

GEOMLAB 2020

August 11, 2020

Contents

1	Motivation	2
2	Problem Statement	3
3	Stacking order algorithms for more general Glyphs	4
3.1	Centered nested disks	6
3.2	Nested disks	7
3.3	Pie charts	7
4	Stuff left to do	14
5	Experimental validation	16
6	Sources	18

1 Motivation

Motivated by the COVID-19 disease, several outlets printed two-dimensional depictions of data which usually contained multidimensional datasets. These datasets usually compromise $\{infections, recovered, death\}$ rates with regard to the *country* and sometimes also the evolution over *time*.

In this lab we focus on depicting data in a meaningful way. Besides its quantitative

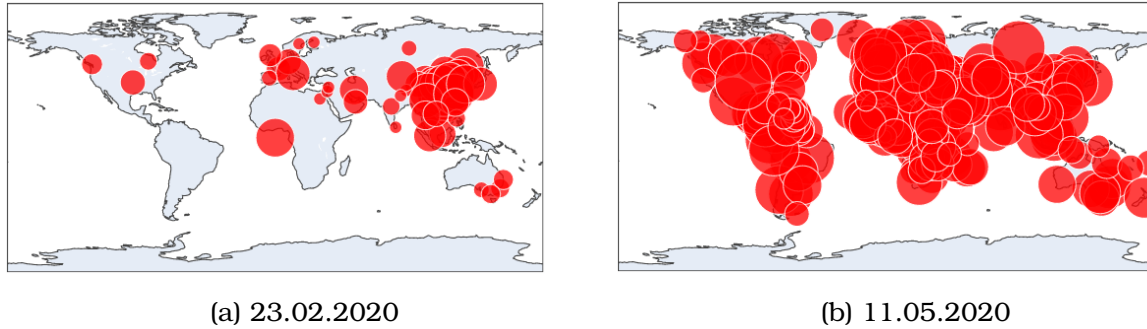


Figure 1: Comparison of a typical scatterplot depicting the *log* of confirmed cases as the radius.

nature this task has some qualitative properties. Similar to many solutions of problems in computational geometry, like deciding if a point is contained in a polygon or the construction of a convex hull for a set of points, the quality of the scatterplot is obvious to the human observer. Not so for a machine. While Fig. 1a is highly informative for a human or potential policy maker, Fig. 1b bears little information content. Despite these facts both figures result from the very same algorithm depicting data from the disease spread of COVID-19 during 2020 [Algo im Appendix].

To investigate the quantification of visual quality mainly Tufte in 1983 in *The visual display of quantitative information graphics press* and Miller et al in *The Need For Metrics In Visual Information Analysis* provided fundamental work. Based on this, follow-up work and further considerations from visibility problems and sorting we develop a novel approach to visualize multidimensional data in a scatterplot which transports as much information content as possible.

Our main resource will be the paper by Sergio Cabello at al *Algorithmic Aspects of Proportional Symbol Maps*. In this paper the hardness of maximizing utility measures with regards to physically realizable drawings was discussed and another type of drawing given by a stacking order was defined. For those drawings some algorithms which we will briefly discuss were derived. Those algorithms were derived for proportional Symbol Maps which only allow us to depict a single quantity. We will try to find more general glyphs and use slightly adapted algorithms to get optimal result with regards to those glyphs and moreover multidimensional data. Hereby, our main focus will be nested circles, centered nested circles and pie charts.

2 Problem Statement

Sergio Cabello at al discussed proportional Symbol Maps. For those we have n circles/disks in R^2 and the radius or size of the circles corresponds to some quantity which is associated to the center point of the circle. The problem is to find a drawing which retains as much of the information about the quantity and the location of the center as possible.

For the definition of a drawing they describe two approaches. One is given by *physically realizable drawings* and the other is *stacking drawings*. As *stacking drawings* are the natural perception and therefore natural restriction of a physically realizable drawing, we define the physically realizable drawing first. A drawing is physically realizable if the order in which the circles or disks – bounded by a perimeter – are stacked upon each other is uniquely determined at every face and no two of such orders conflict. In particular, any two or more disks with a common intersection have a unique ordering. Intutively, this can be imagined by very flexible disks, which somehow maintain their perimeter in the direction of beeing looked at. A stacking drawing is inutively the same thing with stiff disks.

A stacking drawings is realized by a stacking order which is given by a bijection $\phi : [n] \rightarrow [n]$. This bijection describes the order in which the circles are drawn. Circles which are drawn later cover parts over the circles which are drawn earlier. Therefore our task is to find a stacking order such that as little as possible of every circle is covered.

Now we need to discuss what we mean by covering as little as possible. We want that the viewer of the map should be able to identify the size of the circles and the center point. To achieve this goal, we could try to maximize the area of the circles which is not covered, but there are cases in which that would not allow the viewer to get all of the information. The approach we will use is to maximize the visible boundary of our glyph. We can define the visible and occluded boundary for circles formally with

$$U_i^{vis} = \partial C_i \setminus \bigcup_{\phi(j) > \phi(i)} C_j$$

$$U_i^{occ} = \partial C_i \cap \bigcup_{\phi(j) > \phi(i)} C_j.$$

Some results which are given in *Algorithmic Aspects of Proportional Symbol Maps* are the following

Theorem 1. *It is NP-hard to decide if a given collection of congruent disks has a physically realizable drawing where at least some given length of the perimeter of each disk is visible*

This implies that trying to maximize the minimal U_i^{vis} with respect to physically realizable drawings is NP-hard.

Theorem 2. *It is NP-hard to decide if a given collection of disks has a physically realizable drawing whose total visible perimeter is at least a given value.*

This implies that trying to maximize the sum over all the U_i^{vis} with respect to physically realizable drawings is NP-hard.

For stacking orders they give a positive result with regards to the minimum:

Theorem 3. *Given n disks in the plane, a stacking order maximizing the boundary length of the disk that is least visible can be computed in $O(n^2 \log n)$ time.*

If a polynomial algorithm for the sum of all the U_i^{vis} exists remains open to debate.

We won't try to construct an algorithm for the sum problem. Instead we will be looking at glyphs which can depict more information. The first glyph we will discuss are nested disks. Formally every glyph G^i is given by a set of circles G_1^i, \dots, G_k^i such that $r_{j+1} > r_j$ and $G_{j+1}^i \subset G_j^i$ for all $j \in \{1, \dots, k-1\}$. Often times figures using nested disks also center all the discs in the same point. The second type of glyphs we will use are pie charts. A pie chart is formally given by a circle C_i and a set $\{\alpha_1^i, \dots, \alpha_k^i\}$ of dividing lines which we can encode as angle in $[0, 2\pi]$.

In the next section we will be discussing an easy general approach for maximizing special utility functions for glyph stacking order drawings and then apply it to our glyphs and some utility functions.

3 Stacking order algorithms for more general Glyphs

We will be focusing on utility functions which follow the following formula: Maximize the local utility of the worst glyph. In the example of the disk glyphs the local utility would just be the visible perimeter.

We can use a simple greedy algorithm to solve the problem. At each time step just calculate for all discs their visible perimeter as if they were the bottommost disk and then choose the disc C_i with the largest visible perimeter and then draw it. In the next timestep we look at the set of disks without C_i . This is done until we have processed all disks.

We can now formalize this approach for more general objects and prove that the greedy algorithm is optimal with respect to the utility.

Let D be a finite set and for $d \in D$ and $S \subseteq D$ the function $\Gamma(d, S) \mapsto \mathbb{R}_{\geq 0}$ with the property, that for every $d \in D$ we have $S' \subseteq S \Rightarrow \Gamma(d, S') \geq \Gamma(d, S)$.

The set D corresponds to our set of glyphs and Γ corresponds to a utility function, which depends on the insertion order, i.e., the set S describes all the glyphs which lie

```

1  ALG(finite set D)
2     $x := \operatorname{argmax}_{d \in D} \Gamma(d, D \setminus d)$ 
3    return  $[x, \operatorname{ALG}(D \setminus x)]$ 

```

Figure 2: The greedy algorithm

above d . Let $\phi : \{1, \dots, n\} \mapsto D$ be a stacking order. Then the for every glyph $d_i := \phi(i)$ the set of glyphs above d_i is given by $\{d_j \in D \mid j > i\}$.

Theorem 4. *For a given set D and a local utility function Γ the greedy Algorithm as described in Figure 2 returns a stacking order s which maximizes*

$$\lambda(s) = \min_{i \in \{1, \dots, n\}} \Gamma(s(i), \{s(j) \mid j > i\}).$$

Proof. To prove the above theorem we define s^* the stacking order generated by the algorithm. We will then prove that for any other order $s \neq s^*$, there is a modified order s' that fulfills $\lambda(s) \leq \lambda(s')$ and that overlaps with s^* in at least one more additional spot. Consider $s \neq s^*$ and let $i = \min \{ l \mid s(l) \neq s^*(l) \}$ the first index for which s and s^* disagree. The object $s^*(i)$ has a different index according to s , namely, the index is $j = s^{-1}(s^*(i))$ and, because i is the minimal index of disagreement, we have $j > i$.

We modify s as follows. Instead of choosing object $s^*(i)$ only in index j , the new stacking order s' selects it at i as well and increases the index of $s(l)$ by 1 for $l \in [i, \dots, j-1]$. More formally, $s' = s \circ \tau$, where

$$\tau(k) = \begin{cases} k-1, & \text{for } k \in [i+1, \dots, j] \\ j, & \text{for } k = i \\ k, & \text{else} \end{cases}$$

Clearly, $s'(i) = s \circ \tau(i) = s(j) = s^*(j)$ and for $k < i$, $s'(k) = s \circ \tau(k) = s(k) = s^*(k)$, so s' disagrees with s^* only at a later stage. We claim that s' is at least as good of a solution as s , meaning $\lambda(s') \geq \lambda(s)$. If this holds true we can, by iterative application of the previous modification, construct for any stacking order s a non-decreasing sequence s, s', s'', \dots, s^* ending in s^* , the stacking order generated by the algorithm. Thereby proving that s^* is optimal.

The proof of the claim rests on the following observation: Recall that the Γ -value is anti-monotonous, meaning its value increases as the second argument decreases. All the objects that either maintain the same index or get it increased by one only have a decreasing set of objects above them, meaning their values increase. Only the object whose index changes from j to i has a growing set of objects before it. However, this object is selected by the algorithm at index i so it has to be at least as good as the object selected by s .

To make this formal, we prove the following:

$$\forall k' \exists k : \Gamma(s'(k') | \{s'(l), l > k'\}) \geq \Gamma(s(k) | \{s(l), l > k\}) \quad (1)$$

Consider first $k' \notin [i, \dots, j]$, then

$$\Gamma(s(k'), \{s(l), l > k'\}) = \Gamma(s'(k'), \{s'(l), l > k'\})$$

because both $s(k') = s'(k')$ and $\{s(l), l > k'\} = \{s'(l), l > k'\}$.

For $k' \in [i + 1, \dots, j]$, then

$$\Gamma(s'(k'), \{s'(l), l > k'\}) \geq \Gamma(s'(k'), \{s(l), l > k - 1'\}) = \Gamma(s(k' - 1), \{s(l), l > k' - 1\})$$

,where the inequality comes from the second argument on the left-hand side is a subset of the one on the right, so $k = k' - 1$ and lastly for $k' = i$

$$\begin{aligned} \Gamma(s'(k'), \{s'(l), l > k'\}) &= \Gamma(s'(i), \{s'(l), l > i\}) \\ &= \Gamma(s^*(i), \{s^*(l), l > i\}) \geq \Gamma(s(i), \{s(l), l > i\}) \end{aligned}$$

,where the inequality holds because $s^*(i)$ is the choice of the algorithm and thus maximizes precisely this term, so $k = i = k'$ does the job.

In total

$$\begin{aligned} \lambda(s') &= \min_{i \in \{1, \dots, n\}} \Gamma(s'(i), \{s'(j) | j > i\}) \\ &= \Gamma(s'(k'), \{s'(j) | j > k'\}) \text{ for some } k' \\ &\geq^{(1)} \Gamma(s(k), \{s(j) | j > k\}) \\ &\geq \min_{i \in \{1, \dots, n\}} \Gamma(s(i), \{s(j) | j > i\}) = \lambda(s) \end{aligned}$$

proving that s' is as least as good a solution as s . □

We can use this theorem for object of any kind. In particular for the glyphs we discussed before. We now only need to define some reasonable utility functions for our different glyphs and calculate them locally with reasonable run-time.

3.1 Centered nested disks

For centered nested disks we can easily define local utility measures. Every glyph G corresponds to k nested circles therefore we could just calculate the visible perimeter for the k circles and then just combine them by choosing the minimum of the k circles or the sum of the visible perimeter of the k circles. For some set S of objects above G_i we then get formally:

$$\begin{aligned} \Gamma_{min}(G^i, S) &= \min_j^{vis} G_j^i \\ \Gamma_{sum}(G^i, S) &= \sum_{j=1}^k^{vis} G_j^i \end{aligned}$$

We may also be interested in the relative utility which is given by normalizing the size of the circles. This relative approach stops the small circles from dominating big circles. In our applications there is close to no different between the relative utility and

the normal one.

For Γ_{min} it may happen that for all glyphs the smallest circle may be completely covered. Then the utility would be zero for all of the circles. If that happens the algorithm has an optimal solution of value zero. If that happens, we perform the greedy step again and ignore the smallest circle in all of the glyphs until we get a value bigger than zero. This heuristic gives better results than just choosing a random circle.

Remark. *It is NP-hard to calculate physically realizable drawing where $\min \Gamma_{min}$ or $\min \Gamma_{sum}$ is maximized since disc glyphs are a special case of nested discs and both utility measures coincide and are equal to the visible circumference.*

3.2 Nested disks

For dense regions centered nested discs lose a lot of information since the small circles can easily be completely covered up. Therefore we will now derive an algorithm which guarantees, that every nested circle is at least visible. To achieve that we will allow the circles to be moved inside each other. We still want to have $r_{j+1} > r_j$ and $G_{j+1}^i \subset G_j^i$ for all j but the centers can be different.

Our approach is quite simple. We first just look at the biggest circle of each glyph. For those we perform our greedy algorithm with the local utility given by the size of the maximal **continuous** visible perimeter. After we have done that we can take the point p on the circle which is in the middle of the continuous visible perimeter piece and connect it to the center of the big circle by a line. Now every center point for a nested inner circle is uniquely defined by positioning it on that line such that the circle touches the point p . The final glyphs look a little bit like the *Hawaiian earring* topological space.

The resulting drawing guarantees that the perimeter of every circle is at least a little bit visible. The downside of this construction is that the drawing look a little bit strange to the viewer and its more difficult to focus the center of a glyph.

3.3 Pie charts

Recall a pie chart glyph is given by a disk C_i and dividing lines given by a finite set $P_i = \{\alpha_1^i, \dots, \alpha_m^i\}$. The α_j are given by angles in $[0, 2\pi]$. We may rotate the dividing lines like minute or hour hands on a mechanical watch, but the relative distance between all angles must stay the same. This rotation can be described by a single angle due to the fixed relative distance of the dividing lines. Given a rotation of the angles and a set S of glyphs which lie above the pie chart in our drawing, we can define $\Gamma_{pie}(C_i, S)$

as the minimal radial distance of any of the α_j^i to the covered perimeter.

We now need to investigate two things:

- The pie chart glyphs are not a natural generalization of the disk glyphs. Therefore it is not trivial to show that maximizing the discussed utility function with respect to physically realizable drawings is NP-hard.
- 2. How to calculate the optimal rotation given a set over covering glyphs.

NP-hardness

Introduction

Given a finite set D of disks d_i in the plane with centers c_i and positive radii r_i together with a finite set $P_i = \{\alpha_1, \dots, \alpha_{m_i}\}$ of points on the boundary of d_i subdividing the disk into pie-pieces, we wish to determine a physically realizable drawing of D together with a rotation φ_i for each disk, s.t. the minimum distance of any point in $p \in \bigcup P_i$ to the next occluded point on the boundary of d_i is maximized. I.e. we are given Pie-Charts and would like to arrange and rotate them in such a manner that the points where the transition-lines of the pie-chart meet the boundary are as far from being occluded as possible.

NP-hardness

As we will see, the above problem is NP-hard if we allow any physically realizable drawings. To see that, we first fix some notation: We assume that $P_i = \{\alpha_1, \dots, \alpha_{m_i}\}$ is given in angles, i.e. $\alpha_l \in [0, 2\pi)$. Further denote for a given realization Π , $p \in P_i$: $dist_{\Pi}(p) :=$ radial distance to occluded boundary segment on ∂d_i .

Theorem

It is NP-hard to decide, whether for a given set D of disks with boundary points P_i there is a physical realization Π (with rotation), s.t. the

$$\max_{p \in \bigcup P_i} dist_{\Pi}(p) \geq k$$

.

Proof

The proof is inspired and in large parts analogous to [Cabello, Speckmann, Haverkort]. We aim to reduce the NP-hardness from *planar-3-SAT*, i.e. instances \mathcal{I} of 3-SAT with the property to that graph $G(\mathcal{I})$ is planar. The graph $G(\mathcal{I})$ is bipartite, has a node for each variable on one side and each clause on the other. A variable node is connected to the clause iff one of it's literals appears in the clause.

Given a planar instance \mathcal{I} of 3-SAT we need to construct an instance \mathcal{D} of our pie-charts, s.t. \mathcal{I} is satisfiable iff \mathcal{D} has a physical realization with distance $\geq k$ for some suitable k .

All the disks presented have the same radius and we say that two disks overlap by f if the overlapped fraction of the boundary has length f . We introduce the construction of \mathcal{D} .

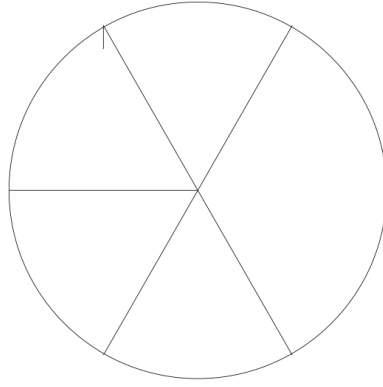


Figure 3: A "neutral" disk used as a connector within a variable ring. The angles are 60 and 120 degrees

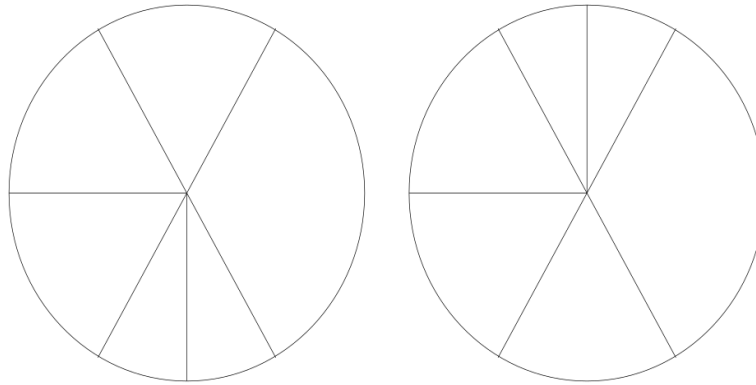


Figure 4: A "true" and "false" disk, similar to the neutral one above, one of the 60 degree segments is cut in half

Construction

We use the three disk types in Figures 1 and 2 to construct for each variable a so-called "gadget". A gadget is a group of neutral disks forming a circle, each overlapping the next by a fraction of $1/6$. Notice that the maximum possible distance of separation points to occluded area is 15 degrees. The gadget is constructed in such a way that it can only be oriented clockwise or counter-clockwise if we want $k > 0$ distance to the occluded area. Figure 3 depicts a prototype of a gadget, all the disks are of type one for now. In reality the number of disks used will be higher because the curvature cannot be this high.

We will interpret a gadget being ordered ascending clockwise as the corresponding variable being "true" and otherwise "false".

For each occurrence of the variable v_i we will, in the gadget g_i substitute a neutral gadget by either a "true" or "false" disk depicted in Figure 2. We use a "false" disk if the literal is appears negated in the clause.

The above segment is of a "true" gadget as the disks are clockwise ascending (I did not figure out how to paint this yet). The "false" and "true" disks will ensure that the added segment will point outwards iff the corresponding literal in the clause is false. Next we will focus on the clauses and how they will be set up. Later we will connect

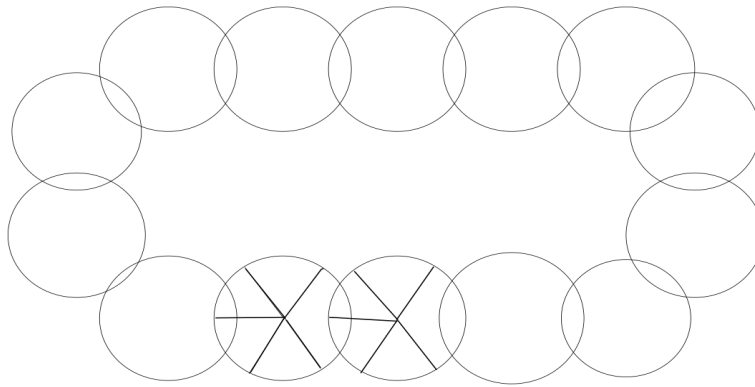


Figure 5: One such gadget for each variable

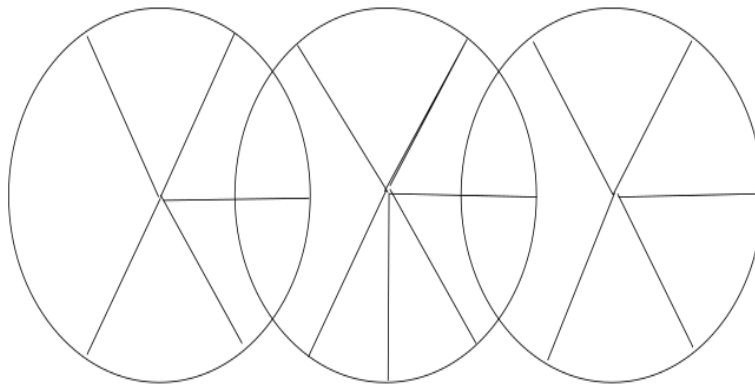


Figure 6: A "true" gadget with a "false" disk

gadgets and clauses (possible because of planarity).

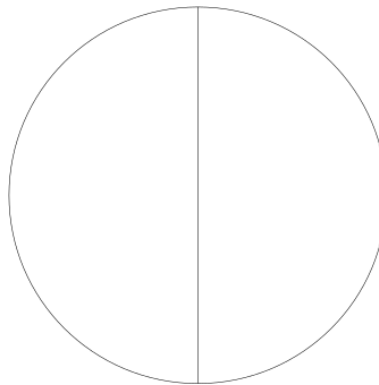


Figure 7: One such disk for each clause

Recall that the instance is satisfiable iff each clause is fulfilled. We instantiate one disk as above for each clause. It has two separating lines exactly opposing one another. For each literal in the clause we overlap it with one of the disks show below by a fraction of $1/4$ as shown in Figure 7.

This connector disk is supposed to be above the clause-disk iff it's associated literal is "false" in the clause.

The clause disk is overlapped by three connector disks, each overlapping 90 degrees and leaving three 30 degree corridors. If the upper connector disk can be rotated as

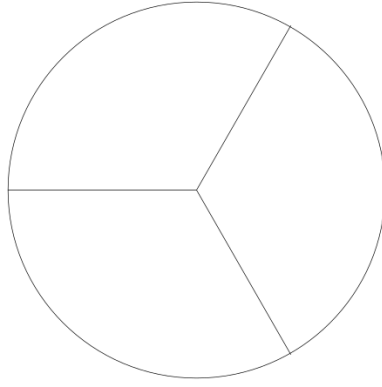


Figure 8: This disk is the endpoint of the edge connecting the gadget to the clause disk

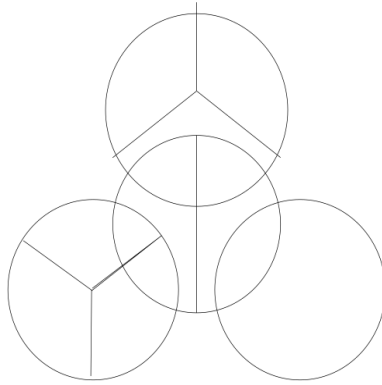


Figure 9: The clause together with the literal connectors

depicted, then it may go under the clause disk, signifying the literal is "true" in the clause. Otherwise it must go over, like the bottom left one. If all three connectors go over the clause, then the clause disk cannot have 15 degrees of space between the separating points and the occluded area. If, conversely, at least one of them goes under, like potentially the top one in Figure 7, then by rotating the clause disk as depicted creates 15 degrees of length between the lines and the occluded area.

Now we need to bring it all together. Recall that we have gadgets that can be oriented clockwise or counterclockwise to signify a "true" or "false" assignment of the variable. The gadgets have for each variable-clause pair a special "true" or "false" disk that will point a separating line outwards iff the *literal* is false in the clause. We also have connectors at the clause that will act as if "false" iff they go over the clause disk, which has to occur if their separating line cannot be arranged as in the top disk of Figure 7.

What we need to ensure is that the connector disk cannot be rotated like that if the literal disk in the gadget points outwards. To do this we introduce the last building block of the construction.

These disks will be arranged in a line connecting the "true"/"false" disks in the gadget with the clause-connector disks at the clause like so:

The top gadget is oriented as "false" and the "true" disk in the middle thus has a

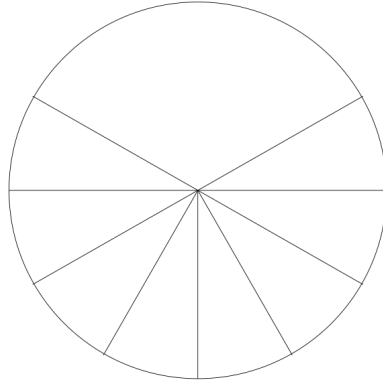


Figure 10: These disks connect the gadgets to the corresponding clauses

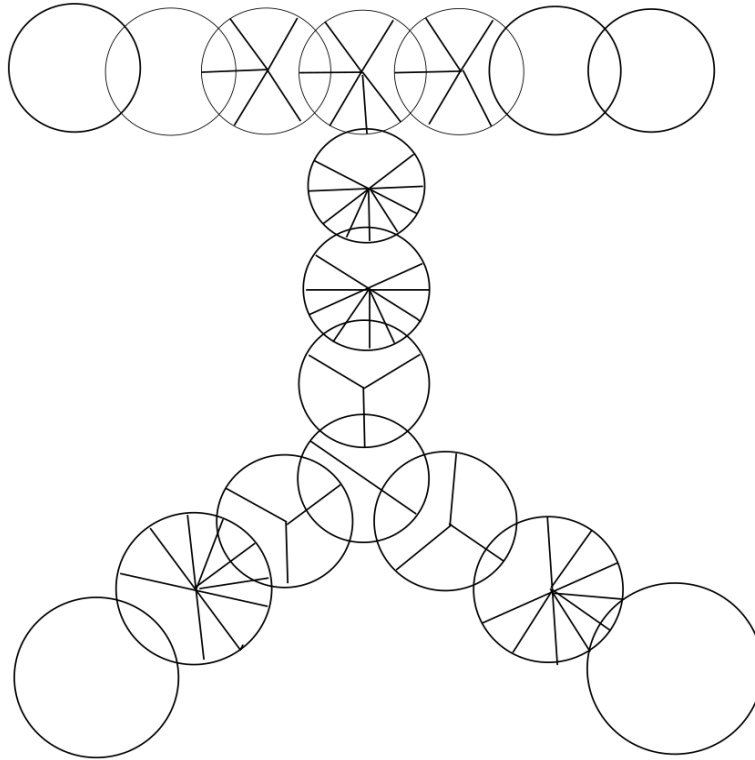


Figure 11: These disks connect the gadgets to the corresponding clauses

separator pointing outward. This forces the first outgoing connector, which overlaps the "true" disk by $1/12 = 30$ degrees to be under it and point its own separators downwards as well. This continues until the clause-connector disk is reached which now cannot be oriented as the bottom-right own and has to go over the clause-disk.

Conclusion

The logical properties were already explained when motivating the construction but will be formalized here again. Notice that the disks in the clause and the gadget can create a distance of 15 degrees between occlusion and separation points. We claim the following. Given a planar instance of 3-SAT \mathcal{I} and the corresponding construction \mathcal{D} as above, \mathcal{I} is satisfiable if and only if there is a physical realization of \mathcal{D} that creates a distance of at least $15 \text{ degrees} = (15/360 * 2\pi)$.

Given a satisfiable instance we orient the gadgets according to the assignment. This will make all "false" connections point a separator outward. Within the gadgets, all disks maintain a distance of at least 15 degrees to the other disks in the gadget. The connector disks at these "false" connections will have to go under so that the outpointing separator has distance ≥ 0 , let alone 15. The "true" connections are not forcing the outgoing connections to go under, we will place them above giving a distance of exactly 15 degrees. This connection line will be going "upwards" (next disk over the previous) until it reaches the clause connector, which now can be placed under the clause-disk enabling it to also create a distance of at least 15 degrees. This implies that all disks can create a distances of 15 degrees to their occluded boundary segments. Conversely, if the constructed instance has a physical realization with degrees ≥ 15 , then clearly in each clause-disk we must have a least one of the connector going below it. This is only possible with distance ≥ 15 if the separators are directed away from the clause disk and thus forcing the rest of the connection to go "upwards" (as viewed from clause to gadget). This means that the first disk in the gadget must be under the last disk in the connecting line. This is only possible if the literal is "true" meaning the variable is "false" and the literal negated or "true" and the literal in normal form. We obtain an orientation of the gadgets that is consistent with all clauses and thus a satisfying assignment of the variables in \mathcal{I} .

4 Stuff left to do

1. Less talking, better figures and drawings. 2. Deal with non-real-RAM (at present, centerpoints may be irrational) 3. Deal with "wobble-space". The precise rotation of the disks is not fixed, there is some interval of possible rotation. We need to convince ourselves that this will not enable unwanted behaviour of the pie-charts that might break the logic.

Optimal rotations for Γ_{pie}

Given a set S of covering glyphs and a fixed pie chart C_i we can calculate the occluded radial intervals I_1, \dots, I_k on the boundary circle of C_i . We now want to find a rotation of the dividing lines that gives us the maximum of the minimal distance of any of the α_j^i to the occluded intervals. We can look at this rotation with respect to some reference line. Here we use α_1 as the reference line.

Now we want to derive all of the positions on the circle where the reference line can be positioned such that none of the dividing lines are covered on the boundary. We will derive the complementary intervals first, i.e., all the intervals that guarantee that at least one dividing line is occluded.

Lemma 1. *Given a set of occluded intervals $I = \{[a_1, b_1], \dots, [a_k, b_k]\}$ (as angles) and a set of dividing lines $\{\alpha_1, \dots, \alpha_m\}$ also given by angles, let β_2, \dots, β_m be the counterclockwise*

distances from α_1 up to α_m with β_j being the distance between α_1 and α_j , and let \mathcal{I} be the set system given by

$$\mathcal{I} = I \cup \bigcup_{i=2}^m \bigcup_{j=1}^k \{[a_j - \beta_i, b_j - \beta_i]\}.$$

Then any rotation which locates α_1 in one of the intervals in \mathcal{I} occludes at least one dividing line.

Proof. This follows from the intervals I being occluded. An occlusion of the intervals results in the dividing line lying in this interval being also included. \square

The complement of \mathcal{I} on the circle describes all of the locations of α_1 which do not occlude any dividing line. This complement may be empty since there may not be a rotation which does not occlude any line. With this observations we now can easily derive an algorithm which returns a rotation which maximizes our utility.

Theorem 5. Given a set of occluded intervals $I = \{[a_1, b_1], \dots, [a_k, b_k]\}$ (as angles) and a set of dividing lines $\{\alpha_1, \dots, \alpha_m\}$ also given by angles, Algorithm 2 returns a rotation which maximizes the minimal radial distance of any of the α_j^i to the covered perimeter.

Proof. Algorithm 2 rotates a given dividing line in such a way, that it lies exactly in the midpoint of the biggest not occluded interval $[a_v, b_v]$. The midpoint in an interval is the point which has the maximum distance from both boundaries, a_v and b_v . As the biggest not occluded interval maps to the circle's perimeter, it also maximizes the distance to the boundaries of the occluded intervals $\{[a_1, b_1], \dots, [a_k, b_k]\}$. \square

```

1 Input:  $I = \{[a_1, b_1], \dots, [a_k, b_k]\} \ \{\alpha_1, \dots, \alpha_m\}$ 
2   calculate  $\mathcal{I} = I \cup \bigcup_{i=2}^m \bigcup_{j=1}^k \{[a_j - \beta_i, b_j - \beta_i]\}$ 
3   let  $\mathcal{V}$  be the set of maximal continuous intervals in  $[0, 2\pi] \setminus \mathcal{I}$ 
4    $v = [a_v, b_v]$  biggest interval in  $\mathcal{V}$ 
5    $r = a_v + (b_v - a_v)/2$ 
6   return  $r$ 

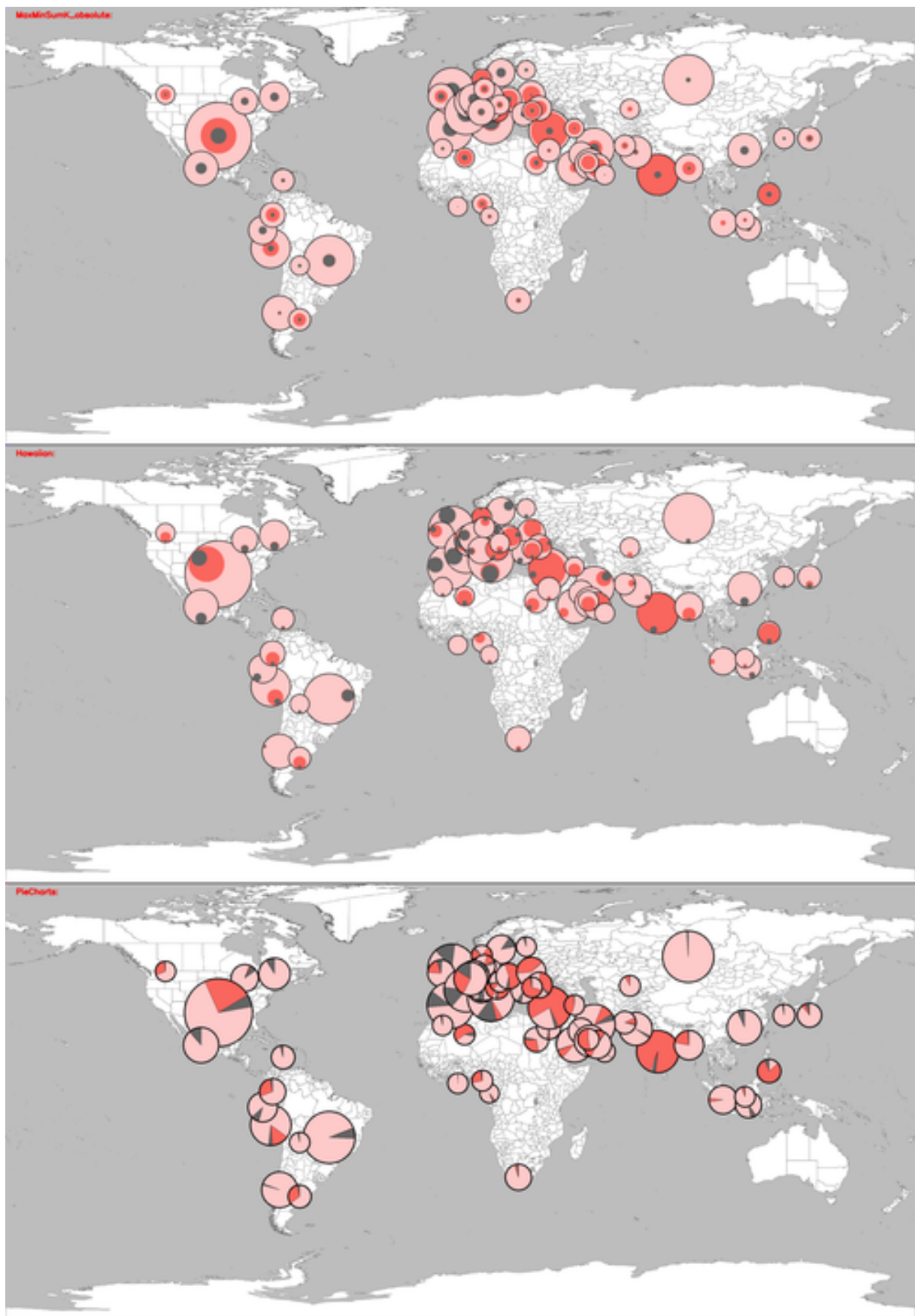
```

Figure 12: Algorithm 2

With this algorithm we can again use our general greedy algorithm for stacking orders.

It may happen again that for all glyphs the local utility is zero. Again we have to use a heuristic to get better results. For instance, we can just ignore one dividing line for each glyphs and try again. This can be done iteratively until the utility is not zero for all glyphs. For the case of exactly one dividing line the utility must be bigger than zero and therefore the heuristic guarantees a better decision than just randomly tie breaking.

5 Experimental validation



6 Sources

Tufte, Edward R. "The visual display of quantitative information graphics press." Cheshire, Connecticut (1983).

Miller, Nancy, et al. "The need for metrics in visual information analysis." Proceedings of the 1997 workshop on New paradigms in information visualization and manipulation. 1997.

Brath, Richard. "Metrics for effective information visualization." Proceedings of VIZ'97: Visualization Conference, Information Visualization Symposium and Parallel Rendering Symposium. IEEE, 1997.

Tatu, Andrada, et al. "Visual quality metrics and human perception: an initial study on 2D projections of large multidimensional data." Proceedings of the International Conference on Advanced Visual Interfaces. 2010.

Urribarri, D. K., & Castro, S. M. (2016). Prediction of data visibility in two-dimensional scatterplots. *Information Visualization*, 16(2), 113–125. doi:10.1177/1473871616638892

Coeurjolly, D., Miguet, S., & Tougne, L. (2004). 2D and 3D visibility in discrete geometry: an application to discrete geodesic paths. *Pattern Recognition Letters*, 25(5), 561–570. doi:10.1016/j.patrec.2003.12.002

Yang-Pelaez J and Flowers WC. Information content measures of visual displays. In: Proceedings of the IEEE symposium on information vizualization 2000 (INFOVIS'00), 2000, pp. 99–103. Washington, DC: IEEE Computer Society.