Lines: 225

Circle Quasi-Cartograms: Dorling Cartograms with Edge Connections and Relaxed Overlap Conditions

Anonymous author(s)

Anonymous affiliation(s)

Abstract

Suppose we are given a graph, $G = (V, E)$, such that vertices have positive weights as well as geometric metadata, such as anchoring (x, y)-coordinates or geometric "home" regions. A ***Dorling cartogram*** is a way to visualize G where vertices are drawn as non-overlapping circles such that the circle for each vertex, $v \in V$, is drawn with size proportional to v's weight and placed close to the anchoring location for v. Typically, the edges of G are not drawn in such visualizations, however; hence, adjacency information can be lost. In this paper, we introduce ***circle quasi-cartograms***, which are modifications of Dorling cartograms where we join each pair of adjacent vertices with a line segment and we provide tunable parameters regarding the drawing, including the degree to which circles overlap. Specifically, we experimentally investigate force-directed circle quasi-cartogram drawing techniques for visualizing graphs where vertices have positive weights and geometric metadata under the the following constraints:

1. Each vertex, $v \in V$ is drawn as a circle anchored to stay within or near a given geometric anchor region associated with v, as determined by an anchor force factor, α.
2. We draw each edge as a line segment joining the centers of its circle endvertices.
3. Circle overlaps are determined by a tunable parameter, ρ.

Our goal is to preserve the geometric and topological information of the graph while still accurately representing the statistical data represented in vertex weights. Applications include population visualization, where we aim to accurately represent the size of populations in a given geographic region while also visualizing connections between regions. Our experiments indicate that these techniques allow us to trade off visualizing spatial and topological information at the cost of visualizing statistical information, which can be controlled by adjusting anchor and overlap forces.

2012 ACM Subject Classification Human-centered computing → Graph drawings; Theory of computation → Computational geometry

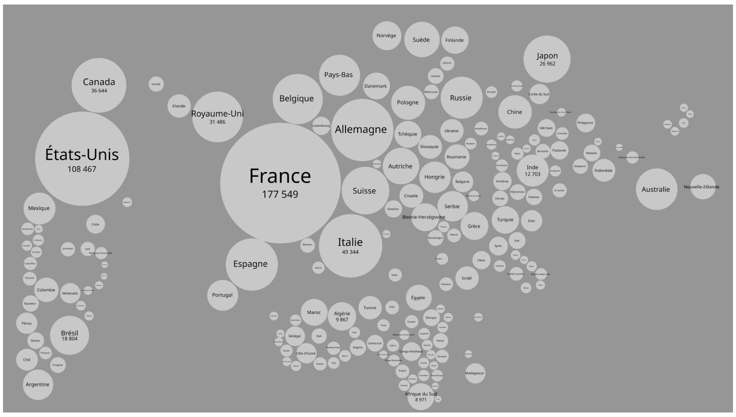
Keywords and phrases circle cartograms, force-directed graph drawing, geo-referenced data, geometric anchors

Category Short (Applied)

1 Introduction

Geometric vertex-weighted graphs, where vertices have associated geometric metadata, such as (x, y)-coordinates, geometric "home" regions, or multi-point associations, are common, but they pose challenging trade-offs when one wishes to visualize them. Cartograms have often been used to visualize such graphs, e.g., in the form of ***value-by-area*** maps, where vertices are drawn as geometric regions and adjacencies are represented by regions that touch; see, e.g., [1, 11, 13–16]. In a circle cartogram, which is also known as a "Dorling cartogram" [7], each vertex is represented as a circle proportional in size to its weight placed close to its geometric anchor so that circles do not overlap. See, e.g., Figure 1.

Typically, as shown in Figure 1, the edges of a graph visualized as a Dorling cartogram are not included. This is unfortunate, of course, since edge connections provide important information, such as topological relationships. Moreover, although the restriction to visualize vertices as non-overlapping circles is aesthetically pleasing, requiring that edges in a circle cartogram be represented only as touching circles severely limits the class of graphs that



■ **Figure 1** A Dorling cartogram of countries weighted by the number of links pointing to a country's article on the French Wikipedia. Image by Wikipedia user Moyogo, licensed under CC By-SA 3.0.

can be visualized. For example, if all the vertex weights are equal, then Dorling cartograms where touching circles represent edges are equivalent to penny graphs, which cannot represent K_4, non-planar graphs, or a number of other interesting graphs; see, e.g., Eppstein [8]. Thus, in practice, if the circles in a circle cartogram do not overlap, and circles are drawn with sizes proportional to their weights, then the visualization necessarily comes at the cost of spatial and topological accuracy, as circles must be shifted without consideration of the original topology. This is usually done via a force-directed algorithm that iteratively repels overlapping circles while keeping them attracted to their initial position and/or edge connections. As a result, Dorling cartograms often fail to preserve the viewer's mental map without additional labeling [16]. Work by Wei, Ding, Xu, Cheng, Zhang, and Wang [17] aims to improve the topology of Dorling cartograms, but it fails to scale to larger graph instances. Our work aims to relax the requirement to completely avoid circle overlaps, while also explicitly drawing a graph's edges as line segments joining the centers of the circles for adjacent vertices, in order to obtain better geographical and topological accuracy.

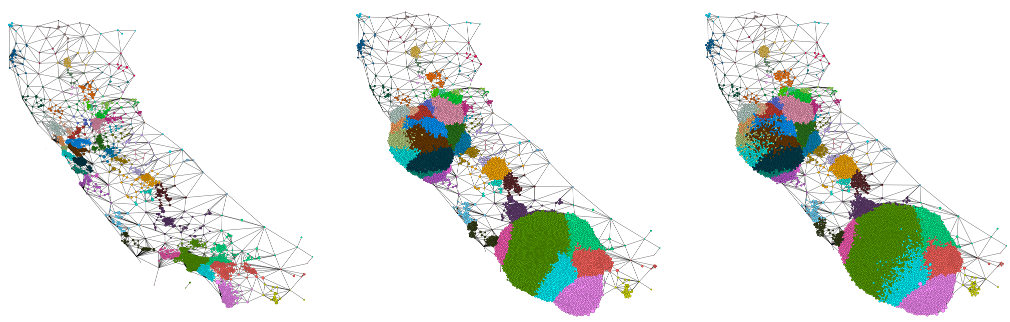
Additional Related Work. Additional related work has considered anchored graph drawing, where the input graph is assumed to have positional information that must be respected in some way [12,18]. Other work has also combined anchor forces with force-directed algorithms to produce nice layouts that respect the initial geography [6].

The standard force-directed algorithm in the literature is the Fruchterman-Reingold algorithm, which treats edges as springs with attractive forces and makes the vertices repel one another iteratively to produce a nice layout. However, this is slow on large graphs, taking $O(n^2)$ time per iteration where n is the number of vertices. Much work has been done to produce faster force-directed algorithms, such as the Fast Multipole Multilevel Method (FM^3) by Hachul and Jünger [10]. The force-directed algorithm on which we build is the Fast Multipole Embedder by Gronemann [9], which uses the same repulsive forces as in Hachul and Jünger's work, but modifies the attractive forces. These algorithms can handle large graphs by approximating the repulsive force calculations between all pairs of nodes using well-separated pair decompositions (WSPD) and multipole expansion [2–4].

Our Results. In this paper, we introduce ***circle quasi-cartograms***, and we provide a flexible force-directed algorithm for drawing circle quasi-cartograms with edge connections explicitly represented and with relaxed overlap conditions, which are controlled by three parameters: a node-scaling factor, ω, an anchor force factor, α, and an overlap force factor, ρ. Since achieving a graph drawing that preserves initial positions while avoiding any node overlaps can be impossible, these three parameters are admittedly potentially in conflict with one another. If, for instance, one sets the node-scaling factor ω arbitrarily close to 0, then we can trivially have no node overlap with the initial layout. If one sets the anchor force factor α to 0, on the other hand, then we no longer have a geographical point of reference. And finally, if one sets the overlap force factor ρ to 0, then we will inevitably have node overlaps with any meaningful set of data. Hence, the desired parameters will be subject to tunable trade-offs, which we explore in this paper.

For instance, by setting the overlap-force parameter ρ sufficiently high, our algorithm produces circle quasi-cartograms that eliminate all circle overlaps. When the input is a geographic map rather than an abstract graph, we first build its dual graph (connecting each pair of adjacent regions) and then apply attractive forces along those dual edges, drawing them explicitly in the quasi-cartogram to preserve regional topology. The anchor-force parameter α governs how strongly each circle is pulled toward its original geographic anchor: reducing α relaxes this pull, allowing nodes more freedom to reposition and overlap less. Conversely, permitting controlled overlap by lowering ρ often yields layouts that better preserve the map's adjacency structure, as illustrated in Figure 2.

In the example shown in Figure 2, the goal is to visualize county populations in California, which is shown in their geographical locations in Figure 2a. We can represent a county in a more granular fashion via its census tracts, which will allow its shape to morph according to the constraints of its geography and topology once we run our algorithm. As census tracts are designed to be relatively uniform in population, we can obtain a much more compact visualization of the population this way. As it stands, we cannot visually see the population size of these color-coded counties due to all the nodes overlapping each other. We show two outputs of our algorithm: Figure 2b that has 1221 overlaps, and Figure 2c that has 9 overlaps. While the latter better represents the true size of these counties due to almost zero node overlap, there is a fair amount of nodes spilling across county borders when this should ideally only happen at the border. The former preserves this topology much better, however, this comes at the cost of node overlaps that produce some minor cartographic error.



(a) Input geographic layout **(b)** $\omega = 100, \alpha = 1, \rho = 1$ **(c)** $\omega = 100, \alpha = 1, \rho = 3$

Figure 2 CA. (a) the dual graph of 8057 census tracts in California, where nodes are color-coded by county and proportional to population. It has 83508 node overlaps. After running our algorithm, (b) shows a drawing with 1221 node overlaps, and (c) shows a drawing with 9 node overlaps.

2 The Force-Directed Algorithm

Fast Multipole Embedder. The base force-directed algorithm we build upon is the Fast Multipole Embedder of Gronemann [9]. As input, we are given a graph layout $G = (V, E)$ where every vertex $v \in V$ has a radius r_v and initial "anchor" xy-position vector $\mathbf{q}_v$. Let two arbitrary vertices, u and v, be distance $d_{uv} = |\mathbf{p}_u - \mathbf{p}_v|$ apart, where $\mathbf{p}_u$ and $\mathbf{p}_v$ are the current xy-position vectors of u and v respectively. Then the magnitudes of the repulsive force $\mathbf{F}_{\text{rep}}(u, v)$ and the attractive force $\mathbf{F}_{\text{attr}}(e)$ if edge $e = (u, v) \in E$ exists will be:

$$|\mathbf{F}_{\text{rep}}(u,v)| = \frac{c_r}{d_{uv}} \text{ and } |\mathbf{F}_{\text{attr}}(e)| = -c_a \cdot \log \frac{d_{uv}}{d_e} \cdot \frac{d_{uv}}{\deg(v)}. \tag{1}$$

Here c_r and c_a are constants, and $d_e = r_u + r_v$ is the ideal edge length for edge $e = (u, v)$, when the two nodes u and v are tangent. Then each iteration of the algorithm will apply these forces until we exceed the minimum number of iterations and the maximum force falls below some threshold, ϵ. However, calculating $\mathbf{F}_{\text{attr}}(e)$ and $\mathbf{F}_{\text{rep}}(u, v)$ exactly will take $O(|E| + |V|^2)$ time per iteration, which becomes infeasible for large graphs.

To overcome this, the repulsive forces are approximated by grouping together far nodes using a well-separated pair decomposition of all pairs of points [3]. This can be found in $O(|V| \log |V|)$ time by constructing a hierarchical partition of the space into quadrants via a quadtree data structure [4]. Then, the calculations themselves are approximated using multipole expansion, which takes the first p terms (depending on the precision needed) of the power series expansion of the forces. For t points, the approximate calculation will now take $O(pt)$ time, rather than the $O(t^2)$ time needed for the exact calculation. The details of these approximations (and the extent to which the error is bounded) can be found in [9].

Anchor and Overlap Forces. We apply two additional forces in each iteration, on top of those supplied by the Fast Multipole Embedder algorithm. First, the anchor force acts like a spring: it takes the displacement vector going from node u's current position $\mathbf{p}_u$ to its initial anchor position $\mathbf{q}_u$, multiplied by the *anchor factor* $\alpha \in [0, 1]$ as seen in Figure 3a:

$$\mathbf{F}_{\text{anchor}}(u) = \alpha(\mathbf{q}_u - \mathbf{p}_u) \tag{2}$$

We allow the anchor factor α to range from $[0, 1]$, where $\alpha = 0$ applies no anchor forces and $\alpha = 1$ means that our anchor force pulls the node equal to its displacement from its original location. There is no need for $\alpha > 1$, otherwise the force will "overshoot" the original anchor position.

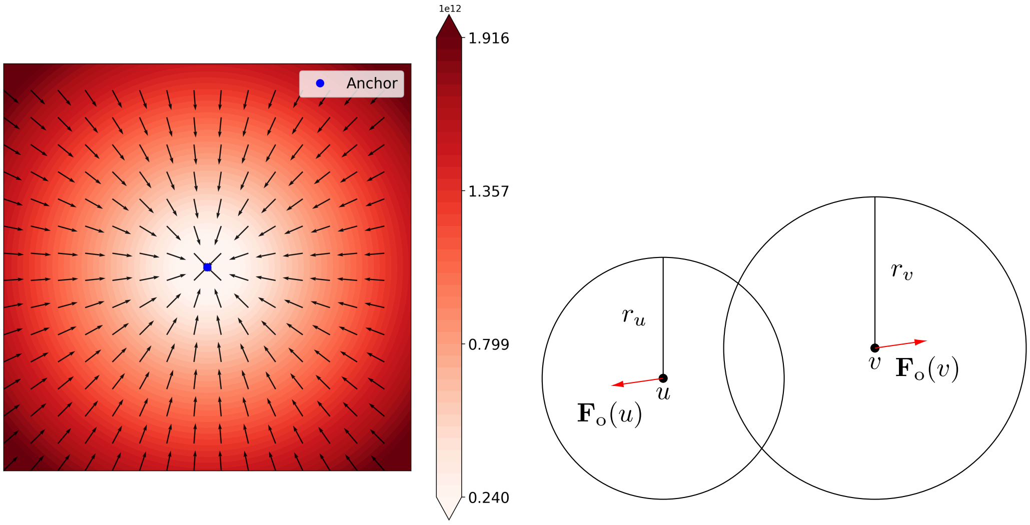
Second, we apply a force if two nodes u, v are overlapping, multiplied by an *overlap factor* $\rho \geq 0$, as seen in Figure 3b:

$$\mathbf{F}_{\text{o}}(u) = \begin{cases} \rho \cdot \frac{r_u + r_v}{|\mathbf{p}_u - \mathbf{p}_v|} \cdot \frac{\mathbf{p}_u - \mathbf{p}_v}{|\mathbf{p}_u - \mathbf{p}_v|} & \text{if } d_{uv} < r_u + r_v \\ 0 & \text{else} \end{cases} \tag{3}$$

We note that while the overlap factor ρ can be unbounded, we will see that there is no need to increase it once there are zero node overlaps.

Finally, each node u has some positive data weight $w_u > 0$, which will be used with the *node-scaling factor* ω to determine its radius r_u. Let w_{max} be the maximum data weight. Then we rescale the weights so that the radius of node u belongs in the range $(0, \omega]$:

$$r_u = \omega \cdot \sqrt{\frac{w_u}{w_{max}}} \tag{4}$$



(a) Force field of anchor force. **(b)** Force vectors applied to overlapping nodes u, v.

Figure 3 Anchor and overlap forces applied in addition to the base force-directed algorithm.

If some input data weights are equal to 0, then we rescale the data to ensure that there are only non-zero radii. Accordingly, the area of the circle corresponding to node u will be proportional to its weight, w_u. Increasing the node-scaling factor ω will spread the nodes as they grow in size, but eventually this will out-scale the constraints of the initial layout.

3 Experiments

We implemented our circle quasi-cartogram drawing algorithm using the Open Graph Drawing Framework (OGDF) [5], and we ran experiments on a computer with 16GB RAM and a 12th Gen Intel Core i5-12400F CPU. We experimentally studied the tradeoffs in various drawing metrics as we adjust the three parameters of the algorithm. Specifically, we measure the impact of various choices a node-scaling factor, ω, an anchor force factor, α, and an overlap force factor, ρ, has on the following drawing metrics:

- The number of node overlaps
- The number of edge crossings
- The average edge length
- The average distance from initial anchor positions

We observe that the number of node overlaps can reach 0 by setting ρ high enough, which often has high priority as it allows statistical data to be represented accurately in the visualization. On the other hand, the number of edge crossings is not necessarily a priority for visualizing geo-referenced data, but it is a natural graph layout metric to observe and can be a proxy for the graph's topology. We mainly use geographic datasets for our experiments, which naturally have few initial edge crossings. The average edge length can also be used as a proxy metric for how well the geography was preserved compared to its value in the initial layout (baseline). Finally, the average distance from the initial anchor positions is a key metric that we do not want to increase too much from the baseline of 0 in the initial layout.

Datasets. Our first dataset, **USA**, is a graph of the 3108 counties of the 48 states in the continental U.S. (which excludes Hawaii and Alaska). We added 7595 edges between

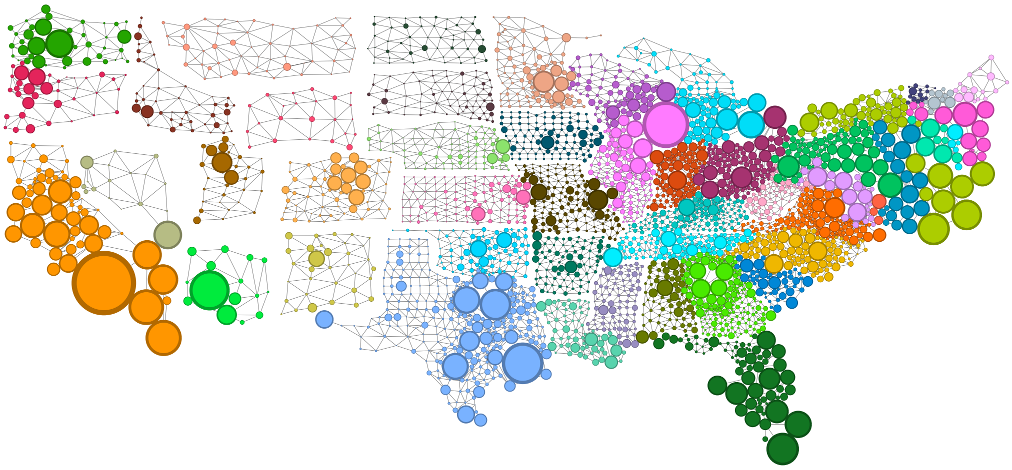


Figure 4 2010 county populations of the continental USA, color-coded by states. It has 0 node overlaps with parameters $\omega = 2500, \alpha = 1, \rho = 3$.

adjacent counties if they belong to the same state, so the underlying graph has 48 connected components, one for each state (each with its own color in Figure 4). For a given county, u, the initial position $\mathbf{q}_u$ was set to the county's centroid and its weight w_u was set to its population in the first dataset. The graph was constructed in this manner to preserve the topology within a state. The Northeast is the noteworthy region here, as it is the most densely populated region of the U.S. and thus we see the most mixing between states there.

Our second dataset, **CA**, is a dual graph of the 8057 census tracts in California, which form 22069 edges between adjacent tracts. For a given census tract, u, the initial position $\mathbf{q}_u$ is its centroid, and its weight w_u is its 2010 population. The baseline geographic layout is shown in Figure 2, along with two runs of the algorithm with $\omega = 100, \alpha = 1, \rho = 1$ and $\omega = 100, \alpha = 1, \rho = 3$. The census tract nodes are color coded by county, of which there are 58 in California. We also note that Figure 2 can be viewed as a more granular representation of the county populations in California seen in Figure 4, which is the connected component in orange on the left.

Parameter Evaluation. We first consider fixing the node-scaling factor, $\omega = 2500$, while changing α or ρ. We compare our algorithm against the "baseline" graph, which is simply the input graph layout for the **USA** dataset with node-scaling factor $\omega = 2500$ that does not run our algorithm. Our results are shown in Figure 5, connected by a dashed line to the layouts for the baseline graph. We note that our algorithm's output with parameters $\alpha = 1$ and $\rho = 0$ is remarkably close to the baseline with respect to the four drawing metrics. Decreasing the anchor factor α increases the average anchor distance as expected, with the slight benefit of fewer node overlaps. This makes sense as a layout that is less constrained to its initial layout provides more space for nodes. However, increasing the overlap factor ρ from 0 achieves the same result of much fewer node overlaps but incurs higher average anchor distance and crossings almost immediately. In fact, we reach exactly 0 node overlaps at $\rho = 3$, and see that there is no real benefit to increasing ρ any further.

In the **CA** dataset, we observe the same trends when it comes to the trade-offs between α and ρ for a fixed node-scaling factor ω. However, increasing ρ also leads to a steady increase

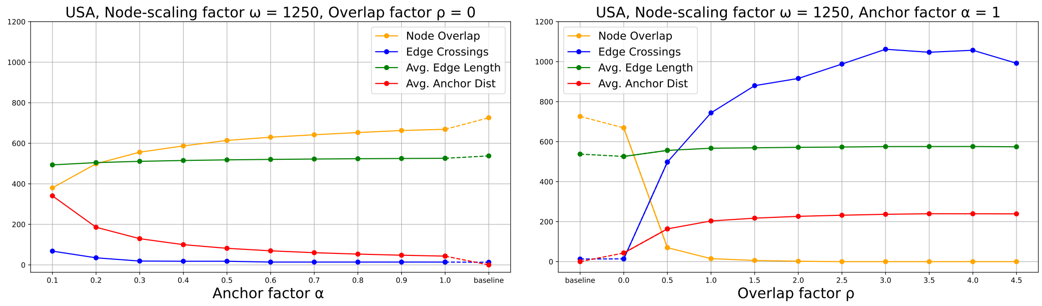


Figure 5 Plots showing graph metrics of USA county graph, fixing ω and varying α or ρ.

in the average edge length and anchor distance. We suspect that this was not as apparent in the **USA** dataset (where both metrics seem to level off) due to the sparser connectivity of the graph. Many of the connected components (the states of the U.S.) could not continue expanding if surrounded by other states, whereas a single connected component would be able to. Regardless, the number of edge crossings increased at a much faster rate and remains the main trade-off for zero node overlaps when increasing ρ.

In the (log-log) plot shown in Figure 6, we see the effects of increasing the node-scaling factor, ω. All four graph metrics increase, so it is up to the user to decide at what point the layout will still be meaningful. When ω is large enough, it eventually out-scales the original layout. Figure 2 shows the data point with $\omega = 100, \alpha = 1, \rho = 1$, which offers a good compromise in preserving the geography while still visualizing the scale of the data. For comparison, the layout that comes before the jump in node overlaps at $\omega = 50$ is shown by Figure 7 in the Appendix.

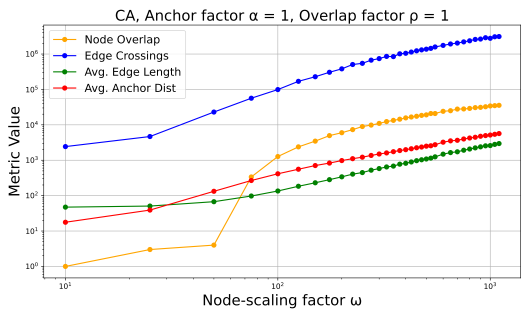


Figure 6 Log-log plot for the effect of the factor ω on various graph qualities of the **CA** graph.

4 Conclusion

In this paper, we introduced a force-directed algorithm for drawing circle quasi-cartograms that explicitly balances geographic fidelity, topological adjacency, and statistical accuracy via controllable node overlap. We can achieve zero node overlaps mainly at the cost of many edge crossings, with slight increases in the average edge length and anchor distance. Although these edge crossings are often hidden by circles in congested parts of the layout, we would like to reduce this metric in future work. If we permit some overlap, the topological and geographical accuracy improves. An interesting future direction would be to model the anchor locations more precisely than as anchor points.

References

1 Md. Jawaherul Alam, Stephen G. Kobourov, and Sankar Veeramoni. Quantitative measures for cartogram generation techniques. *Computer Graphics Forum*, 34(3):351–360, 2015. doi:https://doi.org/10.1111/cgf.12647.

2 Josh Barnes and Piet Hut. A hierarchical $O(N \log N)$ force-calculation algorithm. *Nature*, 324(6096):446–449, December 1986. doi:10.1038/324446a0.

3 Paul B Callahan and S Rao Kosaraju. A decomposition of multidimensional point sets with applications to k-nearest-neighbors and n-body potential fields. *Journal of the ACM*, 42(1):67–90, 1995.

4 Timothy M. Chan. Well-separated pair decomposition in linear time? *Information Processing Letters*, 107(5):138–141, 2008. doi:10.1016/j.ipl.2008.02.008.

5 Markus Chimani, Carsten Gutwenger, Michael Jünger, Gunnar W. Klau, Kerstin Klein, and Petra Mutzel. The open graph drawing framework (ogdf). In *Handbook of Graph Drawing and Visualization*, chapter 17. CRC Press, 2014.

6 Alvin Chiu, Ahmed Eldawy, and Michael T Goodrich. Polygonally anchored graph drawing. In *32nd International Symposium on Graph Drawing and Network Visualization (GD 2024)*, pages 52–1. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPIcs.GD.2024.52.

7 Daniel Dorling. Area cartograms: their use and creation. *The map reader: Theories of mapping practice and cartographic representation*, pages 252–260, 2011.

8 David Eppstein. Edge bounds and degeneracy of triangle-free penny graphs and squaregraphs. *Journal of Graph Algorithms and Applications*, 22(3):483–499, Sep. 2018. doi:10.7155/jgaa.00463.

9 Martin Gronemann. Engineering the fast-multipole-multilevel method for multicore and simd architectures. *Master's thesis. Technische Univ. Dortmund*, 2009.

10 Stefan Hachul and Michael Jünger. Drawing large graphs with a potential-field-based multilevel algorithm. In *Graph Drawing*, pages 285–295. Springer Berlin Heidelberg, 2005.

11 Jan-Hinrich Kämper, Stephen G. Kobourov, and Martin Nöllenburg. Circular-arc cartograms. In *2013 IEEE Pacific Visualization Symposium (PacificVis)*, pages 1–8, 2013. doi:10.1109/PacificVis.2013.6596121.

12 Kelly Lyons, Henk Meijer, and David Rappaport. Algorithms for cluster busting in anchored graph drawing. *Journal of Graph Algorithms and Applications*, 2(1):1–24, 1998.

13 Soeren Nickel, Max Sondag, Wouter Meulemans, Markus Chimani, Stephen Kobourov, Jaakko Peltonen, and Martin Nöllenburg. Computing stable Demers cartograms. In *Graph Drawing and Network Visualization*, pages 46–60, Cham, 2019. Springer International Publishing.

14 Sabrina Nusrat, Md. Jawaherul Alam, and Stephen Kobourov. Evaluating cartogram effectiveness. *IEEE Transactions on Visualization and Computer Graphics*, 24(2):1077–1090, 2018. doi:10.1109/TVCG.2016.2642109.

15 Sabrina Nusrat, Muhammad Jawaherul Alam, Carlos Scheidegger, and Stephen Kobourov. Cartogram visualization for bivariate geo-statistical data. *IEEE Transactions on Visualization and Computer Graphics*, 24(10):2675–2688, 2018. doi:10.1109/TVCG.2017.2765330.

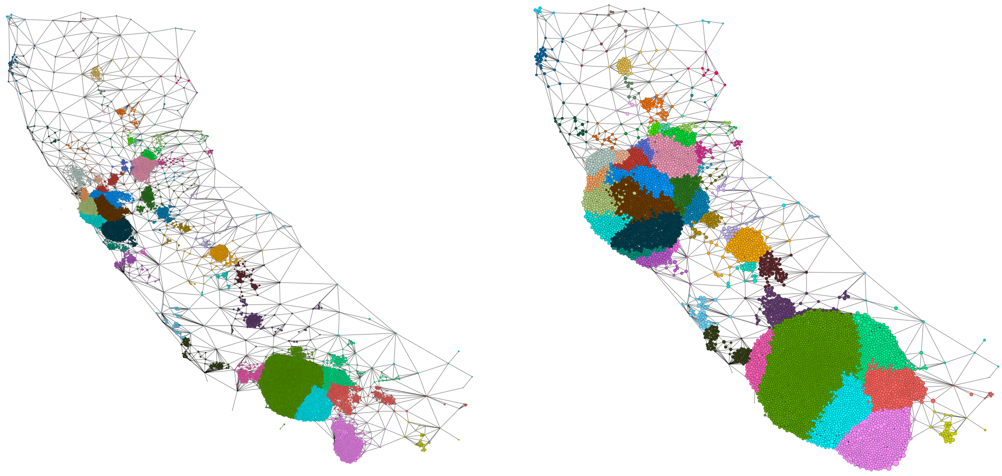
16 Sabrina Nusrat and Stephen Kobourov. The state of the art in cartograms. *Computer Graphics Forum*, 35(3):619–642, 2016. doi:10.1111/cgf.12932.

17 Zhiwei Wei, Su Ding, Wenjia Xu, Lu Cheng, Song Zhang, and Yang Wang. Elastic beam algorithm for generating circular cartograms. *Cartography and Geographic Information Science*, 50(4):371–384, 2023. doi:10.1080/15230406.2023.2196732.

18 Hsiang-Yun Wu, Martin Nöllenburg, and Ivan Viola. Multi-level area balancing of clustered graphs. *IEEE Transactions on Visualization and Computer Graphics*, 28(7):2682–2696, 2022. doi:10.1109/TVCG.2020.3038154.

5 Appendix

The first two datasets **USA** and **CA** can be found at `https://people.csail.mit.edu/ddeford/dual_graphs.html`. Below we show the difference when increasing the node-scaling factor ω, which is a trade-off between geographical accuracy and the scale of data. Orange County (color teal, unfortunately) and San Diego County (color pink) in Southern California do not seem to border in Figure 7a but do in Figure 7b, fully connecting the 5 major counties of Southern California into one big population "blob".



(a) $\omega = 50, \alpha = 1, \rho = 1$ **(b)** $\omega = 100, \alpha = 1, \rho = 1$

Figure 7 The effect of varying the node-scaling factor ω is shown on the **CA** dataset. (a) has 4 node overlaps, while (b) has 1221 node overlaps.

We also show a third dataset **PA** in Figure 8, which can be found at `https://github.com/mggg/GerryChain/blob/main/docs/user/quickstart.rst`:

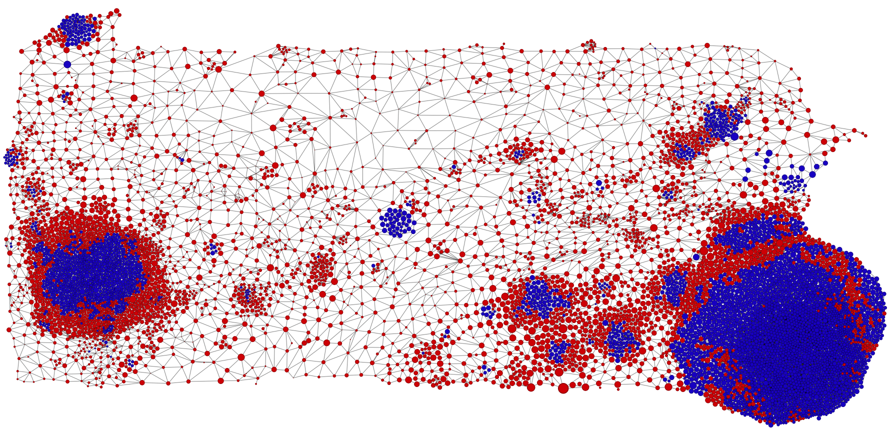


Figure 8 2016 U.S. Election results of Pennsylvania. Its 8921 voting tabulation districts are represented as red nodes if Republican-majority or blue nodes if Democrat-majority, scaled proportional to the number of winning votes. 40 node overlaps with parameters $\omega = 50, \alpha = 1, \rho = 1$.