with e_2 at the same vertex is $60° - \alpha$.

See Figure 3.7 for an illustration of the circular arcs and corresponding circular segments at R_i. The central angles of the circular segments can be given as

$$\Theta_1 = \pi - 2(\frac{\pi}{2} - \alpha) = 2\alpha$$

and

$$\Theta_2 = \pi - 2(\frac{\pi}{2} - (60° - \alpha)) = 120° - 2\alpha = 120° - \Theta_1.$$

The height of the circular segment j ($j = 1, 2$) can be calculated with the formula

$$h_j = \frac{s_i}{2} \tan\left(\frac{\Theta_j}{4}\right).$$

The radius is given as

$$r_j = \frac{4{h_j}^2 + {s_i}^2}{8h_j}.$$

And finally the area of the circular-segment can be obtained from

$$A_j = \frac{{r_j}^2}{2}(\Theta_j - sin(\Theta_j)).$$

We can now write $A_{e_1+e_2}(\alpha)$ as

$$A_{e_1+e_2}(\alpha) = \frac{{r_1}^2}{2}(\Theta_1 - sin(\Theta_1)) + \frac{{r_2}^2}{2}(\Theta_2 - sin(\Theta_2))$$

We want to find the maximum of this function for $\alpha \in [0°, 60°]$. Bending e_1 more than 60° is not possible because intersections would be introduced since the vertical edge of the triangle is fixed and bending with a negative angle, which implies bending outwards, cannot lead to a maximum decrease as e_2 is blocked anyways by the vertical fixed edge when $\alpha = 0°$. On the given interval the two maxima are at $\alpha = 0°$ and $\alpha = 60°$. We calculate the value for $\alpha = 60°$, which implies $\Theta_1 = 120°$. The second term nullifies straightaway:

$$\begin{aligned}
A_{e_1+e_2}(60°) &= \frac{{r_1}^2}{2} \cdot (\Theta_1 - \sin(\Theta_1)) + 0 \\
&= \frac{\left(\frac{4{h_1}^2+{s_i}^2}{8h_1}\right)^2}{2} \cdot (\Theta_1 - \sin(\Theta_1)) \\
&= \frac{\left(\frac{4\left(\frac{s_i}{2}\tan(\frac{\Theta_1}{4})\right)^2+{s_i}^2}{8\frac{s_i}{2}\tan(\frac{\Theta_1}{4})}\right)^2}{2} \cdot (\Theta_1 - \sin(\Theta_1)) \\
&= {s_i}^2 \cdot \frac{(tan^4(\frac{\Theta_1}{4}) + 2tan^2(\frac{\Theta_1}{4}) + 1) \cdot (\Theta_1 - \sin(\Theta_1))}{32\tan^2(\frac{\Theta_1}{4})}
\end{aligned}$$

We set the constant $c = \frac{(tan^4(\frac{\Theta_1}{4})+2tan^2(\frac{\Theta_1}{4})+1)\cdot(\Theta_1-\sin(\Theta_1))}{32\tan^2(\frac{\Theta_1}{4})} = 0.20478283$. This is a constant term since we fixed α and thereby also Θ_1. So, the maximum area decrease of the equilateral triangle depends quadratically on the side length s_i multiplied with a constant factor. This is the property that we can exploit now, because due to our choice of s_i and c we get:

$$A_{e_1+e_2}(60°) = {s_i}^2 \cdot c = \sqrt{\frac{x_i}{c}}^2 \cdot c = x_i$$

Hence, the only way to achieve $\Delta t_i = -x_i$ is to choose an angle of $\alpha = 0°$ or $\alpha = 60°$ which corresponds to bending either e_1 fully or e_2 fully. No other combination of bendings of e_1 and e_2 will attain Δt_i. So, as in the previous proof, if the target area of R_0 can be met we can deduce a subset of the x_i where x_i is in the subset if and only if the edge of R_i which neighbors R_0 is bent. All x_i in this subset sum up to $\frac{1}{2}\sum_{1\leq i\leq n} x_i$ because no area can be added or subtracted from R_0 via the other edges thanks to the stable sea property. Equally all other x_i must sum up to $\frac{1}{2}\sum_{1\leq i\leq n} x_i$. The instance I has thus a solution.

$\Rightarrow$ The inverse proof direction (positive $I \Rightarrow$ positive I_{CACSS}) can be shown easily, similar to the proof for $CACP$ in Section 3.3.1. $\square$

Note that $CACSS$ is rather limited, since the sea and lakes cannot at all be deformed. It might be acceptable though in practice to change the exact form of the sea and lakes in order to obtain a correct cartogram. Therefore, we next present a more natural variant of CAC where this is allowed.

3.3.3 Limitation to Normal Bending Configurations

Circular-Arc Cartogram with Normal Bending ($CACNB$) is another attempt to simplify the Circular-Arc Cartogram problem by reducing the infinite set of valid circular-arc configurations with the introduction of a reasonable restriction. This time we impose that a valid bending configuration obeys the normal bending condition as defined in Section 3.2.2.

Problem 3.3.3 (Circular-Arc Cartogram with Normal Bending) *For a given polygonal subdivision S with target values t_i (positive real numbers) for each region of S decide if S admits a valid bending configuration where all areas of the deformed regions comply with the vector t and where the normal bending condition is respected. The bending of edges is thus limited in a natural way: Regions demanding a decrease of their area can only use inwards bent edges and regions demanding an increase of their area can only use outwards bent edges.*

When imposing this requirement it will be deducible straightaway from the shape of the countries in the resulting cartogram if they have increased or decreased their area. Countries are inflated or deflated as a whole depending on whether they want to grow or shrink and it is not possible to use a country in order to transfer area to another country by gaining area with one edge and transporting it to a neighbor with another. Contrary to $CACSS$ however modifications of the outer shape of the subdivision will be accepted.

Theorem 3.3.3 *CACNB is NP-hard*

Proof For the polynomial time reduction we use Bipartition again and this time we require that $\frac{\sum x_i}{2}$ is integer. Bipartition remains NP-hard in this setting because the answer is always 'no' if the condition does not hold.

For the reduction we only need a slight modification to the previous construction of the subdivision used for $CACSS$. The goal is again to have triangles where only two edges are candidates for being bent. However, in this case here we do not have the possibility to use lakes and say that all edges belonging to a lake or to the sea cannot be bent. Instead we can exploit the normal bending property in order to simulate lakes with a certain trick: The edges that we want to fix will border two regions both of which demand an area decrease. Consequently it is not possible to bend the corresponding edge in either direction and we can apply the same reasoning as before.

The construction of an instance I_{CACNB} will be exactly the one we used before except that all lakes are replaced with auxiliary regions called A_i. So in the horizontal chain after a pair of triangles there will be a region A_i which serves as a buffer between two triangles and shares exactly the left and right edge with these triangles. The form of A_i is thus trapezoidal. The exact number of auxiliary regions between triangles is given by $m = \lfloor \frac{n}{2} \rfloor$. Additionally one or two (depending on the parity of n) auxiliary regions A_0 and A_{m+1} will serve as buffer between R_1 and the sea and between R_n and the sea.

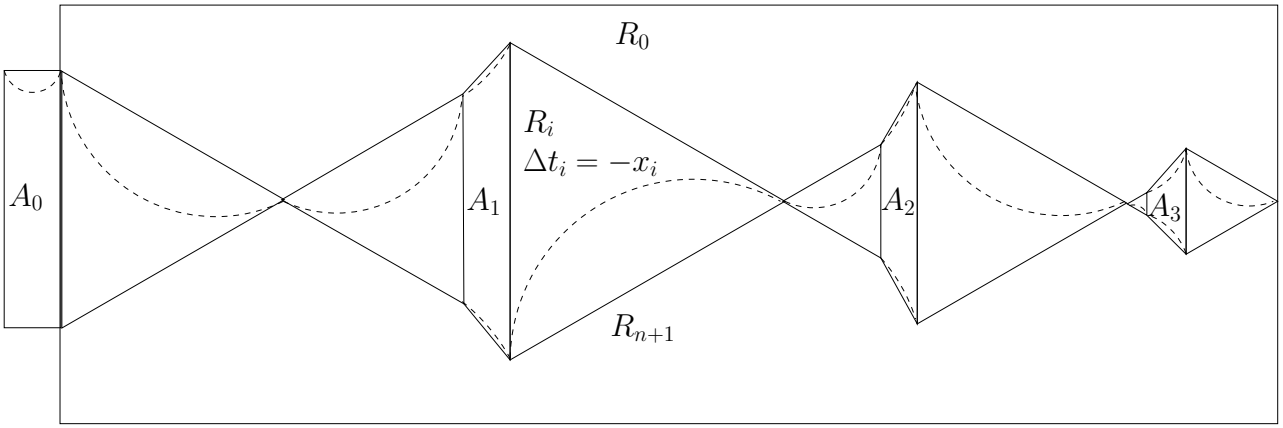


Figure 3.8: Construction of the subdivision for CACNB and a valid configuration (dashed arcs)

Figure 3.8 depicts an example of this construction. The dashed circular arcs illustrate a possible valid bending configuration which satisfies all weights.

By giving negative target area changes to the A_i for $1 \leq i \leq m$ and to the R_i for $1 \leq i \leq n$ we can then conclude that every R_i has one edge that cannot be bent, namely the one that it shares with its adjacent auxiliary region. However, the target values have to be chosen carefully so that they affect the area change of R_0 and R_{n+1} only very slightly. We want this disturbance to be sufficiently small in order to still be capable of deducing a unique assignment of the x_i to one of two subsets. So, the effect that the area change of the A_i has on the area change of R_0 and R_{n+1} should only be a tiny noise which we can clearly distinguish from the area change emerging from the R_i for $1 \leq i \leq n$.

We set the following target area change values Δt and will see later that they are suitable for our goal:

- $\Delta t_{R_i} = -x_i$ for $i = 1 \dots n$
- $\Delta t_{A_i} = -\frac{1}{n}$ for $i = 1 \dots m$
- $\Delta t_{R_0} = \Delta t_{R_{n+1}} = \frac{1}{2} \left(\sum\limits_{1 \leq i \leq n} \Delta t_{R_i} + \sum\limits_{1 \leq i \leq m} \Delta t_{A_i} \right)$
- $\Delta t_{A_0} = -0.1$ $(= \Delta t_{A_{m+1}}$, if existing)

We can now show that the instance of Bipartition has a solution if and only the derived instance of $CACNB$ has a solution.

$\Leftarrow$ Let first I_{CACNB} be positive. Thus a valid bending configuration C exists. In C each triangle R_i has one edge that cannot be bent since all R_i and A_i demand an area-decrease which disallows bending the common borders inwards. In fact this mechanism is the only purpose of the A_i. Therefore (reasoning as in the proof of Theorem 3.3.2) only a full bending of either the upper or the lower triangle edge can satisfy $R_i^{'}s$ area-demand. The total increase demanded by R_0 and R_{n+1} has the same value as the total decrease demanded by the R_i $(i = 1 \dots n)$ and the A_i $(i = 1 \dots m)$ together and also only R_0 and R_{n+1} can absorb the area removed from R_i $(i = 1 \dots n)$ and A_i $(i = 1 \dots m)$. So the six edges of R_0 and R_{n+1} bordering the sea cannot be bent in a valid configuration, since that would increase the area of R_0 or R_{n+1} and could not be compensated by any other edge.

Furthermore we have

$$\sum_{1 \leq i \leq m} \Delta t_{A_i} = \frac{m}{n} < 1.$$

This auxiliary decrease or noise caused by the A_i $(i = 1 \dots m)$ will be absorbed by R_0 and R_{n+1} and is a non-integer value between 0 and 1. Therefore the integer part of Δt_{R_0} must be $\frac{1}{2} \sum_{1 \leq i \leq n} \Delta t_{R_i}$, which is integer according to our requirement placed on the Bipartition instance. This integer part can only come from the R_i whose upper edges are bent downwards. The same holds for R_{n+1}. As before the set of all x_i where R_i's upper edge was used for bending in C is a solution for I.

$\Rightarrow$ The inverse equivalence direction can be established easily: For a given partition of I in two sets of equal total value we bend the upper edges of R_i where x_i belongs to the first set and we bend the lower edges of R_i where x_i belongs to the second set. Then for each A_i we bend the upper and lower edge simultaneously until attaining the desired area decrease of $\frac{1}{n}$ in order to partition the absorption of the decrease from the A_i equally between R_0 and R_{n+1}. The decrease of A_0 and A_{n+1} can be achieved no matter how with the three edges bordering the sea. We have attained all target areas without introducing any intersections or modifying the adjacencies. So the subdivision has a valid configuration and I_{CACNB} is thus a positive instance.

Note that of course in the construction we have to choose suitable dimensions for the A_i in order to realize their area decrease. But this is always possible as their width can be made arbitrarily big. □

3.3.4 The Unrestricted CAC Problem

CAC is the the problem where all previous constraints we imposed (like stable sea or normal bending) are omitted. Circular arcs can then be created on any edge and anyhow as long as no intersections occur. This problem was already introduced as Problem 3.1.1. It is equivalent to $CACP$ when choosing $k = n$ where n is the number of edges in the subdivision.

Theorem 3.3.4 *CAC is NP-hard*

Proof We use Subset sum for the polynomial time reduction. Let $I = (x_1, \dots, x_n, K)$ with all x_i and K integer be a Subset sum instance. We can assume that $n \geq 2$, otherwise the answer is trivial. We transform I into $I' = (x'_1, \dots, x'_n, K')$ with $x'_i = (n+1)x_i$ and $K' = (n+1)K$. Clearly I' is a positive instance if and only if I is positive. For a given solution of I' or I we simply factor out or multiply by $(n+1)$ and obtain a solution for I respectively I'.

The construction of the subdivision from I' builds on the construction used for $CACP$ in Section 3.3.1, except that here we introduce gaps between the rectangles: Between two R_i we add a horizontal line that connects two of their upper vertices. The length of this line is not important, we can simply set it to 1.

The dimensions of R_i (for $1 \leq i \leq n$) are $h_i = \sqrt{\frac{2x'_i}{\pi}}$ (height) and $w_i = 2\sqrt{\frac{2x'_i}{\pi}}$ (width). The target area decrease is chosen to be $\Delta t_i = -x'_i$. The R_i with $1 \leq i \leq n$ are enclosed from the top by R_0 which has target area increase $\Delta t_0 = K'$. Concerning the dimensions of R_0 we only impose that its height is smaller than its width. Until this point the construction is pretty much like the one used for $CACP$, except the introduction of gaps and the rectangles being vertically aligned along their upper horizontal edge. All the components

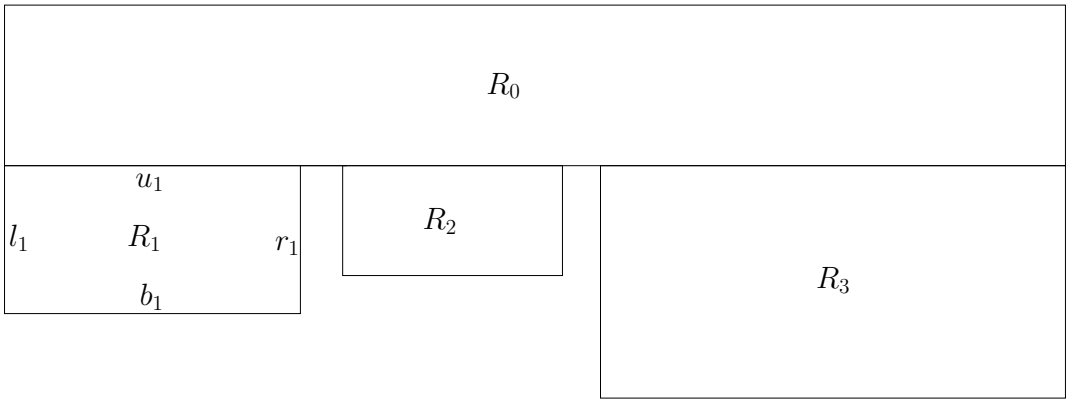


Figure 3.9: Construction of the basic skeleton of the subdivision

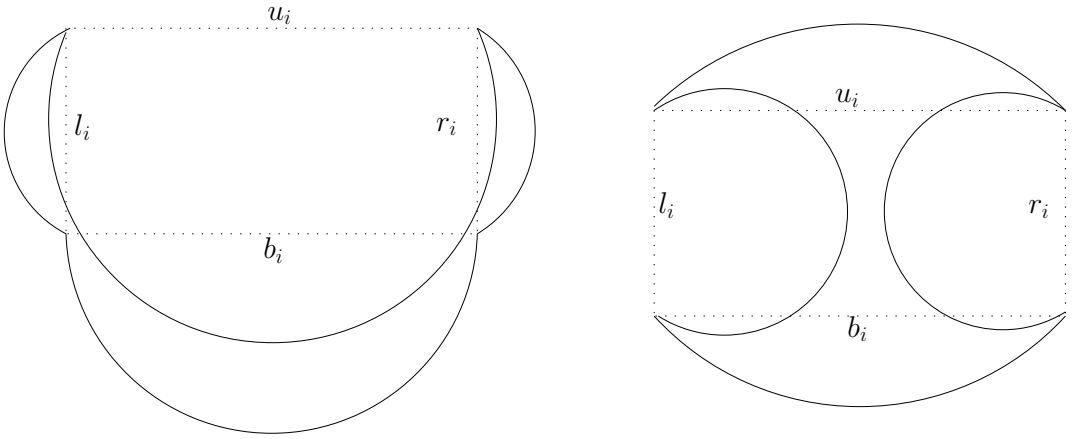


Figure 3.10: Arbitrary deformations can achieve desired area decrease of R_i

of the subdivision described until now form its basic skeleton an example of which is depicted in Figure 3.9. From now on we denote the upper, lower, right and left edge of R_i by u_i, b_i, r_i, l_i respectively.

If we left the construction like this we would not be able to draw any relevant conclusions from valid bending configurations because all we know about a configuration is that the bending of the four surrounding edges of R_i must lead to the desired area decrease. However, the distribution of this shrinkage among u_i, b_i, r_i and l_i is uncontrollable. For example we might obtain some kind of sickle or hammer (or even worse shapes) with correct areas as depicted in Figure 3.10.

In order to remedy this arbitrariness we will add two kinds of gadgets. Let us define the set E to contain all $4n + 3 + (n - 1)$ edges that occur in the construction so far, i.e. all u_i, b_i, r_i and l_i, the upper, right and left edge of R_0 and the connectors between R_i and R_{i+1}. For every edge $e \in E$, except the u_i and b_i, we introduce a new vertex on both sides of e which is centered with regard to e and connected to the two end-vertices of e. This implies two new regions with a triangle shape which enclose e from both sides. These new regions all demand no area change ($\Delta t_R = 0$).

The purpose of the two new vertices around e is to block e when being bent in either direction, let us call these vertices 'blocking vertices'. We can guarantee that e can only swing in a certain interval limited by the blocking vertices. By choosing the distance of the blocking vertices to their associated edges sufficiently small we can later categorize

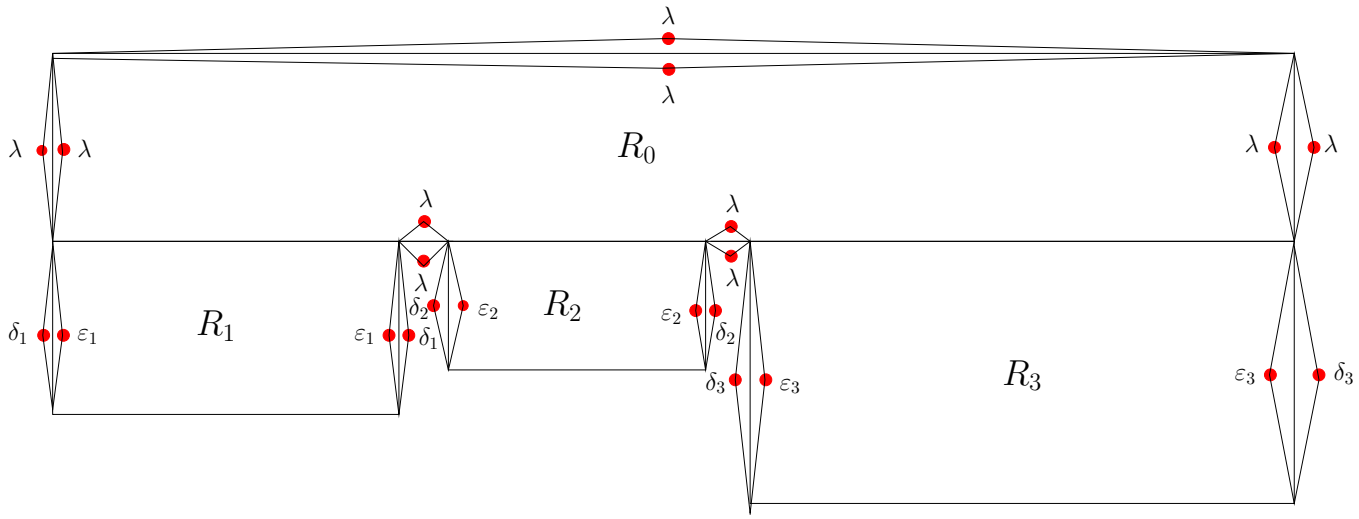


Figure 3.11: Construction with additional blocking vertices

the set of all R_i in a valid bending configuration in exactly two disjoint subsets which will enable the assignment of all x_i to one of two subsets with equal total value.

For R_i we denote ε_i the distance of the two inner blocking vertices from their associated edges, i.e. the vertex lying on the left side of r_i and the vertex lying on the right side of l_i. The distance of the two outer blocking vertices, i.e. the vertex lying on the right side of r_i and the vertex lying on the left side of l_i, is δ_i. All other blocking vertices that do not belong to an edge of R_i with $1 \leq i \leq n$ have a distance of λ to their associated edge. An example of this extended but still not final construction is depicted in Figure 3.11. The positions of blocking vertices are highlighted with red circles and the distances are indicated.

It remains to limit the bending of all u_i and b_i which have not yet been supplied with blocking vertices. The same gadget as before cannot be used here because isolating u_i or b_i from the sea, respectively from R_0, with triangle regions would limit the amount of area change that we can send over u_i or b_i too much. Any area change realized with u_i would also have to be realized with the two isolating triangle edges since the target area change of the triangles is 0. However, their capacity for bending is too restricted to achieve a change of Δt_i which is sometimes necessary, because they are shorter than u_i or b_i and block each other at their common angle. It is therefore necessary not to isolate u_i and b_i. Consequently we have to think of a different technique.

It is important that the u_i and b_i do not become isolated from R_0 respectively from the sea but at the same time there must be a vertex of the subdivision that blocks them at some point. For this purpose we will create $2n$ peninsulas of triangle form, one for each u_i and one for each b_i. The peninsula belonging to u_i has one vertex vertex centered above u_i with distance δ_i. The peninsula belonging to b_i has one vertex vertex centered below b_i with distance δ_i. The two other vertices of the peninsulas may be chosen in many different ways. For example we could use the existing vertices of the blocking triangles or we could introduce new vertices connected to the rest of the subdivision. Again these peninsulas have target area change $\Delta t = 0$ See Figure 3.12 for an illustration of the details at R_i. All vertices with blocking function are highlighted in red and their distance to the associated edge is given as well.

Now, it only remains to set the values of ε_i, δ_i and λ sufficiently small.

At first let us look at region R_i, which wants to shrink by $\Delta t_i = -(n+1)x_i \leq -3$ (we have $x_i \geq 1$ and $n \geq 2$). For a given bending configuration C we call bending zone of an edge e

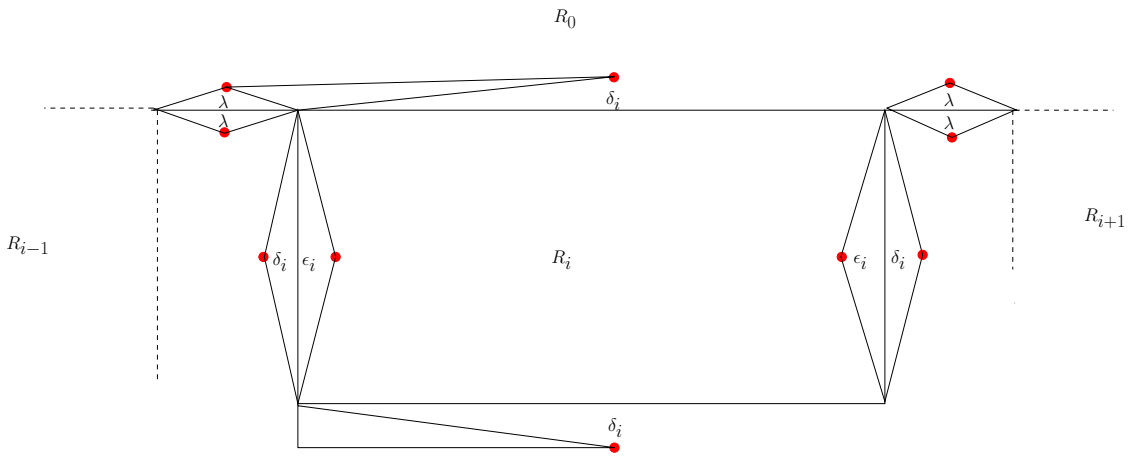


Figure 3.12: Details at R_i, in particular the two peninsulas blocking u_i and b_i

the zone that is enclosed by the circular arc replacing e and the original straight segment e itself. Then we can define ΔA_{u_i} (respectively $\Delta A_{b_i}, \Delta A_{r_i}, \Delta A_{l_i}$) to be the realized area change (positive or negative) in the bending-zone of u_i given by the symmetric difference of C and the initial subdivision S in that zone. Note that the realized area change in the bending zone of an edge can depend on several circular arcs and not only on the single one emerging from that particular edge. This is because circular arcs can lie in the bending zone of other edges. In a valid bending configuration where all target areas are matched it holds for every R_i that the achieved area change is the sum of the area changes in the bending zones of its edges:

$$\Delta t_i = \Delta A_{u_i} + \Delta A_{b_i} + \Delta A_{r_i} + \Delta A_{l_i}$$

With certain values of ε_i and δ_i we can claim that in the resulting setting either ΔA_{u_i} or ΔA_{b_i} must contribute the most significant part to the area decrease Δt_i. More precisely we will achieve that either ΔA_{u_i} or ΔA_{b_i} lies in the interval $(\Delta t_i - \frac{1}{4}, \Delta t_i + \frac{1}{4})$. The remaining difference to Δt_i will be contributed by $\Delta A_{u_i}, \Delta A_{l_i}$ and ΔA_{r_i} (or by $\Delta A_{b_i}, \Delta A_{l_i}$ and ΔA_{r_i}).

We set $\varepsilon_i = 0.01^{h_i}$. This will help us to guarantee the lower interval value of $\Delta t_i - \frac{1}{4}$. We denote by $A_e(h)$ the area of the circular segment created by bending edge e of length s until the circular arc reaches a particular height h:

$$A_e(h) = \frac{0.5 \arctan\left(\frac{2h}{s}\right)(4h^2 + s^2)^2 + hs(4h^2 - s^2)}{16h^2}$$

In a valid bending configuration u_i or b_i must be bent inwards more than $\frac{h_i}{2}$. Otherwise the bending configuration would not attain t_i even if we bend all remaining edges inwards as far as possible:

$$A_{r_i}(\varepsilon_i) + A_{l_i}(\varepsilon_i) + A_{u_i}(\frac{h_i}{2}) + A_{b_i}(\frac{h_i}{2}) < |\Delta t_i|$$

So without loss of generality let u_i be bent inwards more than $\frac{h_i}{2}$ in the following. What is the minimum height h of the circular arc replacing u_i that enables finding a valid configuration? First note that if h is minimum, the circular arc of b_i must be bent as far as possible towards the circular arc of u_i, i.e. its height must be $h_i - h$. If this was not

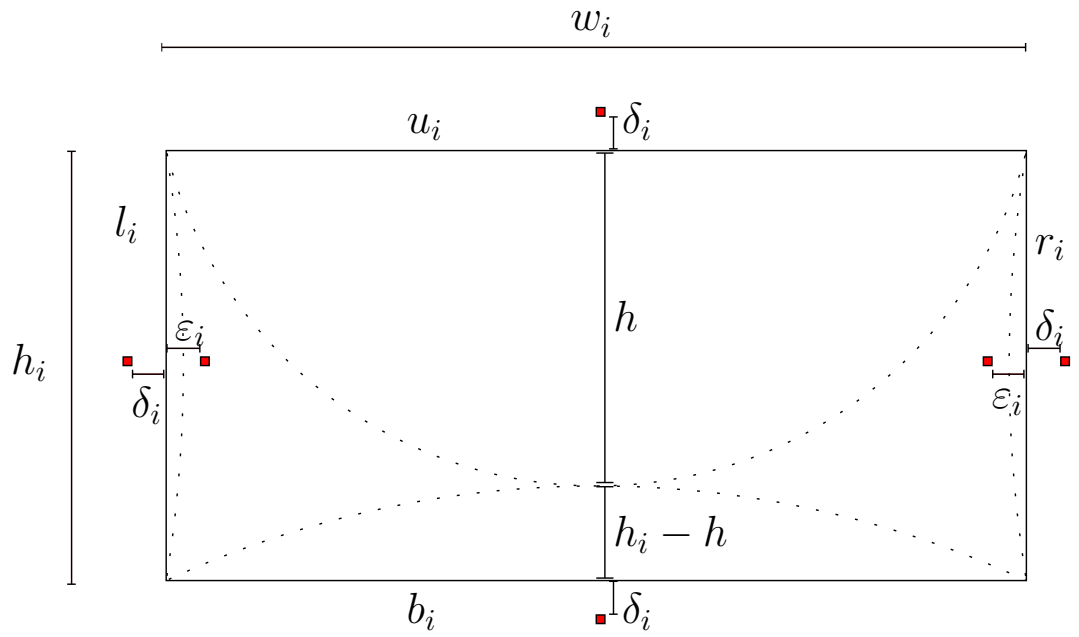


Figure 3.13: Possible circular-arc configuration for R_i

the case, we could reduce h by simultaneously bending u_i and b_i upwards with the right speed. See Figure 3.13 for an illustration.The blocking vertices are the red squares and the associated triangle edges are omitted in this image for a better readability.

Also note that $h \leq h_i$ as the simple half circle under u_i and no bendings at all at the other edges meets all requirements:

$$A_{u_i}(h_i) + A_{r_i}(0) + A_{l_i}(0) + A_{b_i}(0) = |\Delta t_i|$$

So we are searching for $h \in (\frac{h_i}{2}, h_i]$. The difference of $A_{u_i}(h) + A_{b_i}(h_i - h)$ to Δt_i can only be compensated by the decrease from bending r_i and l_i. The maximum capacity for compensation by r_i (or l_i) until it intersects its inner blocking vertex is:

$$\begin{aligned} A_{l_i}(\varepsilon_i) &= A_{r_i}(\varepsilon_i) \\ &= \frac{0.5 \arctan\left(\frac{2 \cdot 0.01^{h_i}}{h_i}\right)(4 \cdot 0.01^{2h_i} + h_i^2)^2 + 0.01^{h_i} h_i (4 \cdot 0.01^{2h_i} - h_i^2)}{16 \cdot 0.01^{2h_i}} \end{aligned}$$

So $\Delta t_i - (A_{l_i}(\varepsilon_i) + A_{r_i}(\varepsilon_i))$ is the minimum area decrease that has to be done with u_i and b_i. For a choice of $h^* = h_i \cdot 0.9^{(0.5^{h_i})}$ this area decrease cannot be achieved. Therefore $h > h^*$. At the same time we have $A_{u_i}(h^*) \geq \Delta t_i - \frac{1}{4}$. Plus $A_{u_i}(h)$ is monotonously increasing on the interval $(\frac{h_i}{2}, h_i]$, which means that $A_{u_i}(h) \geq \Delta t_i - \frac{1}{4}$ This concludes the proof of: $\Delta A_{u_i} \geq \Delta t_i - \frac{1}{4}$ in a valid configuration (respectively ΔA_{b_i}).

Guaranteeing the upper interval value of $\Delta t_i + \frac{1}{4}$ is more easy. We need to ensure that by bending any three edges of R_i outwards one can at most gain an area of $\frac{1}{4}$. If this is the case the area decrease caused by a single edge of R_i is bounded by $\Delta t_i + \frac{1}{4}$, because if it were more we could not compensate the surplus with the other three edges. We set

$\delta_i = 0.1^{h_i}$. A calculation shows that $A_{l_i}(\delta_i) + A_{r_i}(\delta_i) + A_{u_i}(\delta_i) \leq \frac{1}{4}$ (for $h_i \geq \sqrt{\frac{6}{\pi}} \Leftrightarrow x'_i \geq 3$ which is the case because of our condition $n \geq 2$ on the original instance I).

Now we consider R_0. The upper, right and left boundary of R_i in the original edge set E (before introducing blocking vertices) were single edges and the lower boundary consisted of $2n-1$ edges (n upper R_i edges and $n-1$ connectors between the R_i). So we had a total of $2n+2$ edges that defined R_0. Now these edges in E are isolated from R_0 by the new edges E' added when introducing blocking vertices. However, the area change of R_i is still bounded by the edges in E since they do entirely enclose R_0: Every area change of R_0 has to flow over the original edge set E. We will exploit this property and limit the area change of R_0 by limiting the area change of the old boundaries of R_0. For n of these edges we have already established the distances δ_i of their stoppers (namely for the u_i, which only have one stopper).

The remaining edges all have two blocking vertices and their distances will be set in such a way that edge bending can vary the area in either direction only in the interval $(-\frac{1}{4}, +\frac{1}{4})$ for each single one of them. We will achieve this for the topmost edge of R_0 in E which is the longest edge overall and has length $\sum_{1 \leq i \leq n} w_i + (n-1)$, with the initial assumption that all auxiliary edges connecting two R_i have length 1. Since all other edges are shorter the same value λ will be small enough for them as well. A sufficiently small value is $\lambda = 0.5^{\sum_{1 \leq i \leq n} w_i + (n-1)}$.

Let us sum up what we know so far about a valid bending configuration where all target areas are matched. We have specified the exact positions of all blocking vertices which help us to guarantee the following four properties:

1. Every R_i gets a portion of $A \in (\Delta t_i - \frac{1}{4}, \Delta t_i + \frac{1}{4})$ of its decrease from either its upper or lower edge.
2. The three remaining edges of R_i compensate the difference to Δt_i, and each single one can vary only in the interval $(-\frac{1}{4}, +\frac{1}{4})$.
3. Every other edge that does not make part of a R_i can cause an area change of R_0 in the interval $(-\frac{1}{4}, +\frac{1}{4})$.
4. The new triangle shapes always lie completely in another region and they demand no area change. So they do not affect the feasibility of the given weights in the subdivision.

Having this we can prove that I is a positive instance if and only if the derived subdivision has a solution:

$\Rightarrow$ Let I be a positive instance of Subset Sum ($n \geq 2$ and $K \geq 1$, otherwise trivial) and S be a subset with total value K. I' is positive if and only if I is positive because the same subset would be a solution. It is straightforward to see that if I' is positive, then the constructed subdivision admits a valid configuration: Bend all u_i inwards until they form half-circles where $x'_i \in S$ and do likewise for all b_i where $x'_i \notin S$. This configuration clearly satisfies all target areas and therefore I_{CAC} is a positive instance as well.

$\Leftarrow$ Now, assume that the subdivision is a positive instance. Hence, it admits a valid configuration C. We define the set U as all i where u_i is dominant in C, i.e. where the circular arc replacing u_i accounts for the main part of the area decrease of R_i lying in the interval $(\Delta t_i - \frac{1}{4}, \Delta t_i + \frac{1}{4})$.

The at most $2n+2$ edges in E enclosing R_0 without the u_i where $i \in U$ can cause an area-increase of strictly less than $(2n+2) \cdot \frac{1}{4} = \frac{n+1}{2}$. Equally they can cause an

area-decrease of strictly less than $\frac{n+1}{2}$. Thus we have the interval $I_1 = (-\frac{n+1}{2}, +\frac{n+1}{2})$ for the variation of the area of R_0 without the edges in U.

The at most n edges u_i with $i \in U$ can cause an area change of R_0 within the interval

$$\begin{aligned} I_2 = \left(\sum_{i\in U}(\Delta t_i - \frac{1}{4}), \sum_{i\in U}(\Delta t_i + \frac{1}{4})\right) \subseteq \\ \left(\sum_{i\in U}\Delta t_i - n\frac{1}{4}, \sum_{i\in U}\Delta t_i + n\frac{1}{4}\right) \subseteq \\ \left(\sum_{i\in U}\Delta t_i - \frac{n+1}{2}, \sum_{i\in U}\Delta t_i + \frac{n+1}{2}\right) \end{aligned}$$

The total variation of the R_0 area lies thus in the sum of the two interval I_1 and I_2:

$$I_1 + I_2 \subseteq \left(\sum_{i\in U}\Delta t_i - (n+1), \sum_{i\in U}\Delta t_i + (n+1)\right)$$

We conclude that $K' = \sum_{i\in U}\Delta t_i$ since K' is a multiple of $n+1$ and so is $\sum_{i\in U}\Delta t_i$. The deviation from $\sum_{i\in U}\Delta t_i$ in the interval is not big enough to attain any multiple of $n+1$ other than K' because all interval boundaries are open. So, the x_i with $i \in U$ form a subset of value K which implies a solution for I. □

4. Relaxations

The previous Chapter 3 presented NP-hardness results for some variants of the Circular-arc Cartogram problem. At first we looked into $CACP$ which is the most general variant among the ones considered here and we proved it to be NP-hard. Then we focused on the unrestricted CAC problem where we allow bending any edge of the subdivision without any further constraints as long as intersections are avoided. Besides its proof of NP-hardness we came up with proofs for two variants: $CACSS$ where no modifications to the shape of the non-land areas (seas and lakes) of the subdivision are allowed and $CACNB$ where unnatural edge bending is disallowed, i.e. bending an edge inwards although the size of the region needs to be increased or vice versa.

Now, it makes sense to look for reasonable relaxations of the problem that enable finding a practically satisfying cartogram generation algorithm. Possible approaches are:

- Relaxation of adjacencies
- Relaxation of areas
- Relaxation of vertex related constraints

However, in most applications adjacencies must be preserved as this is an essential structural property of any map which one does not wish to disturb. Compared to that the exact shape and position of regions is a less important quality criterion. As long as shapes are not distorted too badly and one is still able to recognize every region in the map it is acceptable to have deformations of regions. Plus, when allowing the replacement of straight segments with curved shapes (e.g. circular arcs) one accepts already that shapes are distorted and the additional distortion from vertex movements is likely to remain acceptable as well.

So, the most reasonable and practically satisfying relaxation might be the relaxation of the vertex positions. There are several possibilities: A first approach might be to ignore vertices of degree two, which means that we can replace several straight line segments which lie on the same border between two faces of the subdivision by one circular arc or – in other words – merge two circular arcs into one arc. Another alternative would be to allow the movement of vertices. Area relaxations where the error which we allow for each region is bounded are also worth a consideration. Of course combined relaxations of area and vertex constraints are thinkable as well.

In this chapter we consider the above mentioned relaxation strategies. Area constraint relaxations are discussed first in Section 4.1 and another NP-hardness result is presented.

The topic of the second Section 4.2 is dropping vertex constraints. In the last Section 4.3 we propose a combined relaxation approach which lies the theoretical basis for the implementation and the case study presented in the following Chapter 5.

4.1 Area Relaxations

From the NP-hardness of all CAC variants considered so far it was concluded that appropriate relaxations of the area or vertex constraints are the only way to tackle the problem at hand. We are looking for a compromise which abandons exactness of the produced cartograms and gains computational efficiency.

An algorithm conceived on the basis of these relaxations should be able to provide cartograms in polynomial time but has no guarantee if the region's areas correspond to their prescribed weights. Here we will discuss a possible area relaxation which admits a fixed additive area-error per region and we show that the problem remains NP-hard in this easier setting.

4.1.1 CAC Area Relaxation

The adapted problem can be stated as follows: The input is as before a polygonal subdivision S with regions R_i of size A_i. Weights t_i are also provided which represent the target area for each region. Again it is allowed to bend the edges of S in order to satisfy the weights of all regions. However, we will not be so strict about the achieved areas. In the classic setting the constraint was that for all i the resulting area T_i of R_i after all bending has been done equals the target area:

$$T_i = t_i \tag{4.1}$$

The relaxation of this constraint with an absolute area-error is:

$$|T_i - t_i| \leq \gamma \tag{4.2}$$

where the maximum area-error γ is a positive real number $\gamma \in \mathbb{R}^+$. So, we allow a deviation of γ from the target area for each single region. All CAC decision problems can be defined with Constraint 4.2 replacing Constraint 4.1. We add the suffix WAT ('with area tolerance') to indicate this. A related relaxation is the one with an admitted relative area-error per region. We would then have a multiplicative factor instead of an absolute additive constant.

The question is if the absolute area-error relaxation changes anything about the complexity of the different problem variants?

As in the version without area-error we can find positive as well as negative instances:

- Positive instances are trivial as we can simply set the target areas equal to the initial areas, $t_i = A_i \Leftrightarrow \Delta t_i = 0$. Than a bending configuration with all radiuses set to $+\infty$ does the necessary because all resulting area-errors are 0.
- A negative instance, i.e. a subdivision S, a target vector t and a predefined maximum area-error γ as parameter where no bending of the edges is possible such that in the resulting configuration all areas differ less than γ from the corresponding component in t can also be constructed: We can take the example from Section 3.2.3 where a polygon was given, that has an upper bound on its area increase and set the target area of this polygon to a value higher than this upper area bound plus γ. Then all valid bending configurations have an error of more than γ and the instance is a negative one.

Let us now ask the question how hard it is to distinguish between negative and positive instances.

4.1.2 Complexity of CACPWAT

Theorem 4.1.1 *CACPWAT is NP-hard*

Proof We show that the polynomial time reduction used for the $CACP$ (see Section 3.3.1) can be reused. We will only add a phase of preprocessing the input before constructing the subdivision. So let $I = (x_1, \ldots, x_n)$ with all x_i integer be an instance of Bipartition with two additional properties:

1. $n > 1$
2. $\frac{1}{2} \sum_{1 \leq i \leq n} x_i \in \mathbb{N}$

If one of these two conditions is violated the instance is always negative. So Bipartition remains NP-complete under these assumptions.

We define a new instance of Bipartition as $I' = (x'_1, \ldots, x'_n)$ where $x'_i = (n+1)\lceil\gamma\rceil x_i$. Clearly I' is a positive instance if and only if I is positive: We have scaled all integers with the same factor $(n+1)\lceil\gamma\rceil$. A short calculation shows that the existence of a solution for I implies a solution for I' and vice versa.

The subdivision S is then constructed from I' following exactly the same rules as in the case of $CACP$ without area-error (see Figure 3.5). We also set the same target area change vector Δt and parameter $k = n$ as before.

In the following A_i is the initial size of region R_i, T_i is the attained area in a valid configuration after bending and ΔT_i is the implied area change: $\Delta T_i = T_i - A_i$. As before we know that for each rectangular region R_i exactly one edge must be bent inwards. The justification was that a total of n edges can be bent, at least one edge must be bent for each region R_i and none of the regions R_i share any edges. From the construction and choice of the dimensions of R_i it followed that only a full bending of either the lower or the upper edge can achieve exactly the target area decrease of $\Delta t_i = -x'_i$. This is also the largest decrease we can get because bending the lower or upper edge further would introduce intersections and because the right and left edge are shorter and have the same blocking angle of 90° as the upper and lower edge. So, in our consideration we can already exclude the case where the area-error is positive. We always have $T_i \leq t_i$. It might be possible though to use the left or right edge to attain an area within an accepted deviation from t_i: $T_i \geq t_i - \gamma$. Let us investigate if this is really possible.

If the right edge – respectively the left edge – of R_i is fully bent then the circular segment is a half-circle which subtracts a maximum area of

$$\Delta T_i = \frac{\left(\frac{h_i}{2}\right)^2 \pi}{2} = \frac{\left(\sqrt{\frac{2x'_i}{\pi} + \frac{2}{\pi}}\right)^2 \frac{\pi}{4}}{2} = \frac{x'_i + 1}{4} = \lceil\gamma\rceil \frac{(n+1)x_i}{4} + \frac{1}{4}.$$

With the initial condition $n \geq 2$ we have $(n+1)x_i \geq 3$. Therefore, the maximal decrease with the right or left edge of R_i is bounded as follows:

$$\begin{aligned}
\Delta T_i &\leq \lceil\gamma\rceil \frac{(n+1)x_i}{4} + \frac{1}{4} < \lceil\gamma\rceil \frac{(n+1)x_i}{4} + \lceil\gamma\rceil \frac{(n+1)x_i}{4} \\
&= \lceil\gamma\rceil \frac{(n+1)x_i}{2} = \lceil\gamma\rceil \left((n+1)x_i - \frac{(n+1)x_i}{2}\right) \\
&< \lceil\gamma\rceil \left((n+1)x_i - 1\right) = \lceil\gamma\rceil (n+1)x_i - \lceil\gamma\rceil \\
&\leq \lceil\gamma\rceil (n+1)x_i - \gamma = \Delta t_i - \gamma
\end{aligned}$$

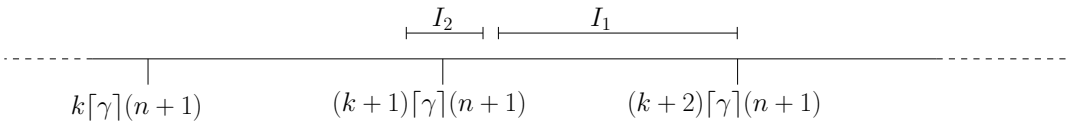


Figure 4.1: Intervals I_1 and I_2 on the number ray when $\frac{1}{2}\sum_{1\leq i\leq n} x'_i$ and $\sum_{i\in U} x'_i$ are different multiples of $\lceil\gamma\rceil(n+1)$

This is equivalent to $T_i < t_i - \gamma$. Hence, we cannot achieve an error of less than γ when bending the right or left edge of R_i and not introducing intersections. Therefore in a valid configuration we are obliged to bent either the upper or the lower edge of R_i inwards far enough to obtain an area change in the interval $[x'_i - \gamma, x'_i]$.

Now, in a valid bending configuration consider the set U of all i where $R_i's$ upper edge has been used. Note that the cardinality of U is at most $n-1$. If it was n then no lower R_i edge would be bent and the area of R_{n+1} would not change which is not valid because its target area change is always positive $\Delta t_{n+1} > 0$. The absolute value of the total decrease of the regions with $i \in U$ lies within the interval

$$\begin{aligned} I_1 &= \left[\sum_{i\in U}(x'_i - \gamma), \sum_{i\in U} x'_i\right] \\ &\subseteq \left[\sum_{i\in U} x'_i - (n-1)\gamma, \sum_{i\in U} x'_i\right]. \end{aligned}$$

This decrease equals exactly the increase of R_0 which according to Constraint 4.2 has to lie in the interval

$$I_2 = \left[\frac{1}{2}\sum_{1\leq i\leq n} x'_i - \gamma, \frac{1}{2}\sum_{1\leq i\leq n} x'_i + \gamma\right].$$

But now $\frac{1}{2}\sum_{1\leq i\leq n} x'_i$ as well as $\sum_{i\in U} x'_i$ are integer multiples of $\lceil\gamma\rceil(n+1)$. This leads us to the final conclusion that

$$\frac{1}{2}\sum_{1\leq i\leq n} x'_i = \sum_{i\in U} x'_i.$$

Assume that this was not the case. So the difference between $\frac{1}{2}\sum_{1\leq i\leq n} x'_i$ and $\sum_{i\in U} x'_i$ is bigger than $\lceil\gamma\rceil(n+1)$. But then the two intervals I_1 and I_2 do not overlap. This is a contradiction because the area change being shipped over the upper edges of all R_i in U must lie in both intervals, their intersection therefore be non-empty. See Figure 4.1 for an illustration of the two non-intersecting intervals on the number ray which is the contradictory case.

The precedent conclusion $\frac{1}{2}\sum_{1\leq i\leq n} x'_i = \sum_{i\in U} x'_i$ means that the x_i with $i \in U$ are a solution for the instance of Bipartition. □

4.1.3 Area Relaxation of the other CAC Variants

In the previous subsection NP-hardness was shown for the area relaxation $CACPWAT$ of one particular CAC variant. For the polynomial time reduction we used a factor scaling technique of the input instance of Bipartition. The scaling factor has to be chosen big enough in such a way that the imprecision caused by area-errors is always kept small compared to the demanded area changes in order to be able to distinguish this disturbance from the really relevant ongoing area change between the inner R_i and the two outer regions R_0 and R_{n+1}.

For the other variants of CAC that we defined in Section 3.3 the same technique will work as well. Of course the different topology of the constructions requires other scaling factors, but the principle remains the same. We can therefore assume that the other area relaxed variants of CAC are NP-hard, too. This kind of area relaxation with an additive constant γ is therefore not sufficient with regard to an easy computable method resolving CAC problems optimally. Possible area relaxations with a multiplicative factor are more promising than the one with an additive constant which we proposed here. Their exploration is an open research direction for the future and especially approximation algorithms or even approximation schemes could bring the efficient resolution of CAC problems very much forward.

4.1.4 Area-error Optimization Problem

$CACPWAT$ is directly related to the following optimization problem:

Problem 4.1.1 (CAC Optimization Problem) *For a given polygonal subdivision S with target values t_i (positive real numbers) for each region of S and an integer k, find an optimal valid bending configuration C of S (no intersections, adjacencies preserved and at most k edges bent) with the following minimization criteria:*

$$minimize \quad \max_{1 \leq i \leq n} (|T_i - t_i|)$$

where T_i is the area of region R_i in the bending configuration C. Hence, the maximum area-error of a region in S has to be minimized.

This is the optimization version of $CACP$ and we know that it must be NP-hard, unless $P = NP$. Otherwise we could resolve $CACPWAT$ and even $CACP$ in polynomial time.

4.2 Relaxation of Vertex-related Constraints

The relaxation of regional areas with an absolute constant is not a sufficient relaxation as we have seen in the previous section. The problem remains NP-hard if we allow an arbitrary additive maximum area-error per region. Since our policy is to preserve adjacencies in any case, we are left with the possibility to relax the vertex positions in the subdivision. That way we are not anymore limited to changing the shape of single regions locally in the subdivision by replacing straight line segments with circular arcs while keeping all vertices at their original position, but we can also significantly change the global topology of the map by moving the vertices anywhere as long as we preserve adjacencies or even by removing certain vertices. For example relative positions of the regions or their aspect ratio can be modified almost arbitrarily when moving the vertices which was not possible with the circular arcs only approach where the fixed vertex positions helped to preserve the shape of the polygon at least roughly.

In Subsection 4.2.1 a definition of the vertex movement relaxation is stated. Then we show that this relaxation solely is not sufficient in the respect that negative instances still do exist, see Subsection 4.2.2. Whether or not the problem is NP-hard as well remains open. Other vertex related relaxations are touched upon in Subsection 4.2.3.

4.2.1 Moving Vertices

Given a polygonal subdivision S – let us identify S with a plane graph $G = (V, E)$ – and a weight function $t : F \mapsto \mathbb{R}^+$ which assigns positive weights to the faces of G. The question we pose is: Does a plane graph G' exist whose face areas realize the weight function t and which is obtained from G by applying the following rule *R1* as often as required?

R1: move vertex $v \in V$ to any other position in the plane as long as the action of moving does not cross an edge or another vertex in the subdivision

Any graph obtained from G by applying this rule has the same dual graph as G and has the same complexity as G for each face if we define complexity of a polygon as the number of vertices on its boundary. So this corresponds exactly to the initial idea of transforming a subdivision by moving its vertices around. In the literature graphs which can be obtained from each other with this rule are said to be homotopically equivalent.

This definition can also be restated more formally with the notion of combinatorial embeddings. A combinatorial embedding defines for any vertex of the graph the cyclic order of all edges incident to that vertex. The set of all of these cyclic orders is called a rotation system. A topological embedding is more precise in the sense that it fixes the exact positions of all vertices in the plane. Thus a topological embedding is associated with an unique combinatorial embedding, but a combinatorial embedding can correspond to an infinite number of topological embeddings. Combinatorial embeddings form an equivalence relation of planar graphs, see [BETT99] for more detail.

So, our problem can be reformulated as follows:

Problem 4.2.1 *Given a planar graph G with a topological embedding E. Is it possible to find another topological embedding T that satisfies all prescribed face areas and that is combinatorially equivalent to E, i.e. the circular ordering of the incident edges is the same as in the original topological embedding E at all vertices?*

4.2.2 Negative Instance

Of course it is of high interest if the presented rule *R1* is sufficient to realize any weights on arbitrary graphs. To prove the contrary an embedded graph G has to be found together with weights assigned to its faces such that no drawing of G that has the same combinatorial embedding can attain the required weights.

And indeed such a graph can be found. Consider G and its given topological embedding which is depicted in Figure 4.2 with seven faces A, B, C, D, E, F, H, all of them triangles. We set the following weights:

- $t(A) = t(B) = t(C) = t(H) = 1$
- $t(D) = t(E) = t(F) = 3$

This graph has one important property: It is 3-layered in the sense that the layer-1 faces A, B, C are the only ones that share edges with the outer-face. Then layer-2 faces D, E, F share no edges with the outer-face but with the first layer and with H. Hence, H is the 3rd layer and can only see the layer-2 faces.

Now, the weights that we have set require all layer-2 faces to be big and all layer-3 and layer-1 faces to be relatively small. And this is the intuitive reason why this instance is not feasible. However, we need to establish a geometrically correct proof of this.

In the following for a polygon P we denote $a(P)$ the area of P and for an edge e we denote $|e|$ the length of e. Assume that we are given a topological embedding of G such that all

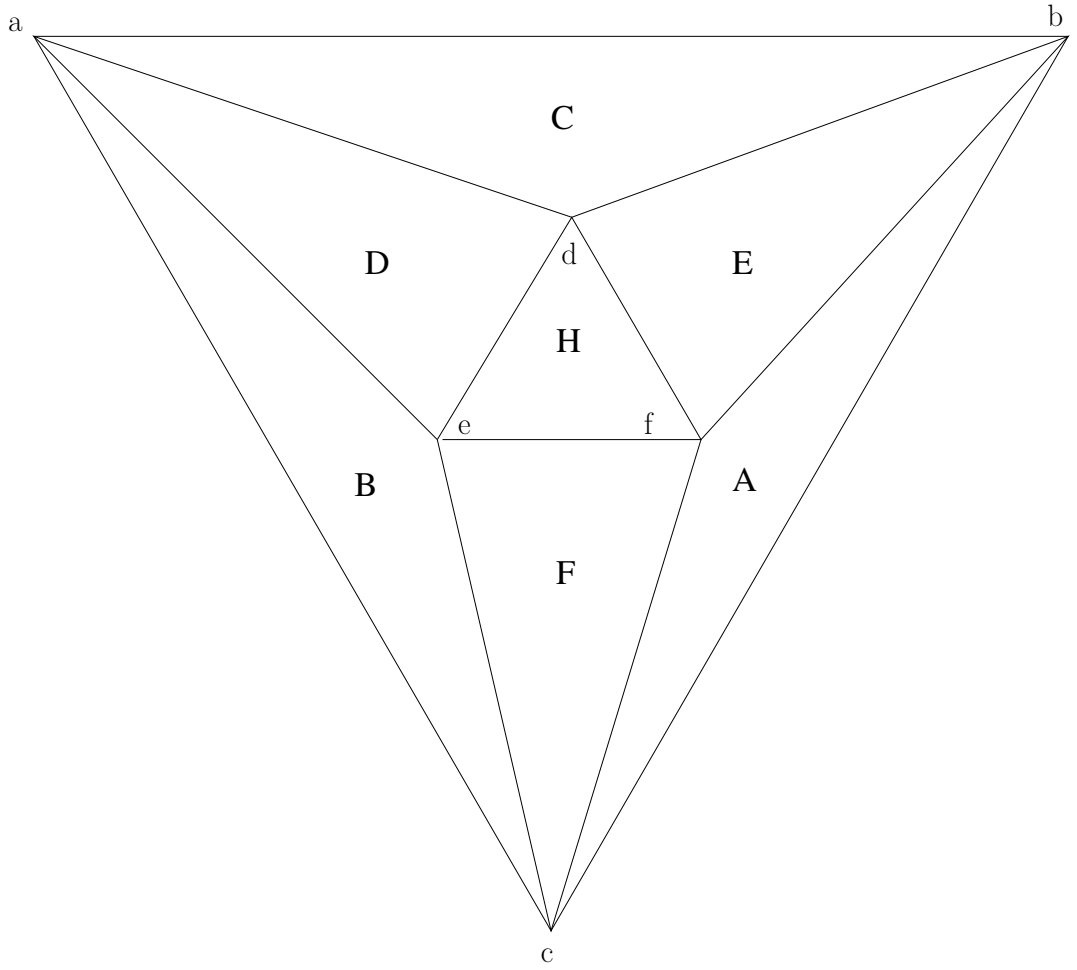


Figure 4.2: Graph G

weights are realized. We will construct a partition of all layer-2 face and give area bounds for the subfaces in the partition. Then a theorem going back to Debrunner ([Deb56]) can be applied which leads us to a contradiction.

For any topological embedding of G with the same rotation system we define the rays $\overline{ab}', \overline{bc}', \overline{ca}'$ to be parallel to the outer edges $\overline{ab}, \overline{bc}, \overline{ca}$ moved until they touch the inner vertices d, f, e (the rays are illustrated as dashed lines in Figure 4.3).

The point x is defined as the intersection of $\overline{cb}'$ and $\overline{db}$. This intersection always exists because $\overline{db}$ and $\overline{cb}$ share the same endpoint b and $\overline{cb}'$ has been moved towards d. If $\overline{cb}'$ has even been moved beyond d then let x simply be identical to d. We now compare the triangle A with the triangle X formed by b,f and x. They both have the same height h which is the distance between $\overline{cb}'$ and $\overline{cb}$. The length of the base of A for this height is $|\overline{cb}|$ and the length of the base of X corresponding to the height h is $|\overline{xf}|$.

Lemma 4.2.1 $|\overline{xf}| < |\overline{cb}|$

Proof We prove that $|\overline{xf}| < |\overline{cb}|$ which means that the area of A is strictly larger than the area of X. To see this consider the piece of $\overline{cb}'$ which is bounded by $\overline{ab}$ and $\overline{ca}$. This straight segment must be shorter than $\overline{cb}$ since these two form a trapezoid with the connecting pieces of $\overline{ab}$ and $\overline{ca}$ where the sum of the inner angles at b and at c is less than 180°. Otherwise $\overline{ab}$ and $\overline{ca}$ wouldn't approach each other and the outer triangle consisting

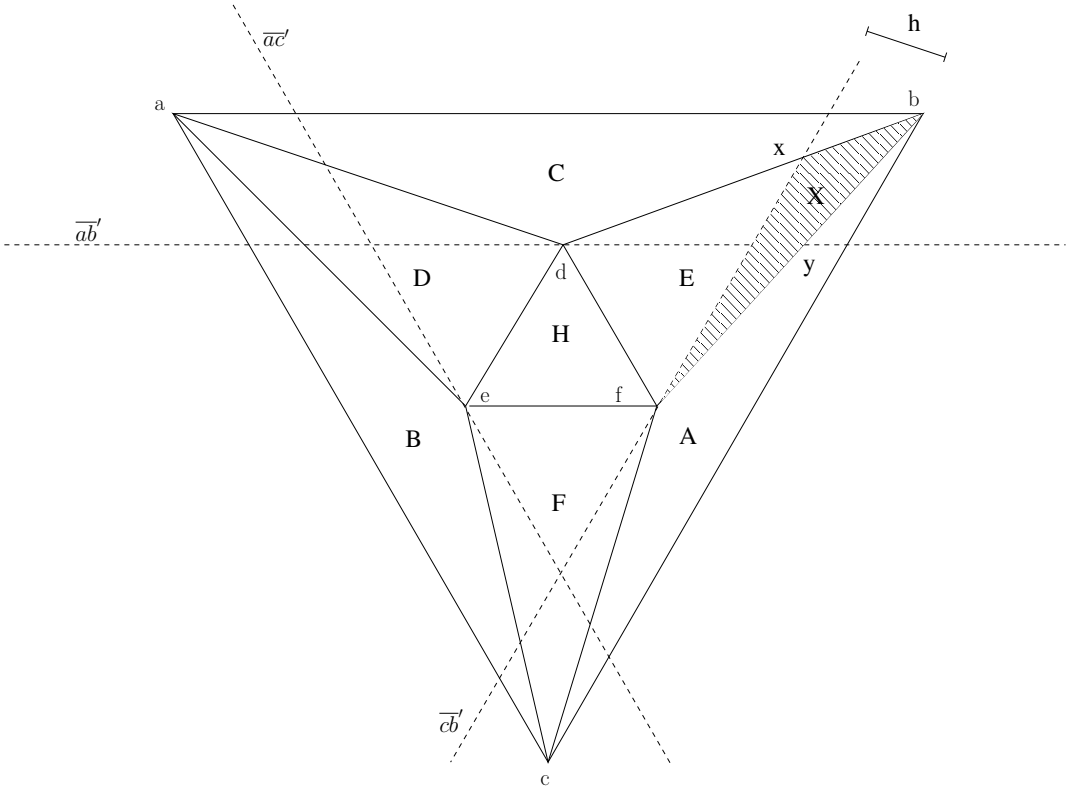


Figure 4.3: Partition of face E

of a,b and c could not be closed. So the piece of $\overline{cb}'$ which is bounded by $\overline{ab}$ and $\overline{ca}$ is shorter than $\overline{cb}$ itself. But now $\overline{xf}$ is a part of the piece of $\overline{cb}'$ bounded by $\overline{ab}$ and $\overline{ca}$. Hence, $\overline{xf}$ is shorter than $\overline{cb}$. □

We have proven $1 = a(A) > a(X)$. The same can be done analogously for the triangle Y which consists of the three vertices b,d and y where y is the intersection of $\overline{ab}'$ with $\overline{fb}$. With the same reasoning as before we obtain $1 > a(Y)$. Then consider the triangle T_E which is defined by the part of E that does not belong to X or Y. Its area is $a(T_E) = a(E) - a(X) - a(Y) + a(X \cap Y) \geq a(E) - a(X) - a(Y) = 3 - a(X) - a(Y) > 3 - 1 - 1 = 1$. The same holds for the triangles T_F and T_D which are defined in the same way as T_E with the other two edges of H.

But now $\overline{ab}', \overline{bc}', \overline{ca}'$ from a triangle which circumscribes the triangle H. For this configuration of a triangle circumscribing another triangle Debrunner ([Deb56]) stated the following theorem:

Lemma 4.2.2 (Debrunner) *Let ABC be a triangle circumscribed by another triangle $A'B'C'$, i.e. ABC has 3 triangles $ABC', AB'C$ and $A'BC$ on its sides. Then:*

$$a(ABC) \geq min(a(ABC'), a(AB'C), a(A'BC))$$

We conclude that the area of H is at least the area of either T_D or of T_E or of T_F, in other words $a(H) \geq min(a(T_D), a(T_E), a(T_F))$. Earlier we saw that $a(T_D), a(T_E), a(T_F) > 1$ and therefore we have $a(H) > 1$. But this is a contradiction to our requirement $a(H) = 1$. So the given areas can not be realized for the graph G with its given combinatorial embedding which finishes this proof. □

Note that Ringel already showed what we just proved in this section for a similar graph, see [Rin90] . However, his proof is less intuitive.

To our best knowledge a complexity result does not exist for Problem 4.2.1. Thomassen showed that it is polynomially solvable if the graph is cubic [Tho92]. He could not bring up a similar result for more general graphs. So, this is an interesting research direction for future work. However, even without a proof of its NP-hardness the fact alone that there are negative instances means that also this relaxation is not enough to tackle the problem since we might encounter an input which is a negative instance and in that case a result can not be computed.

4.2.3 Removing vertices

Instead of only allowing the movement of vertices we could allow to remove vertices in subdivision. This makes sense above all when combining this relaxation with circular arcs. For a given circular-arc configuration which has been produced somehow (for example with a heuristic) and which does not attain all prescribed face areas because neighbored circular arcs that must be bent further are already bent to a maximum, i.e. have the same tangent with another circular arc at their common vertex v, we look for a new technique to balance the remaining target area changes. In that case the ability to remove v from the subdivision and replace the two existing circular arcs with one circular arc can supply a larger edge bending capacity in terms of area transported over the circular arc and that way help to reduce the area-error of the involved faces.

For this relaxation only vertices of degree must be considered. Vertices of higher degrees can not be removed as easily because they can not be replaced by one new circular arc connecting its two neighbors. But obviously not all vertices of degree two are suited equally well to be removed. Whether or not it is desirable to remove a vertex depends above all on the following criteria:

1. the angle enclosed by the two adjacent straight edges in the initial subdivision
2. the length, height and orientation of the two neighboring circular arcs
3. the remaining desired area changes of the adjacent regions

A vertex which encloses an angle of nearly 180° with its adjacent edges is a good candidate to be removed since no big amount of topological information will be lost as opposed to a vertex which represents a spike in the subdivision for instance. Moreover in an existing bending configuration the circular arcs give a hint where it makes sense to remove vertices of degree two. If two adjacent countries have a big difference in their area demands, i.e. one wants to grow considerably and the other one wants to shrink considerably, then it makes sense not to be to strict about the vertices on the border between the two countries even if a loss of information might occur.

Figure 4.4 shows two typical situations where it can be advantageous to remove a vertex. In the left image if the country below still wants to increase its size then an appropriate action could be to remove the vertex and gain more area with the new circular arc (dotted lines). The right graphic shows the inverse case.

Probably the most promising relaxation is to allow movement for vertices of degree three and higher and to allow removal of vertices of degree two under certain requirements related to the three criteria discussed above. This makes sense in the context of a combination with circular arcs since we then have full flexibility to merge circular arcs by removing vertices of degree two and simultaneously to shift countries by moving vertices of degree three or higher. The next section treats combined relaxations deepening this idea and involve area relaxations as well.

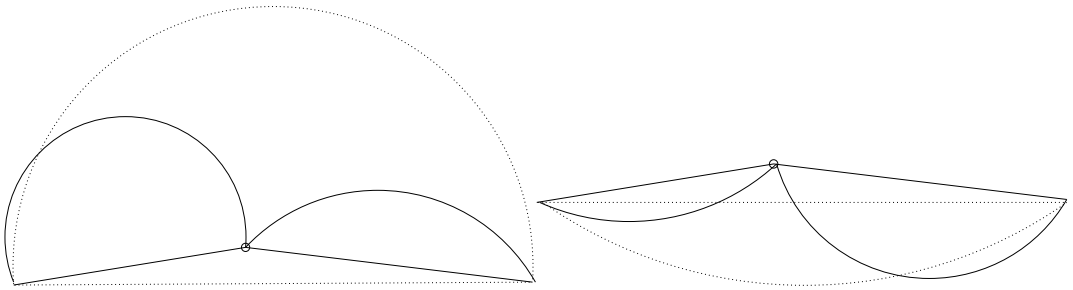


Figure 4.4: The marked vertices can be removed without a loss of essential information

4.3 A Combined Relaxation Approach

From the number of complexity results for Circular-arc Cartogram problems (see Section 3.3) we concluded that constraints in the problem setting have to be dropped or at least relaxed in order to allow for efficient algorithmic methods which solve the problem. But the more the constraints are relaxed the heavier the quality of resulting maps potentially degrades due to the new degrees of freedom. We tried to build up a series of relaxations with the goal to make the problem a little bit easier to solve with each relaxation and at the same time still guarantee a certain level of quality.

So far a first relaxation approach was considered where the strict area-constraints are dropped, i.e. we allow an absolute area-error between the actual weight and the required weight for each region. We saw that the problem remains NP-hard even under this condition (Section 4.1.2). This also means that the optimization version of the problem, where for a map the circular-arc configuration with the smallest area-error has to be found, is NP-hard.

We then showed in Section 4.2 how vertex positions can be relaxed: We do not anymore stick with all the original vertex positions but we allow the vertices to move. The idea behind this is that in general vertex movements can correct cartographic errors more significantly than the action of flipping edges because the circular arcs somehow block each other in any subdivision whereas vertex movement is less restricted in most cases. We gave an example of a graph and weights on the graph's faces where the weights are not feasible as areas on that graph when only allowing vertex movements.

Now, let us look at a method based on a combination of area and vertex constraint relaxations that uses circular arcs as well. The method is divided in two phases:

1. Bend edges in the subdivision according to some heuristic or manually with the goal to balance between the weights of the countries, i.e. reduce the sum of all area-errors in the subdivision.
2. Based on the bending configuration proposed in phase 1 merge circular arcs and move vertices of degree three or higher in order to establish a better balance.

These two phases can be iterated over, for instance until reaching a certain quality measured by the overall area-error. The heuristic discussed in the next Chapter 5 has been designed according to this method and parts of it have been implemented and evaluated with a case study.

5. Heuristic Resolution

In this chapter we present a heuristic for the Circular-arc Cartogram problem based on network flows. The algorithm has been implemented and an evaluation has been conducted with real data. In Section 5.1 of this chapter we describe the heuristic. Subsection 5.1.1 and 5.1.2 contain the geometric fundamentals of the modeling of the Circular-arc Cartogram problem with a flow network the precise construction of which is presented in Subsection 5.1.3. In Section 5.2 we discuss some visual output generated with our implementation.

5.1 A Flow-network Based Modeling of the CAC Problem

An input instance of the Circular-arc Cartogram problem consists of a polygonal subdivision $S = (V_S, E_S)$, identified by its nodes V_S and its edges E_S, plus a weight vector t whose components t_i are the quantities that are to be represented proportionally by the areas of the regions of S. The dual graph of S is $G = (V, E)$. Members of E_S will be called primal edges and members of E will be called dual edges in the following. In G every face f_i of S is represented by a vertex v_i and vertices in G are adjacent if and only if they share a common border in S. We require G to be a directed graph which means that for any two vertices that are adjacent we have two edges, one for each direction. Additionally, we augment G with one vertex v_{out} for the outer-face and edges from v_{out} to all v_i and vice versa where f_i borders the outer face. As a bijection exists between the primal faces f_i and the dual vertices v_i we will use these two expressions interchangeably in the remainder of this chapter.

Any edge $e \in E$ of the dual graph represents an adjacency between two countries in the subdivision possibly consisting of several disjoint borders. Every dual edge is hence associated with several primal edges that lie on borders between the two corresponding countries, but every primal edge maps to exactly one dual edge. There are thus no multiple edges in the dual graph between two vertices. We define B to be a function that associates all the primal edges of the corresponding borders in S to its dual edge e:

$$B(e) = \{a \in E_S, \text{ where } a \text{ is a shared edge of the source and target country of } e\}$$

We have the trivial property that $B(e) = B(e')$ where e' is the twin edge of e in G.

Every vertex v_i in G is assigned with two values: the area A_i of face f_i in the initial map and the given weight t_i of that face. From these two values we will now derive a third value

A_{t_i} which is the actual target area of the face, i.e. the area that the face needs to have in a correct cartogram. As weights are most often statistical data it is likely that they are not in the same numeric dimension as the size of the map and therefore they have to be scaled up or down in oder to obtain appropriate sizes for the cartogram. The areas A_{t_i} must thus be proportional to the weights t_i because the cartogram is a proportional representation of the weights. So, we will set the target areas in such a way that the target area of face f_i is exactly the portion of the total initial area of S that f_i should have according to the quotient of f_i's weight and the total weight. Hence, for any vertex v_i of G (or equivalently every face f_i of S) we have a triple (A_i, t_i, A_{t_i}) where the first two are given in the input as faces sizes and weights and where the third can be obtained from the first two with the following formula:

$$A_{t_i} = \sum_{f_j \in S} (A_j) \cdot \frac{t_i}{\sum\limits_{f_j \in S} (t_j)}$$

This transformation is justified by the fact that it brings the convenience of ensuring that the achieved areas are proportional to the target weights and at the same time guaranteeing the total area of the subdivision to remain unchanged in a correct cartogram:

$$\sum_{f_i \in S} (A_i) = \sum_{f_i \in S} (A_{t_i}) \tag{5.1}$$

For a face f_i we denote by $b_i = A_{t_i} - A_i$ the *desired area change* of this face (or *target area change*). The sum of all desired area changes of all faces is then equal to 0 due to the transformation above. Additionally we denote by T_i the achieved area for a region in a specific circular-arc configuration. The *cartographic error* – or *area-error* – of a region is then given by $T_i - A_{t_i}$. There two basic ways to define the cartographic error of a whole cartogram. The first one is to use the maximum area-error of all regions ($\max\limits_{1 \le i \le n} T_i - A_{t_i}$) and the second one is to use the sum of all area-errors ($\sum\limits_{1 \le i \le n} T_i - A_{t_i}$).

If the cartogram is required to realize the exact prescribed weights as face areas and not only the proportional values A_{t-i} that we derived, we can simply scale the resulting cartogram up or down without any deformations until reaching the desired values. For the outer-face vertex we can simply set A_i, t_i and A_{t_i} to 0 since the area of the outer face will not be affected which also follows from the transformation.

The problem in this equivalent setting is in other words a problem of transferring the existing area of the subdivision between the countries in order to balance their sizes according to the target values A_{t_i}. Countries are able to transfer area to or from a neighboring country by bending the edges on the border in either sense. As mentioned earlier the sum of all target area changes b_i is guaranteed to be 0 at the end of this process if a cartographic error of 0 is achieved: $\sum\limits_{f_i \in S} (b_i) = 0$. If a country f_i has $b_i > 0$, i.e. it desires to increase its size, then we call v_i a *supply vertex*. If f_i has $b_i < 0$, i.e. it demands a decrease of its size, then v_i is a *demand vertex*. Countries with $b_i = 0$ are called transfer vertices. Thus v_{out} is always a *transfer vertex*.

As a next step we will assign capacities to the edges in G. We define the capacity $c(e)$ of a dual edge $e = (v_s, v_t) \in E$ to be the ability to transfer area from the source face f_s of e (f_s is the face corresponding to v_s) to its target face f_t (f_t is the face corresponding to v_t) over the primal edges $a_1 \dots a_k \in B(e)$ that lie on the border of size k between source face and target face in the subdivision. The following subsection presents the method used for the computation of the capacities.

5.1.1 Computation of Bending Capacities with the Straight Skeleton

In order to determine the capacity of a dual edge $e = (v_s, v_t) \in E$ we regard the set $B(e)$ as a system of primal edges that must be bent into the target face f_t and where the capacity will be exactly the amount of area which can be transferred from f_s to f_t by the circular arcs of the edges in $B(e)$ respecting some predefined rules. We are therefore interested in circular arcs which can not be bent further without violating constraints such as intersections or modified adjacencies. A first idea is to compute the capacity of each single edge $a \in B(e)$ straightforward by calculating the circular arc emerging from a with maximal height that does not intersect any edge of the target face f_t. But clearly the circular arcs from the different edges in $B(e)$ interfere with each other: If one edge is fully bent into the target face then probably the edge next to it only has a very small maximal height for its circular arc. Likewise also edges from different borders interfere. If an edge is bent considerably far into face f_i then possibly this blocks edges lying on the opposite side of f_i.

Hence, the problem of computing the capacities is not as obvious as it first seemed because the capacities need to be static and should not depend on each other which would require them to change dynamically during the execution of any cartogram algorithm relying on circular arcs. Consequently, we have to think of a better technique to determine capacities. A good alternative is to make use of the straight skeleton concept for polygons. The straight skeleton of a polygon is defined by a process of shifting the polygon's edges inwards simultaneously until intersection events occur. There are two types of events. The first type is the *edge event* which means that an edge is omitted since two of its neighboring edges collide and the second type is the *split event* where an edge collides with a vertex. In the latter case the polygon is split into two at that vertex. The result of this process is a subdivision of the original polygon into disjoint sub-polygons where every sub-polygon corresponds to exactly one edge of the original polygon. The straight skeleton concept was introduced in [AAAG95]. They suggest to construct the straight skeleton by simulating the polygon shrinking process which can be implemented with a runtime lying in $O(n^2 \log n)$ when a priority queue for storing the events.

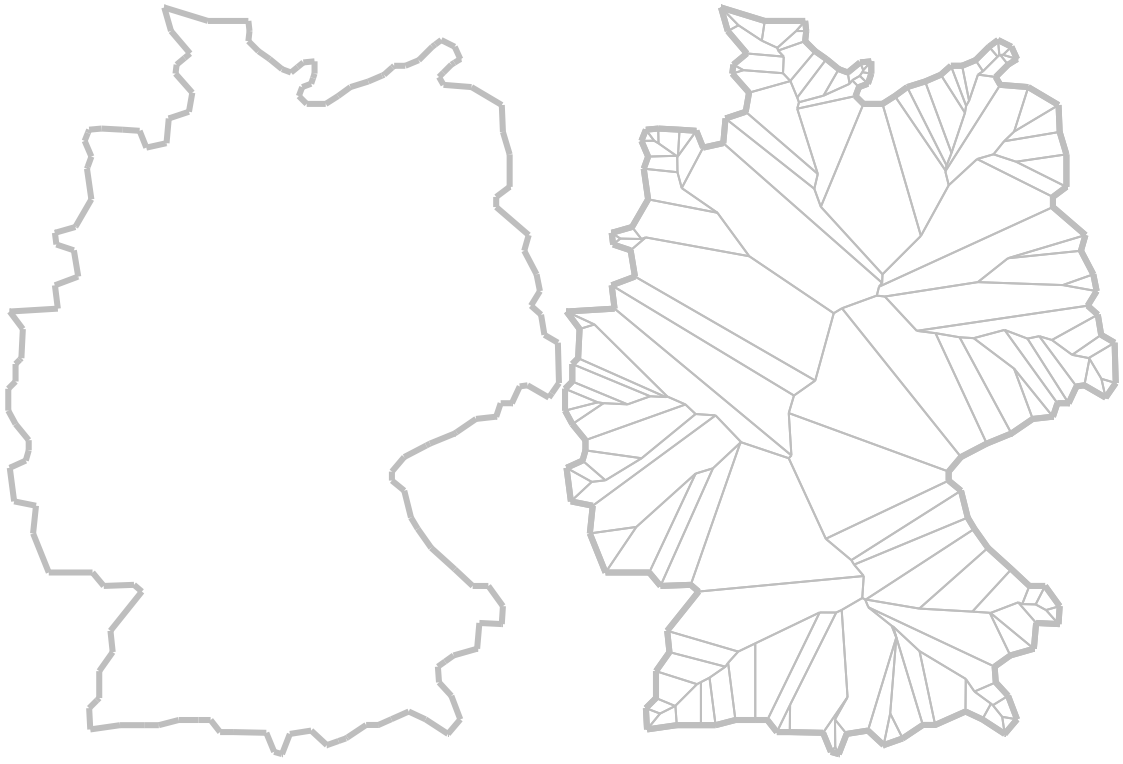
An example of a straight skeleton is depicted in Figure 5.1. The initial polygon in Figure 5.1a is an octilinear schematization of the boundaries of Germany and the unique straight skeleton for this polygon is depicted in Figure 5.1b.

Besides the fact that all components of a straight skeleton are straight edges two additional properties make them very useful for our purpose:

1. Every polygon edge has one dedicated skeleton region that it belongs to
2. All skeleton regions are disjoint

Hence, if we require the circular arcs in our application to stay inside the skeleton region of their corresponding edge then we are sure that intersections of circular arcs can not occur in any bending configuration. Therefore the capacities can be computed independently for each edge. For a primal edge a and a face f which lies on one side of a we call $cap(a, f)$ the maximal area that can be subtracted from f by replacing a with a circular arc that does not exceed f's skeleton region, i.e. the area of the circular segment corresponding to the maximal circular arc. Note that in this context the expressions 'maximal circular arc' or 'maximal area change' always signify that the circular arc is maximal with respect to the straight skeleton region of the corresponding polygon edge.

The capacity of a dual edge $e = (v_s, v_t) \in E$ with target face f_t can then be given as the sum of all maximal area changes of the primal edges lying on its border:



(a) Input polygon (b) Straight skeleton of the input polygon

Figure 5.1: Straight skeleton creation

$$c(e) = \sum_{a \in B(e)} (cap(a, f_t))$$

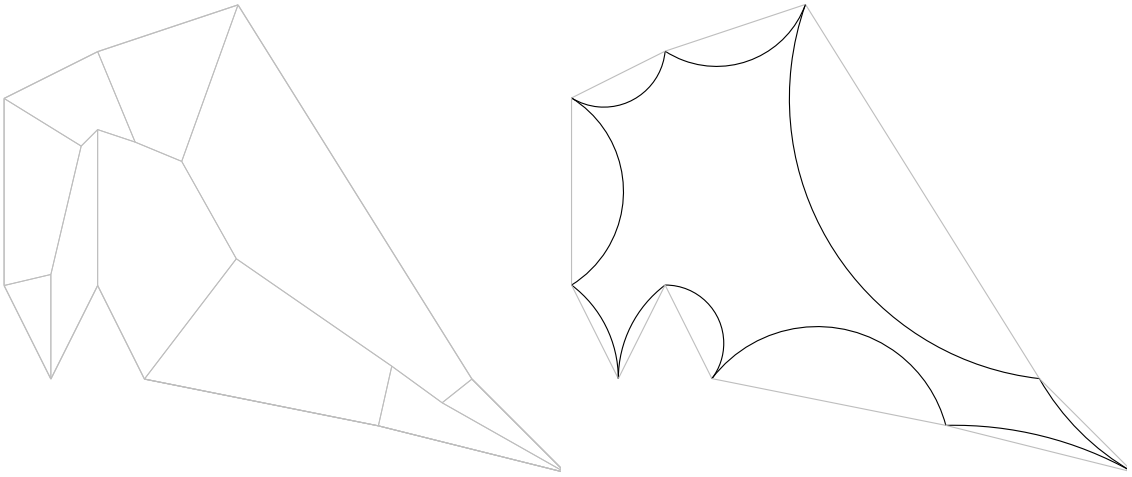
Note that while we had $B(e) = B(e')$ for $e \in E$ and its reverse edge e', the equation $c(e) = c(e')$ does not hold in general since the straight skeletons of the two faces adjacent to an edge may not at all look alike.

Figure 5.2a shows a polygon together with its straight skeleton. A bending configuration with maximally inwards bent circular arcs with respect to the straight skeleton of the polygon is depicted in Figure 5.2b.

With all that we are able to determine all capacities $c(e)$ of dual edges that are directed into a regular face, i.e. have an actual country in the map as their target face. This works fine as all regular faces are simple polygons and we can compute the straight skeleton of a simple polygon. The outer face however is not bounded and therefore is not a polygon. We can fix this problem by using a rectangle as bounding box with very large margins which surrounds the subdivision. Then the former outer face becomes a regular face in which all continents of the initial map are now holes. The straight skeleton is defined for polygons with holes as well – see [AA96] – and this allows us to compute the remaining capacities in the same way as before.

A bending configuration of the polygon in Figure 5.2 with maximally outwards bent circular arcs (i.e. bent into the outer face) is depicted in Figure 5.3. These maximal circular arcs were computed as explained in the previous paragraph.

The methodology of computing capacities with the straight skeleton has been described in this subsection. It remains to describe the exact geometric computation of the height for maximal circular arcs. This is the topic of the next subsection.



(a) Input polygon and its straight skeleton (b) All circular arcs are maximally bent inwards

Figure 5.2: Maximal bending configuration

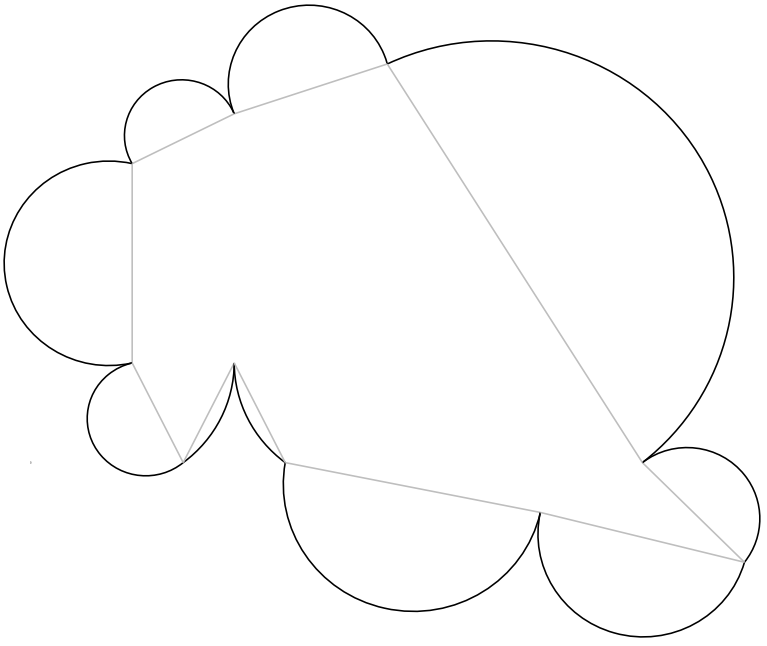


Figure 5.3: Maximal outwards bending configuration

5.1.2 Finding Maximal Circular Arcs

In order to compute the unique maximal circular arc for a primal edge a with respect to the skeleton region associated with a we need to compute all maximal circular arcs with respect to every skeleton edge s of that skeleton region, i.e. the height or radius for which the arc generated from a touches s at one single point. The smallest of these maximal circular arcs determines the bending capacity of a in one direction. In this context small refers to the height of the circular arc. It always holds that if the circular arc C_1 does not intersect a certain skeleton edge s then any circular arc C_2 that is flatter than C_1 (i.e. has a smaller height) does not intersect s as well since C_2 is included in C_1. Because of that we know that the flattest of all maximal circular arcs does not intersect any skeleton edge.

Hence, this problem reduces to the computation of the maximal height of a circular arc being bent into face f_i having a as its chord (the chord is the edge that separates the circular arc from the rest of the circle) such that the circular arc does not intersect a certain skeleton edge s.

We simplify the problem by rotating and translating the two straight segments a and s such that a lies on the x-axis and is centered (i.e. its middle point is the origin of the coordinate-system) with f_i bordering a from above the x-axis. The two endpoints of a then have the form $(0, -d)$ and $(0, +d)$, where $2d$ is the length of a. From now on a and s denote the rotated and translated objects. Figure 5.4 illustrates this simplified setting. The edge a (drawn in fat) has a circular arc with center $(0, b)$ which is maximally bent into the skeleton region of a (dashed edges) that lies within the target face f_i (dotted lines).

In this simplified setting if s is not vertical let s be described by the following straight line equation where m is its slope and n is its y-axis intercept:

$$s : y = mx + n \tag{5.2}$$

If s is vertical it will be described as follows:

$$s : x = n \tag{5.3}$$

Note, that this does not describe the straight segment s only but the whole line passing through it. Later when we compute intersection points we will always have to check if the intersection point is actually contained in the segment s or not.

Then we define a circle C which passes through the two endpoints of a and has the center $(0, b)$:

$$C : x^2 + (y - b)^2 = d^2 + b^2 \tag{5.4}$$

The radius of C is $\sqrt{d^2 + b^2}$. The maximal circular arc which we look for is the part of C which lies above the x-axis.

We distinguish five cases with respect to the positioning of s in order to compute the maximal circular arc for a and s. The non-trivial cases 2,3,4 and 5 are illustrated in Figure 5.5.

case 1) s lies entirely below the x-axis:

This case is trivial as the circular arc will never intersect s and therefore the maximal height of the arc is ∞.

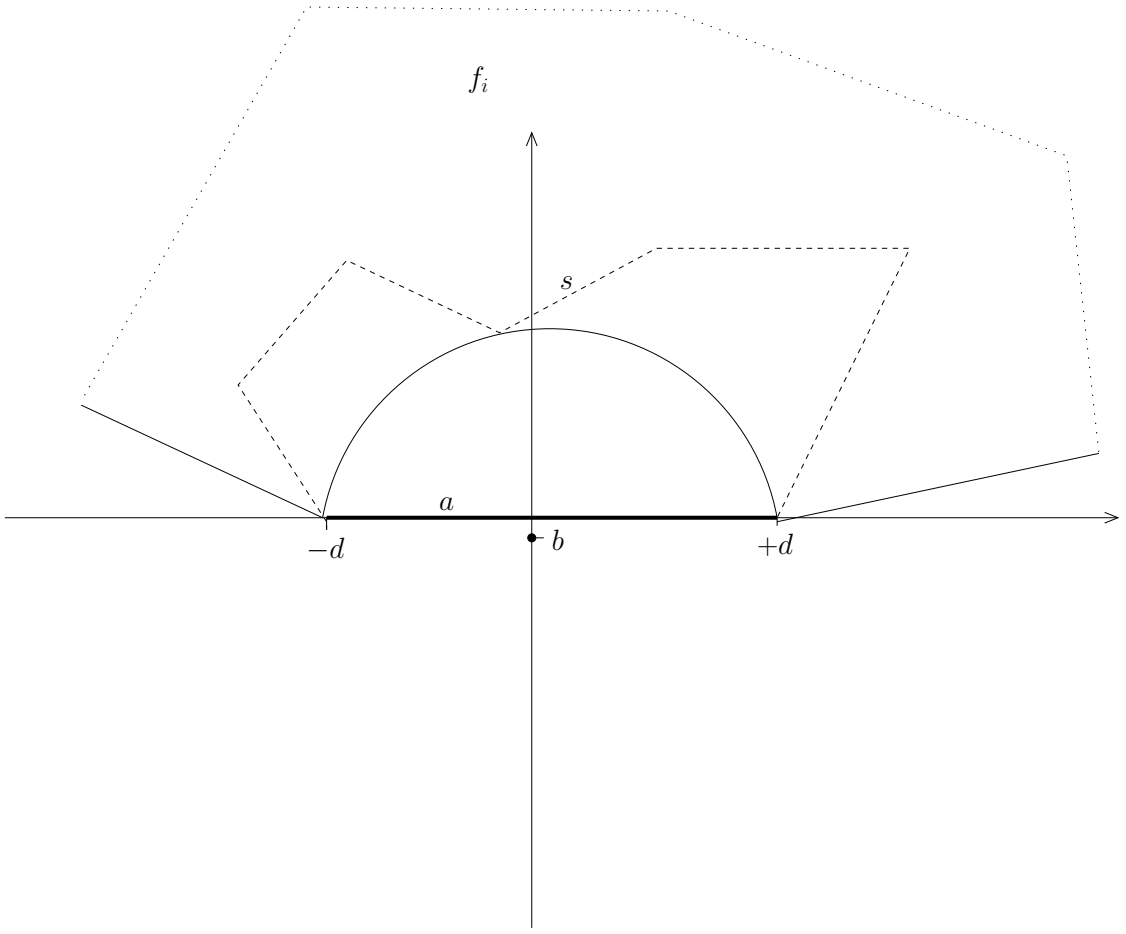


Figure 5.4: Maximal circular arc for a

case 2) s **and** a **share a common vertex:**

The maximal circular arc is then only limited by the angle between a and s. We choose the circular arc in such a way that its tangent at the common vertex has the same slope as s. Then the only intersection point of the circular arc and s is the common vertex of s with a.

case 3) s **is a vertical segment and its** x**-value lies in the interval** $[-d, +d]$**:**

The intersection of the circular arc and s will always occur at the lower vertex (x_{low}, y_{low}) of s. We can therefore simply substitute $x = x_{low}$ and $y = y_{low}$ in Equation 5.4 and resolve for b:

$$b = \frac{x_{low}^2 + y_{low}^2 - d^2}{2y_{low}}$$

By giving a value for b the circular arc is uniquely defined.

case 4) s **is a vertical segment and its** x**-value does not lie in the interval** $[-d, +d]$**:**

In this case the intersection can occur at any point of s. We can write s as shown in Equation 5.3. This substituted into Equation 5.4 and resolved for y gives

$$y_{1,2} = b \pm \sqrt{d^2 + b^2 - n^2}$$

We want this equation to have exactly one solution (i.e. the circular arc does not really intersect but just touch the segement s) which means that the term under the

square root must equal 0 which leads to

$$b_{1,2} = \pm\sqrt{n^2 - d^2}$$

From the fact that s lies entirely to the left or right of a we conclude that the circular arc can be bent further than a half-circle without intersecting s because only then the arc will exceed the x-boundaries of a. Thus b must be positive: $b = \sqrt{n^2 - d^2}$. It remains to detect if this value of b actually leads to an intersection point that is contained in the segment s. This can be done by comparing the coordinates of the intersection point with the endpoints of s. Otherwise the circular arc only intersects the line passing through s. But then finding the circular arc which touches s at exactly one point reduces to case 3 because the intersection will occur at the lower vertex of s.

case 5) none of the cases above applies:

We substitute Equation 5.2 into Equation 5.4 and resolve for x:

$$x_{1,2} = -\frac{mn - bm}{1 + m^2} \pm \sqrt{\frac{b^2m^2 + 2bn + d^2m^2 - n^2 + d^2}{(1 + m^2)^2}}$$

Again, we only want the arc to touch s and therefore the equation above is required to have exactly one solution which leads to

$$b_{1,2} = -\frac{n}{m^2} \pm \sqrt{\frac{n^2}{m^4} - d^2 - \frac{d^2}{m^2} + \frac{n^2}{m^2}}$$

One of the two solutions for b will lead to an intersection point with positive y-value and the other to a negative one. We are interested in the circular arc above the x-axis so we only regard the intersection point with positive y-value. Then it only remains to check whether the intersection really occurs on a point of s or just on the line (as in case 4). If the latter is the case then again the problem reduces to case 3.

5.1.3 Algorithmic Concept

From the two preceding subsections we know how to construct a directed dual graph G for a subdivision S which has capacitated edges where the capacities signify how much area can be transfered over the corresponding border in S from one face to another. These capacities were defined in such a way that they are independent of each other. So, when creating circular arcs somewhere in the subdivision this does not change anything about the existing capacities in the graph. The outer face was included in these considerations, too. In addition to the capacities we derived a value b_i for each vertex signifying how much area the corresponding face demands or supplies.

The goal now is to find a bending configuration which balances the b_i's as good as possible, i.e. the demands of all vertices are met, by sending flow over the edges respecting the capacities. The problem in this setting is equivalent to the so-called transshipment problem on G where units are sent over the edges of a network and the quantity of units flowing on an edge e is limited by its capacity $c(e)$. The overall goal is to find a flow $f : E \mapsto \mathbb{R}^+$ that satisfies all b_i's and minimizes the total costs caused by flow over an edge measured in cost per flow unit $p(e)$. A feasible flow f must thus meet these two requirements:

$$f(e) \leq c(e) \quad \forall e \in E \qquad \text{(capacity constraint)}$$

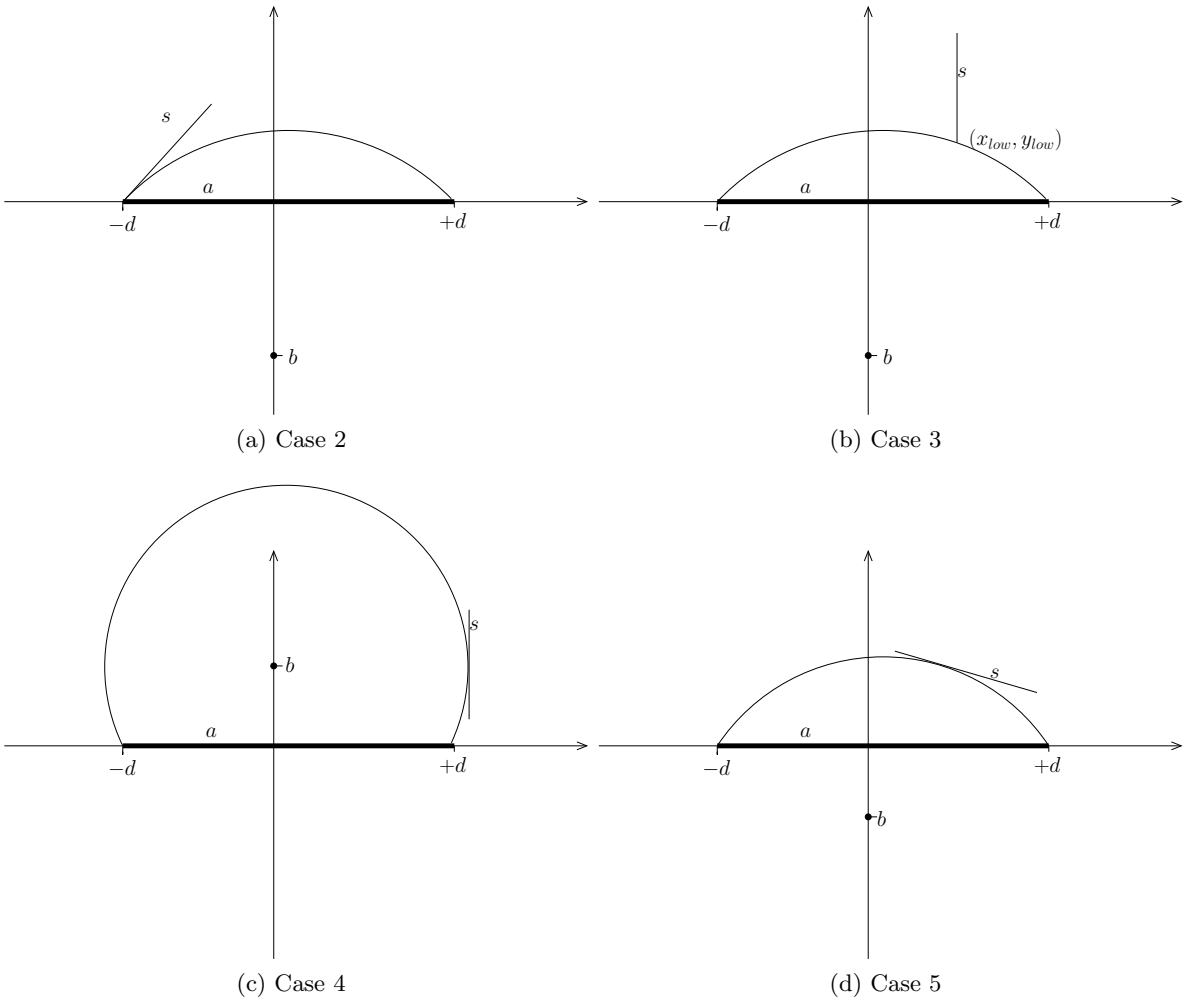


Figure 5.5: Non-trivial cases of the position of s with respect to a in the simplified setting

$$\sum_{e=(v_j,v_i)} f(e) - \sum_{e=(v_i,v_j)} f(e) = b_i \quad \forall v_i \in V \qquad \text{(demand/supply constraint)}$$

In our context as costs we can set $p(e) = 0$ for all edges $e \in E$ since we simply want to know if it is possible to balance the demands and supplies of the different vertices with a feasible flow and it is not important which edges are used for transferring area. So we do not penalize the use of any edge. Let us call this problem $\mathcal{T}$.

The transshipment problem is closely related to the maximum-flow problem which is well explored and for which numerous algorithms exist. In fact the transshipment problem when only asking if a feasible flow exists can be solved with a maximum-flow algorithm. Hence, we will remodel the problem and solve it with a max-flow algorithm. A source and target vertex v_S and v_T must be added to G and connected to the existing vertices as follows:

- $(v_S, v_i) \in E$ with $c(v_S, v_i) = b_i$, if $b_i > 0$
- $(v_i, v_T) \in E$ with $c(v_i, v_T) = -b_i$, if $b_i < 0$

We denote this extended graph by $G' = (V', E')$. In Figure 5.6 an example for G' is depicted. The initial subdivision is drawn in bold black and G' is illustrated in blue. The max-flow problem $\mathcal{F}$ is to find a feasible flow f maximizing

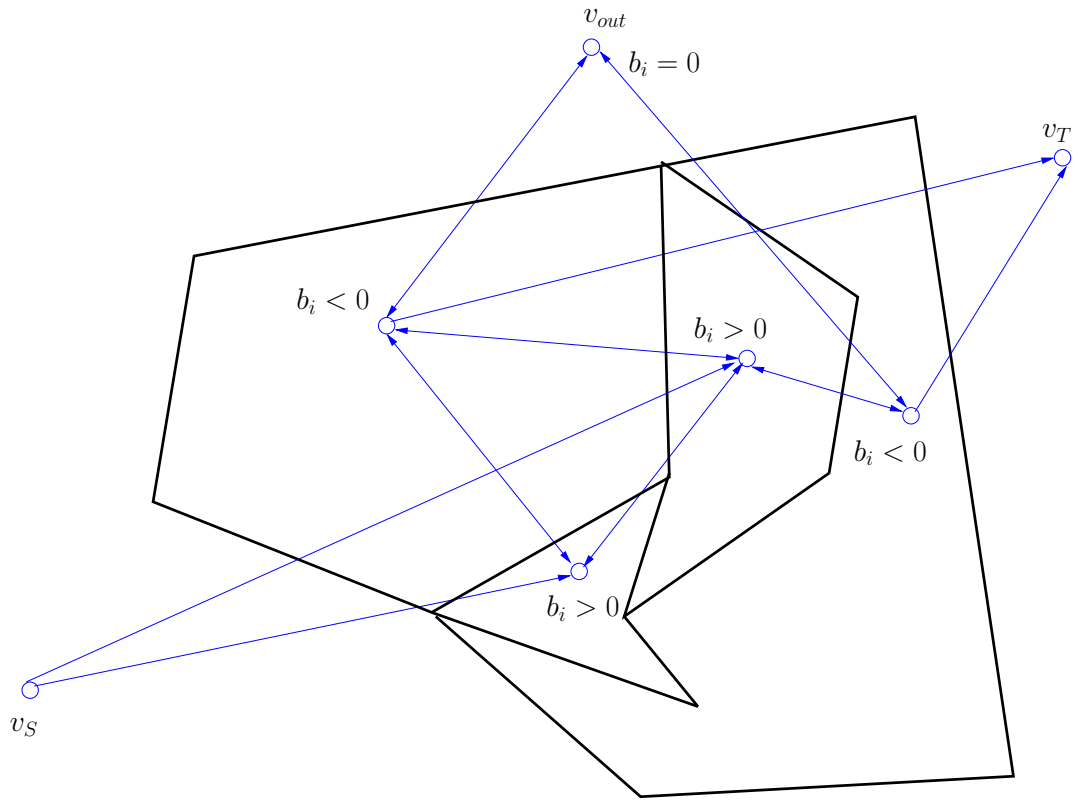


Figure 5.6: Flow network G'

$$\sum_{e=(v_S,v_i)\in E} f(e). \qquad \text{(flow value)}$$

In contrast to $\mathcal{T}$ this problem defines a feasible flow by replacing the demand/supply constraint above with the following flow conservation constraint:

$$\sum_{e=(v_j,v_i)} f(e) - \sum_{e=(v_i,v_j)} f(e) = 0 \quad \forall v_i \in V \setminus \{v_S, v_T\} \qquad \text{(flow conservation)}$$

Obviously there is a strong relationship between $\mathcal{T}$ and $\mathcal{F}$ which manifests itself in the following theorem:

Theorem 5.1.1 *A feasible flow for $\mathcal{T}$ exists if and only if the optimum solution for $\mathcal{F}$ has value $\sum_{b_i>0} b_i$*

Proof We show both implications:

$\Rightarrow$ Let f be a feasible flow for $\mathcal{T}$ on G.
We define the flow f' as follows:

- $f'|_E = f$
- $f'(e) = c(e)\ \forall e \in E' \setminus E$

Thus all edges from the source and to the sink are saturated and all other edges have the same flow as in f. Clearly f' is a feasible flow for $\mathcal{F}$ and its flow value is $\sum_{b_i>0} b_i$. No other flow can be higher since all source edges are saturated.

$\Leftarrow$ Let f be a maximum flow for $\mathcal{F}$ with a flow value of $\sum_{b_i>0}(b_i)$.
In f all source and sink edges must be saturated, otherwise the value could not be attained. The flow f restricted to the edges in E is a feasible flow for $\mathcal{T}$ on G: Capacities are still respected and the demand or supply of each vertex is also met since a surplus or lack of exactly b_i occurs at vertex v_i because of the missing source and sink edges. Therefore $f' = f|_E$ is a feasible flow for $\mathcal{T}$. □

Hence, if a max-flow in G' with a value of $\sum_{b_i>0} b_i$ can be found we know that all target areas can be achieved. Furthermore even if a max-flow value of $\sum_{b_i>0} b_i$ can not be attained, the max-flow still optimizes problem $\mathcal{T}$ since the difference to $\sum_{b_i>0} b_i$ is exactly the total area-error. This fact justifies using a max-flow algorithm heuristically for the resolution of the Circular-arc Cartogram problem.

There is one last issue related to the structure of the given flow network G'. In G' we have positive capacities for both edges of one edge pair. However, most popular max-flow algorithms require that for an edge e with $c(e) > 0$ the reverse edge e' must have $c(e') = 0$. In our application we need to have $c(e') > 0$. The method of Boykov-Kolmogorov ([BK04]) can deal with this and therefore our heuristic relies on their max-flow algorithm.

In the last three subsections we described all important components of our heuristic which consists of two major phases. First the straight skeletons of all faces in the subdivision and the resulting capacities of the dual graph are computed. Then the algorithm of Boykov-Kolmogorov is run on the extended dual graph and the resulting flow is applied to the subdivision as a valid bending configuration. The runtime of this method lies therefore in $O(ln^2 \log n + ml^2|C|)$ where n is the number of vertices in the subdivision, m is the number of edges, l is the number of faces and $|C|$ is the cardinality of a minimum cut in the dual graph of the subdivision. It depends thus on the runtime of the algorithm of Boykov-Kolmogorov (second term) which can be very slow theoretically in the worst case but in practice performs better than other known max-flow algorithms.

5.2 Case Study

The heuristic described in the previous Section 5.1 has been implemented in `C++`. Geometric data-structures, geometric operations and graph algorithms from the geometry library `CGAL` and the `Boost Graph Library` were employed in the implementation as well. In this section we showcase some visual output produced by this implementation and we evaluate the underlying heuristic by assessing the quality of the obtained cartograms.

The schematized input maps used for producing these cartograms were provided by Wouter Meulemans generated with his algorithm for area-preserving subdivision schematization [MvRS10].

Figure 5.7a shows a circular-arc cartogram for the 2010 gross domestic product (GDP) in Europe[1]. The original octilinear schematization is drawn in gray and the resulting circular-arc cartogram in black. This first cartogram already gives some hints about what are the strengths and weaknesses of our method. The regions in the output are all recognizable.

[1]GDP data of 2010 from `http://www.imf.org/external/pubs/ft/weo/2011/01/weodata/index.aspx`

We can easily identify the different countries and shapes are not distorted too bad. Even the aspect ratio of most regions does not change significantly. The length of all borders is at least as big as in the input, so adjacencies are even more easily readable.

However, the cartographic error for some regions is extremely high. Table 5.7b maps countries to their quotient of realized area change by desired area change. Noticeably there are a lot of countries for which less than 10% of the target area change could be realized. Especially countries lying in the middle of a cluster of countries that all want to increase their size (respectively decrease their size) have difficulties to do so as there is no neighbor that the can get area from (respectively transfer their area to). Switzerland, Luxembourg and Belgium are such cases in this cartogram. This deficiency of having countries unable to change their area because they are enclosed by neighbors which have the same goal to grow or to shrink can not be rectified with circular arcs or structural modifications to the existing flow network. The problem lies deeper and is caused by the static requirement that vertex positions are fixed. Thus a strategy needs to be conceived in future to move vertices in order to deal with these closed-in countries. On the other hand countries like Italy, Spain or the Scandinavian countries that have a long border with the sea can realize a considerable change of their size which leads to a good success quotient.

We conclude from this output that the existing implementation does not always provide high-quality cartograms, i.e. with low cartographic errors, if the map is very much clustered in increasing and decreasing zones which are not equipped with a long sea border. In practice this is however often the case. Many maps have zones where some over-proportioned or under-proportioned countries are clustered no matter which statistical data we regard and also very often these zones do not necessarily have a long common border with the sea which makes it difficult and sometimes impossible to change the area of regions in the middle of the cluster at all.

The cartogram in Figure 5.8a illustrates the distribution of Italy's population per region in 2010[2]. Again the success rate in % is depicted in Table 5.8b. From the table it can be seen that this cartogram has by far a better quality than the previous one. The average success rate is 78.045% and most regions are at 100% which means that their size is correct according to the prescribed value. Very few vertex movements could correct the remaining area-errors completely if such a technique was implemented. For instance consider Sardegna which is the leftmost island. It is shrunk to the smallest possible configuration with circular arcs according to our method but it still is over-proportioned with respect to the prescribed quantity. Only a movement or removal of vertices as discussed in Section 4.2 could bring the necessary degree of freedom to shrink Sardegna's size further in order to deal with the remaining area-decrease that has to be done.

The reason why this input leads to a considerably better cartogram than the one before lies in the structure of the initial map and the structure of the provided weight vector. A first structural property which benefits the cartogram generation is the skinny shape of Italy with a high aspect ratio where almost all regions have a common border with the sea. Areas can than easily be exchanged with the sea which leads to a low cartographic error. In addition the weight vector matches the sizes of the regions quite well in the initial map. There are no big deviations between areas of the regions and their population since Italy is quite uniformly populated. The algorithm can realize these moderate area changes per region sufficiently well and the produced cartogram is thus a good one with acceptable cartographic error.

Figure 5.9 shows a cartogram for the population distribution in the Netherlands in 2004[3]. The Netherlands are not as evenly populated as Italy. For example the provinces Noord-

[2]Population data of 2010 from `http://demo.istat.it/pop2010/index1_e.html`

[3]Population data of 2004 from `http://en.wikipedia.org/wiki/Ranked_list_of_Dutch_provinces`

(a) GDP in Europe

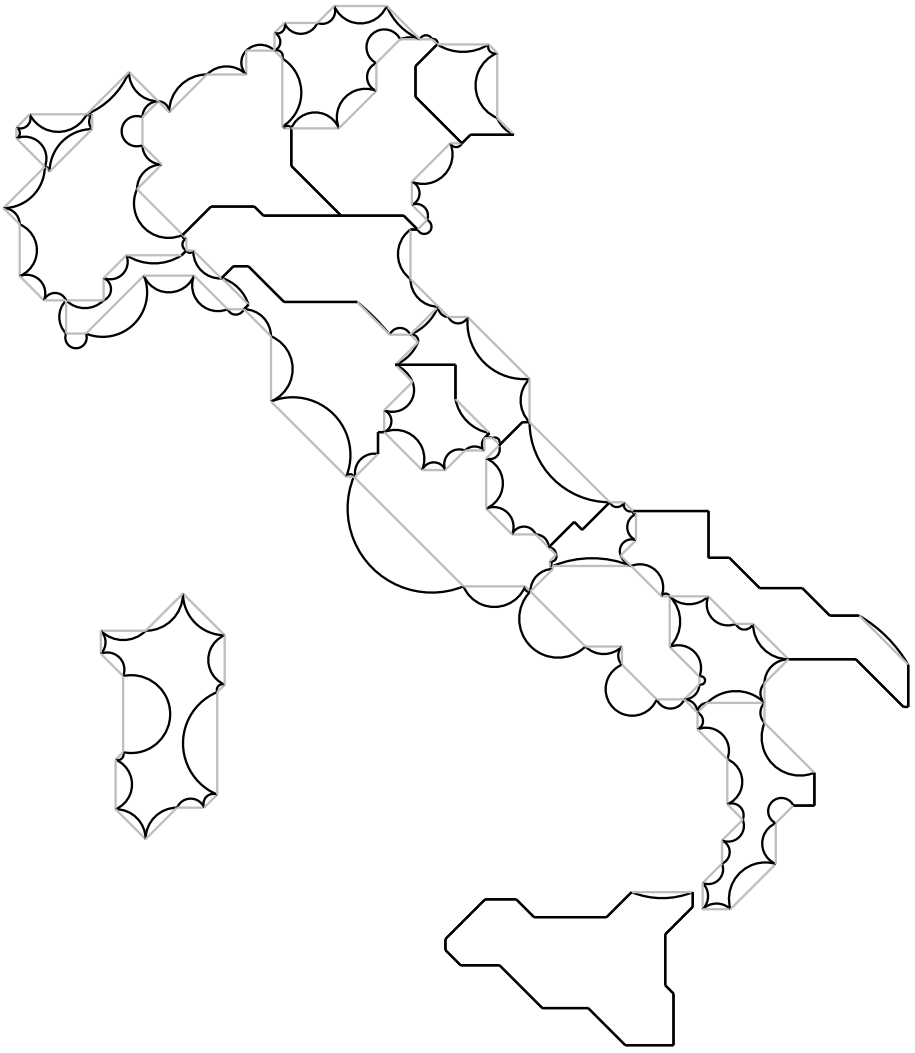
Country	Success in %
Portugal	100%
Iceland	54.6%
Ireland	100%
United Kingdom	31.5%
Spain	100%
Belgium	7.1%
Luxembourg	0%
Netherlands	10.65%
France	29.13%
Italy	73.6%
Denmark	54.4%
Switzerland	0%
Germany	4.5%
Slovenia	100%
Austria	39.5%
Czech Republic	100%
Sweden	44.6%
Croatia	20.2%
Bosnia	6%
Norway	47.4%
Slovakia	34.7%
Albania	38.1%
Poland	19.6%
Hungary	2.9%
Finland	57.83%
Lithuania	6.2%
Serbia	7%
Estonia	32.6%
Macedonia	22.1%
Latvia	21.7%
Greece	100%
Moldavia	0%
Belarus	12.1%
Romania	2.7%
Bulgaria	17.2%
Ukraine	20.3%

(b) Realized area change in %

Figure 5.7: Circular-arc cartogram: GDP in Europe

Brabant as well as Zuid- and Noord-Holland containing all important urban areas make up the largest part of the Dutch population. There is thus a big imbalance between south and north and this fact also expresses itself in the cartogram. The regions of the metropolitan south are shaped like clouds and the northern rural areas look more like snowflakes with all the edges bent inwards. The imbalance in the distribution of the Dutch population is thus indicated by the cartogram, the significance of this imbalance however does not become evident. Zuid- and Noord-Holland still want to grow a lot and both have not even achieved one third of their desired growth. In this case an action of merging two circular arcs would be very effective because both regions have long sea borders with circular arcs blocking each other at their common vertices. If these vertices were dropped the regions could gain a lot more area over the new larger circular arcs. The same holds for the northern territories which still want to shrink considerably.

An advantageous side-effect of this circular-arc cartogram is the fact that adjacencies are much better visible than in the original map. The bigger length of the border when



(a) Population distribution in Italy

Province	Success in %
Valle d'Aosta	78.8%
Piemonte	100%
Sardegna	68.1%
Liguria	100%
Lombardia	17.9%
Toscana	100%
Trentino	62.3%
Emilia-Romagna	100%
Veneto	64.3%
Umbria	56.2%
Friuli	100%
Marche	100%
Lazio	100%
Abruzzo	86.5%
Molise	34.4%
Sicilia	100%
Campania	47.5%
Basilicata	44.9%
Calabira	100%
Puglia	100%

(b) Realized area change in %

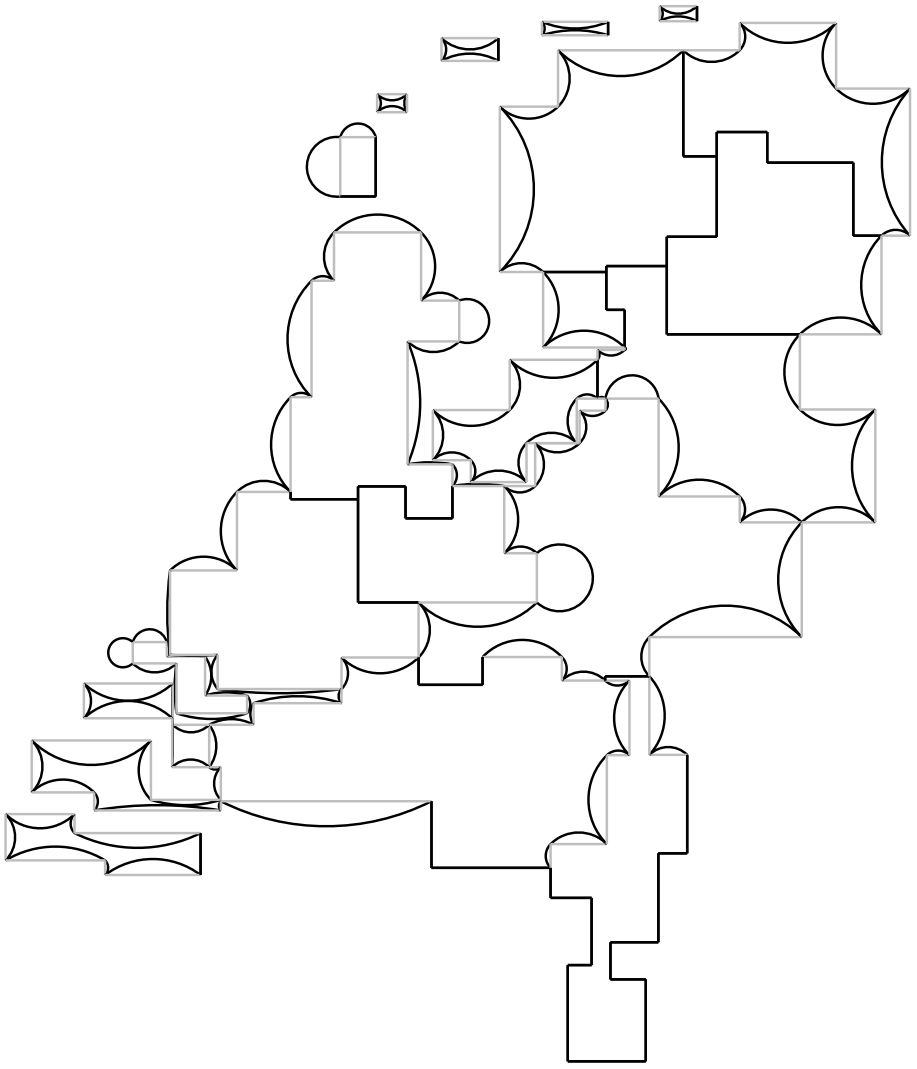
Figure 5.8: Circular-arc cartogram: Population distribution in Italy

replacing straight lines with circular arcs was already discussed before as one advantage of circular-arc cartograms and here we see a practical example of this. Consider the border of the province Zealand which consists of the three bottom left islands in the map. In the original schematized map (gray lines) their boundaries were hard to read because they are a bit nested, i.e. they lie close to each other and all have approximately the same form. The circular arcs reach a better clarity since they bring more contrast into the drawing and that way help to distinguish between the different regions more easily.

The evaluation of this cartogram is thus a mixed one. High cartographic errors of some regions stand in opposition to a better readable overall shape of the cartogram compared to the original map. The size of the southern metropolitan provinces deviates too much from a proportionally correct cartogram so we can not say that this cartogram is one that might be used to illustrate this statistic realistically.

Figure 5.10 shows a cartogram of total agricultural exports in the USA by states in 2010[4]. Note that Alaska and Hawaii have been omitted for ease of illustration. This cartogram confirms the conclusion that we drew from the preceding population cartogram of the Netherlands. The cartogram indicates where the big agricultural US-states lie. From the

[4]Data of 2010 taken from the United States Department of Agriculture `http://www.ers.usda.gov/Data/StateExports/`



(a) Population distribution in the Netherlands

Province	Success in %
Zealand	100%
Zuid-Holland	11.4%
Noord-Holland	32.4%
Friesland	35.9%
Utrecht	49.4%
Flevoland	74%
Noord-Brabant	100%
Limburg	100%
Gelderland	100%
Overijssel	67.8%
Drenthe	11.6%
Groningen	47.8%

(b) Realized area change in %

Figure 5.9: Circular-arc cartogram: Population distribution in the Netherlands

cloud shape of the Midwest-states and California one can deduce that these states export more agricultural products than the rest of the country with respect to their size. However, the actual proportions do not shine through. Most states in the Midwest like Arkansas, Minnesota, Kansas or Oklahoma have only achieved less than one third of their desired area increase, see Figure 5.11. Likewise states on the East Coast and in the mountain range have only realized a very small portion of their desired area decrease. In some cases the success rate is only at 10% or less which degrades the quality of the cartogram significantly. So this example also demonstrates the necessity to implement other relaxations, namely the possibility to move or remove vertices.

One effect that is striking in this cartogram is the dependence of our heuristic on a input map that has been schematized to a considerable extent. If many straight line segments in the input are very short then the circular arcs can not realize a big area change in general. For instance Texas has an advantageously long border with the sea. However, all these edges are pretty short and therefore Texas does not get the flake shape that we would like it to have but instead looks a bit nibbled. Long edges favor good edge bending capacities and therefore lead to better cartograms in terms of a smaller cartographic error and a more appealing shape (recall the cloud and flake analogy). Obviously the necessity to have long edges is also linked to the vertex removal relaxation discussed previously an

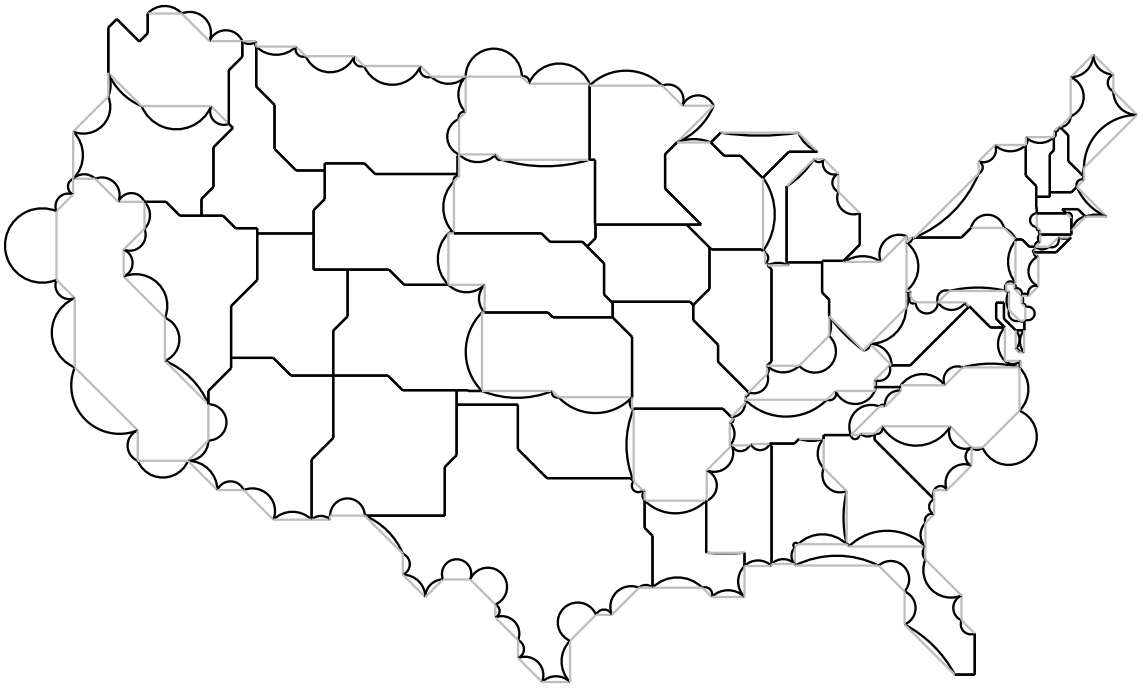


Figure 5.10: Circular-arc cartogram for the exports of agricultural products in 2010 by states

Country	Success in %
California	31.1%
Oregon	37.8%
Washington	100%
Nevada	12.9%
Idaho	0.6%
Arizona	12.7%
Utah	0%
New Mexico	3.4%
Wyoming	5.8%
Montana	15.5%
Colorado	13.5%
North Dakota	72.3%
South Dakota	100%
Nebraska	5.3%
Kansas	20.4%
Oklahoma	31.8%
Texas	38.9%
Iowa	0%
Minnesota	13.3%
Arkansas	33%
Louisiana	48.7%
Missouri	3.1%
Wisconsin	17.3%
Illinois	1.2%
Mississippi	100%
Michigan	100%
Indiana	9.6%
Alabama	26.2%
Kentucky	100%
Tennessee	100%
Ohio	22.4%
Georgia	100%
West Virgina	28.5%
South Carolina	85.6%
Florida	100%
Pennsylvania	100%
North Carolina	96.5%
Virginia	57.3%
District of Columbia	71.2%
Maryland	100%
Delaware	51.6%
New Jersey	100%
New York	68.7%
Vermont	11.2%
Connecticut	100%
New Hampshire	3.8%
Rhode Island	36.9%
Massachusetts	21.9%
Maine	40.7%

Figure 5.11: Realized area change in %

implementation of which would allow to generate long edges where needed during execution of the heuristic.

To finish the evaluation of our heuristic let us have a look at the running time of the algorithm for the generation of the four circular-arc cartograms above. We split the running time up into the duration of the setup of the flow network which includes computing the capacities with straight skeletons and the duration of the max-flow algorithm including the application of the found flow as bending configuration. The resulting running times are listed in Figure 5.12. We conclude that the method is fast and the running times depend directly on the number of countries in the map. This holds for both the generation of the

flow network as well as for the execution of the maximum flow and its application.

Cartogram	Setup of the flow network in *sec*	Max-flow algorithm in *sec*
Europe, GDP	2,276180	1,948683
Italy, Population	1,744108	1,730795
Netherlands, Population	1,686170	1,810870
USA, Agrar exports	3,535535	2,156636

Figure 5.12: Running times of the heuristic for the four examples in this section

6. Conclusion

We presented a new approach to generate cartograms by replacing straight line segments of a subdivision with circular arcs. Several constraints that could be imposed on such a cartogram were proposed. The preservation of adjacencies was decided to be a hard constraint, i.e. they should in no case be modified by a cartogram algorithm. Area constraints and vertex-related constraints are subject to relaxations.

In Chapter 3.3 NP-hardness results were presented for four different variants of the circular-arc cartogram problem and took this as a justification to relax some constraints. At first we tried how the area-constraint could be relaxed, i.e. admitting a bounded area-error per region. The problem with a fixed additive area-error turned out to remain NP-hard as well, so that possible relaxations of the vertex positions were then discussed.

We then proposed a heuristic based on a max-flow algorithm. A maximum flow on the augmented dual graph of the map where flow signifies the transfer of area between faces of the map is susceptible to provide a good approximation to a near-optimal bending configuration minimizing the total area error. No guarantees on the quality can be given though. We produced some results with an implementation of this heuristic in `C++` which point out where the advantages and disadvantages of this heuristic lie. The produced cartograms have a very good quality in terms of their visual appeal. Boundaries that are shaped a bit like clouds and the inverse of clouds or flakes are pleasant to look at and the basic information of the map can be read pretty well from the cartograms (e.g. adjacencies or which country corresponds to which cartogram region). Moreover when imposing the normal bending condition the cartogram benefits from the fact that all regions with a positive target area change are inflated along their entire boundary and regions with a negative target area change are deflated as a whole. However, for certain inputs the cartographic error can be very high for some countries. Especially maps where a lot of countries are clustered that all want to grow or shrink cause this problem. If additionally the provided weights differ too much from the actual proportions of the countries in the map than our heuristic reaches its functional limits and big cartographic errors can be the result.

Numerous further problems emerge from this thesis. On the theoretic side it remains to study possible approximation algorithms for the problem at hand with a guaranteed quality. Perhaps an approximation with a multiplicative factor can be found when the target area changes per region are suitably bounded. Another promising resolution technique is the relaxation of vertex-related constraints. We briefly discussed what this could look like

in Section 4.2 and we saw that the question whether the problem of satisfying all weights in a subdivision with vertex-movements is NP-hard remains open.

Nevertheless a specific strategy was not given and also an incorporation into the implementation has not been conducted. An optimization of the software could also be to use minimum cost flow algorithms and then prioritize sending flow over long edges instead of short edges as then the resulting circular arcs will be flatter which leads to a better readability.

Abstracting from circular arcs could also bring many new possibilities. If we do not anymore limit ourselves to replacing straight segments with circular arcs but with more general kinds of curves (e.g. cubic splines) we gain a new level of flexibility in modifying the initial map. If still requiring a smooth shape of the boundaries as it is automatically the case with circular arcs and cubic splines respectively than we could hope for visually appealing results where the ability to reduce the cartographic error is a lot higher.

A detailed comparison of our method with other cartogram generation methods could be conducted as well as an evaluation by users in order to assess its acceptance in practice.

Bibliography

[AA96] Oswin Aichholzer and Franz Aurenhammer. Straight skeletons for general polygonal figures in the plane. In *COCOON*, volume 1090 of *Lecture Notes in Computer Science*, pages 117–126. Springer, 1996.

[AAAG95] Oswin Aichholzer, Franz Aurenhammer, David Alberts, and Bernd Gärtner. A novel type of skeleton for polygons. *J. UCS*, 1(12):752–761, 1995.

[AAH+07] Oswin Aichholzer, Franz Aurenhammer, Thomas Hackl, Bert Jüttler, Margot Oberneder, and Zbynek Sír. Computational and structural advantages of circular boundary representation. In *WADS*, volume 4619 of *Lecture Notes in Computer Science*, pages 374–385. Springer, 2007.

[ABF+11] Jawaheroul Alam, Therese Biedl, Stefan Felsner, Andreas Gerasch, Michael Kaufmann, and Stephen G. Kobourov. Linear-time algorithms for proportional contact graph representations. *22nd Symposium on Algorithms and Computation (ISAAC)*, 2011. To appear.

[BETT99] Giuseppe Di Battista, Peter Eades, Roberto Tamassia, and Ioannis G. Tollis. *Graph Drawing: Algorithms for the Visualization of Graphs*. Prentice-Hall, 1999.

[BG05] Therese C. Biedl and Burkay Genç. Complexity of octagonal and rectangular cartograms. In *In Proceedings of the 17th Canadian Conference on Computational Geometry*, CCCG, pages 117–120, 2005.

[BH98] Vasco Brattka and Peter Hertling. Feasible real random access machines. *J. Complexity*, 14(4):490–526, 1998.

[BK04] Yuri Boykov and Vladimir Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(9):1124–1137, 2004.

[BL07] Xiang Bai and Longin Jan Latecki. Discrete skeleton evolution. In *EMMCVPR*, volume 4679 of *Lecture Notes in Computer Science*, pages 362–374. Springer, 2007.

[BOO] The boost graph library (BGL). http://www.boost.org.

[BV09] Therese C. Biedl and Lesvia Elena Ruiz Velázquez. Drawing planar 3-trees with given face-areas. In *Graph Drawing*, volume 5849 of *Lecture Notes in Computer Science*, pages 316–322. Springer, 2009.

[BV11] Therese C. Biedl and Lesvia Elena Ruiz Velázquez. Orthogonal cartograms with few corners per face. In *WADS*, volume 6844 of *Lecture Notes in Computer Science*, pages 98–109. Springer, 2011.

[CGA] CGAL, Computational Geometry Algorithms Library. http://www.cgal.org.

[CHvKS10] Sergio Cabello, Herman J. Haverkort, Marc J. van Kreveld, and Bettina Speckmann. Algorithmic aspects of proportional symbol maps. *Algorithmica*, 58(3):543–565, 2010.

[CLRS09] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms (3. ed.)*. MIT Press, 2009.

[dBMS09] Mark de Berg, Elena Mumford, and Bettina Speckmann. On rectilinear duals for vertex-weighted plane graphs. *Discrete Mathematics*, 309(7):1794–1812, 2009.

[dBvKS95] M. de Berg, M. van Kreveld, and S. Schirra. A new approach to subdivision simplification. In *Proc. ACSM/ASPRS Annual Convention*, pages 79–88, 1995.

[DCN85] James A. Dougenik, Nicholas R. Chrisman, and Duane R. Niemeyer. An algorithm to construct continuous area cartograms. *The Professional Geographer*, 37(1):75–81, 1985.

[Deb56] H. Debrunner. Problem 260. *Elem. Math.*, 11:20, 1956.

[dFdMR94] Hubert de Fraysseix, Patrice Ossona de Mendez, and Pierre Rosenstiehl. On triangle contact graphs. *Combinatorics, Probability & Computing*, 3:233–246, 1994.

[Dor96] Daniel Dorling. *Area Cartograms: Their Use and Creation*. Environmental Publications, 1996.

[EMSV09] David Eppstein, Elena Mumford, Bettina Speckmann, and Kevin Verbeek. Area-universal rectangular layouts. In *Symposium on Computational Geometry*, pages 267–276. ACM, 2009.

[Fár48] István Fáry. On straight line representation of planar graphs. *Acta Univ. Szeged. Sect. Sci. Math.*, 11:229–233, 1948.

[FF56] L. R. Ford and D. R. Fulkerson. Maximal flow through a network. *Canadian Journal of Mathematics*, 8:399–404, 1956.

[GHK10] E. Gansner, Y. Hu, and S. Kobourov. Gmap: Drawing graphs as maps. In *Proceedings of IEEE Pacific Visualization Symposium*, pages 201–208, 2010.

[GHKK10] Emden R. Gansner, Yifan Hu, Michael Kaufmann, and Stephen G. Kobourov. Optimal polygonal representation of planar graphs. In *LATIN*, volume 6034 of *Lecture Notes in Computer Science*, pages 417–432. Springer, 2010.

[GJ79] M. R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.

[GN04] 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.

[GT86] Andrew V. Goldberg and Robert Endre Tarjan. A new approach to the maximum flow problem. In *Proceedings of the Eighteenth Annual ACM Symposium on Theory of Computing*, STOC, pages 136–146, 1986.

[HK98] Donald H. House and Christopher J. Kocmoud. Continuous cartogram construction. In *Proceedings of the conference on Visualization '98*, VIS '98, pages 197–204, 1998.

[HKPS04] Roland Heilmann, Daniel A. Keim, Christian Panse, and Mike Sips. Recmap: Rectangular map approximations. In *INFOVIS*, 10th IEEE Symposium on Information Visualization (InfoVis2004), pages 33–40. IEEE Computer Society, 2004.

[HS04] J.-H. Haunert and M. Sester. Using the straight skeleton for generalisation in a multiple representation environment. In *Proceedings of the 7th ICA Workshop on Generalisation and Multiple Representation, August 20–21, 2004, Leicester, UK*, 2004.

[KNP04] Daniel A. Keim, Stephen C. North, and Christian Panse. Cartodraw: A fast algorithm for generating contiguous cartograms. *IEEE Trans. Vis. Comput. Graph.*, 10(1):95–110, 2004.

[Koe36] Paul Koebe. Kontaktprobleme der konformen Abbildung. *Berichte d. math.-phys. Kl. d. Sächs. Akad. d. Wissenschaften zu Leipzig*, 88(2):141–164, 1936.

[MG07] D. Merrick and J. Gudmundsson. Path simplification for metro map layout. In *Proc. 14th Int'l Symposium on Graph Drawing (GD'06)*, volume 4372 of *Lecture Notes in Computer Science*, pages 258–269. Springer-Verlag, 2007.

[MvRS10] Wouter Meulemans, André van Renssen, and Bettina Speckmann. Area-preserving subdivision schematization. In *GIScience*, pages 160–174, 2010.

[new] Maps of the 2008 US presidential election results. http://www-personal.umich.edu/∼mejn/election/2008/.

[PS85] Franco P. Preparata and Michael Ian Shamos. *Computational Geometry - An Introduction*. Springer, 1985.

[Rai34] Erwin Raisz. The rectangular statistical cartogram. *Geographical Review*, 24(2):292–296, 1934.

[Rin90] Gerhard Ringel. Equiareal graphs. In R. Bodendiek, editor, *Contemporary Methods in Graph Theory*, page 503–505. 1990.

[RMN09] Md. Saidur Rahman, Kazuyuki Miura, and Takao Nishizeki. Octagonal drawings of plane graphs with prescribed face areas. *Comput. Geom.*, 42(3):214–230, 2009.

[Tho92] Carsten Thomassen. Plane cubic graphs with prescribed face areas. *Combinatorics, Probability & Computing*, 1:371–381, 1992.

[Tob86] Waldo Tobler. Pseudo-cartograms. *The American Cartographer*, 13:43–50, 1986.

[vKS07] Marc J. van Kreveld and Bettina Speckmann. On rectangular cartograms. *Comput. Geom.*, 37(3):175–187, 2007.

[WEW97] Emo Welzl, Herbert Edelsbrunner, and Roman Waupotitsch. A combinatorial approach to cartograms. *Comput. Geom.*, 7:343–360, 1997.