On Proportional Symbol Maps - An applied perspective

David Göckede, Philip Mayer, Roland Siegert

October 13, 2020

Geometry Lab SS 2020

Overview - ToC

Introduction

Motivation (1/2)

../covid_spread_20200223.png

Motivation (2/2)

../covid_spread_20200223.png

(a) 2020.02.23

Proportional Symbol Maps

Open Access | Published: 12 February 2009

Algorithmic Aspects of Proportional Symbol Maps

Sergio Cabello, Herman Haverkort, Marc van Kreveld & Bettina Speckmann

Algorithmica **58**, 543–565(2010) | Cite this article

Abstract

Proportional symbol maps visualize numerical data associated with point locations by placing a scaled symbol—typically an opaque disk or square—at the corresponding point on a map. The area of each symbol is proportional to the numerical value associated with its location. Every visually meaningful proportional symbol map will contain at least some overlapping symbols. These need to be drawn in such a way that the user can still judge their relative sizes accurately.

Figure 3: Algorithmic Aspects of Proportional Symbol Maps

Physically realizable drawings

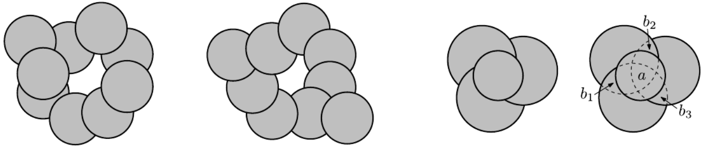


Fig. 4 A stacking drawing (*left*), a physically realizable drawing that is not a stacking drawing (*middle*), a drawing that may seem physically realizable, but is not—any order for face a will conflict with one of b_1, b_2, or b_3 (*right*)

Figure 4: Physically realizable vs. stacking drawings vs. impossible

Utility functions

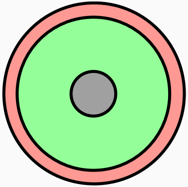
$$\lambda_{min}(D) = \min_{i \in [n]} |U_i^{vis}|$$

$$\lambda_{sum}(D) = \sum_{i=1}^{n} |U_i^{vis}|$$

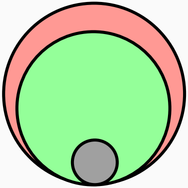
Why this lab?

$$\phi : \{1, ..., n\} \mapsto D$$

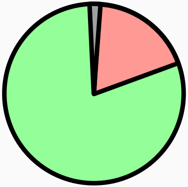
Maps and Glyphs



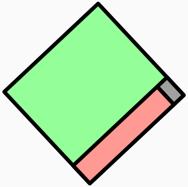
(a) Coffee.



(b) More coffee.



(c) Tasty coffee.



(d) Too much coffee.

Algorithms

Visibility of general glyphs

- assign local utility to each glyph
- define global utility as minimum (or sum) over all local utilities
- we will focus on minimum approaches
- examples
    1. visibility (yes/no)
    2. visible area
    3. visible boundary length absolute or relative
    4. visibility of special points

Greedy Stackings

Theorem: Given as finite set D of n glyphs, a local utility function $\Gamma(d, S) \in \mathbb{R}_{>0}$ for $d \in D$ and $S \subseteq D$, which is anti-monotonous in S and a global utility function λ as the minimum of the local utilities, then the GreedyStacking algorithm computes an optimal stacking order.

Greedy Stackings

Theorem: Given as finite set D of n glyphs, a local utility function $\Gamma(d, S) \in \mathbb{R}_{>0}$ for $d \in D$ and $S \subseteq D$, which is anti-monotonous in S and a global utility function λ as the minimum of the local utilities, then the GreedyStacking algorithm computes an optimal stacking order.

GreedyStacking(D)

if($D \neq \emptyset$) {

$\quad x = argmin_{d \in D} \Gamma(d, D \setminus d)$

$\quad$ *yield* x

$\quad \mathbf{GreedyStacking}(D \setminus x)$

}

GreedyStacking

Give proof.

Centered Nested circles

- natural generalization and seen in action

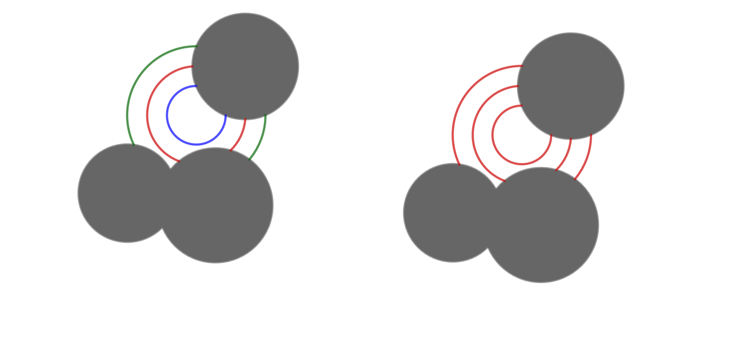


Figure 6: Minimum attained at red circle

- optimization is NP-hard for physically realizable
- greedy runs in $O(n^3 \cdot k^2)$

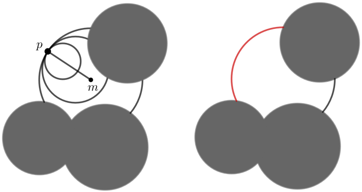
Freely Nested circles

- allow to move disks inside
- same utility as above
- increased freedom comes with increased complexity

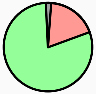
Freely nested disks are to complex to go over all possible placements $\implies$ simplify to Hawaiian Earring setup.

Freely Nested circles

- all disks will be visible
- the greedy algorithm does NOT imply optimality
- verify into experiments
- greedy with longest continuous visible segment



Pies



Seen in practise

Allow individual rotations

We want the separating lines to be clearly visible

Local utility given by the distance of the separating boundary points to the covered area

Pie Utility

Local pie utility given by

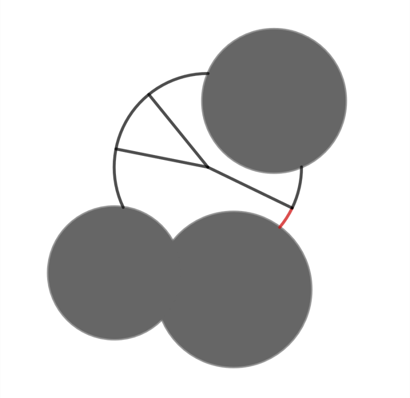
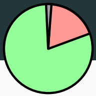


Figure 7: Covered Piechart with relevant segment marked in red

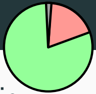
Pie Questions



Two questions arise naturally in the discussion of pies

1. Can we compute the optimal drawing even among physically realizable drawings?
2. How can we rotate the pies such that their local utility is optimal?

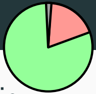
Pie NP-hardness



Theorem: It is NP-hard to decide whether for a given set of pie glyphs there is a physically realizable drawing with global utility greater or equal to some value k.

Corollary: If we find an algorithm that computes an optimal physically realizable drawing in polynomial time, then $P = NP$.

Pie NP-hardness



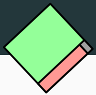
Theorem: It is NP-hard to decide whether for a given set of pie glyphs there is a physically realizable drawing with global utility greater or equal to some value k.

Corollary: If we find an algorithm that computes an optimal physically realizable drawing in polynomial time, then $P = NP$.
Proof: The proof will be sketched briefly.

We reduce this question from *planar*-3SAT. It is known to be NP-complete to decide whether a given 3SAT instance has a satisfying assignment.

We construct a set of pies that have a sufficiently good drawing if and only if the 3SAT instance is satisfiable.

Squares



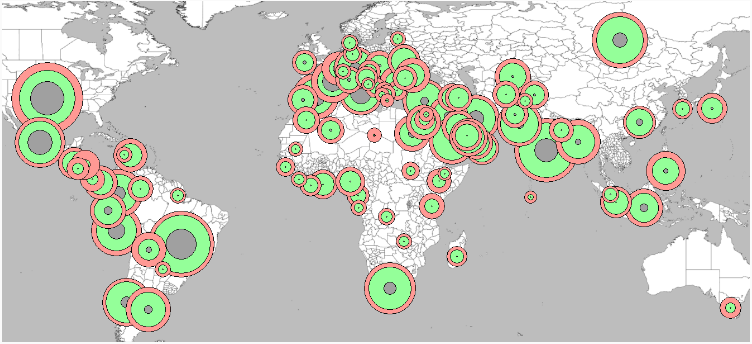
Experimental results

Experimental Setup

- We use the John Hopkins University Covid-19 data
- recovered are coloured green, deceased are coloured black and the infected are coloured red
- logarithmic scaling dependent on two parameters:

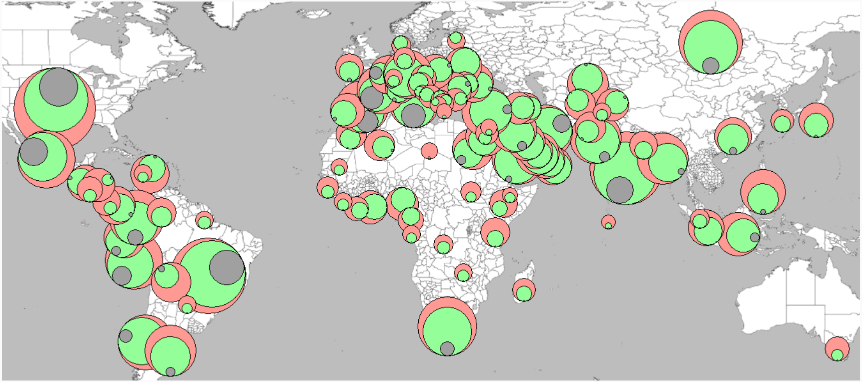
$$r = M * \log\left(\frac{c_i S}{c_{max}} + 1\right)$$

M is the maximum size of a glyph, S is a scaling factor and c_{max} is the maximum number of cases



algorithm	covered	minVis (rel)	minVis (abs)	min one glyph	average rel vis	absolute perc
random	44	0.011 (0)	0.995 (0)	0	0.658	0.677
LeftToRight	42	0.053 (0)	2.189 (0)	0	0.641	0.678
RightToLeft	43	0.053 (0)	0.995 (0)	0	0.656	0.693
Painter	16	0.064 (0)	6.283 (0)	34.991	0.761	0.718
MinMinStacking (abs)	16	0.075 (0)	2.189 (0)	44.467	0.757	0.724
MinMinStacking (rel)	18	0.11 (0)	2.189 (0)	37.327	0.748	0.725
MinSumStacking (abs)	18	0.111 (0)	3.974 (0)	44.467	0.75	0.721
MinSumStacking (rel)	18	0.111 (0)	2.189 (0)	37.327	0.744	0.723

Table 1: Date: 02.08.2020, $n_{records}$: 78, parameters: $M = 50$, $S = 500$ and minimum number of cases $= 5000$



algorithm	covered	minVis (rel)	minVis (abs)	min one Glyph	average rel vis	absolute perc
random	21	0.0001 (0)	0.589 (0)	0	0.765	0.714
LeftToRight	12	0.15 (0)	2.743 (0)	0	0.775	0.725
RightToLeft	13	0.106 (0)	2.89 (0)	0	0.783	0.735
Painter	0	0.093	6.283	47.758	0.857	0.759
our Stacking	0	0.373	6.283	75.034	0.859	0.77

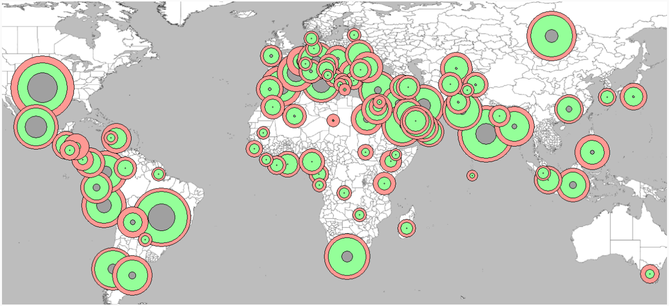
Table 2: date: 02.08.2020 , $M = 50$, $S = 500$ and MnC $= 5000$

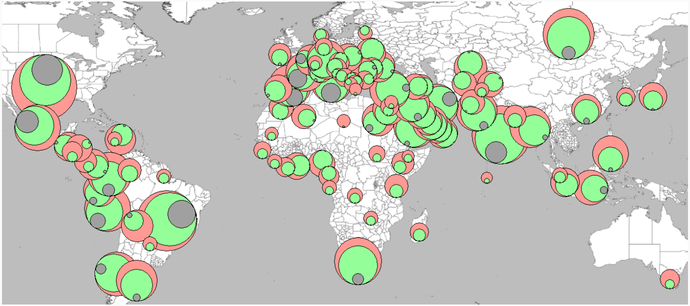
algorithm	covered	minVis (rel)	minVis (abs)	min one glyph	average rel vis	absolute perc
random	44	0.011 (0)	0.995 (0)	0	0.658	0.677
LeftToRight	42	0.053 (0)	2.189 (0)	0	0.641	0.678
RightToLeft	43	0.053 (0)	0.995 (0)	0	0.656	0.693
Painter	16	0.064 (0)	6.283 (0)	34.991	0.761	0.718
MinMinStacking (abs)	16	0.075 (0)	2.189 (0)	44.467	0.757	0.724
MinMinStacking (rel)	18	0.11 (0)	2.189 (0)	37.327	0.748	0.725
MinSumStacking (abs)	18	0.111 (0)	3.974 (0)	44.467	0.75	0.721
MinSumStacking (rel)	18	0.111 (0)	2.189 (0)	37.327	0.744	0.723

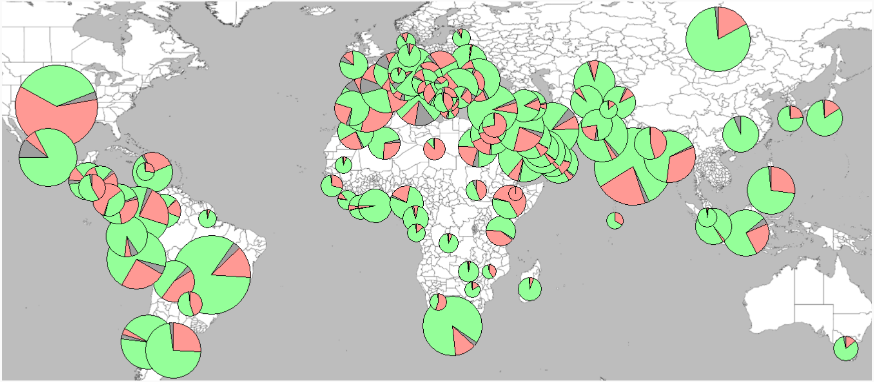
Table 3: centered disks

algorithm	covered	minVis (rel)	minVis (abs)	min one Glyph	average rel vis	absolute perc
random	21	0.0001(0)	0.589 (0)	0	0.765	0.714
LeftToRight	12	0.15 (0)	2.743 (0)	0	0.775	0.725
RightToLeft	13	0.106 (0)	2.89 (0)	0	0.783	0.735
Painter	0	0.093	6.283	47.758	0.857	0.759
our Stacking	0	0.373	6.283	75.034	0.859	0.77

Table 4: date: 02.08.2020 , $M = 50$, $S = 500$ and $\text{MnC} = 5000$

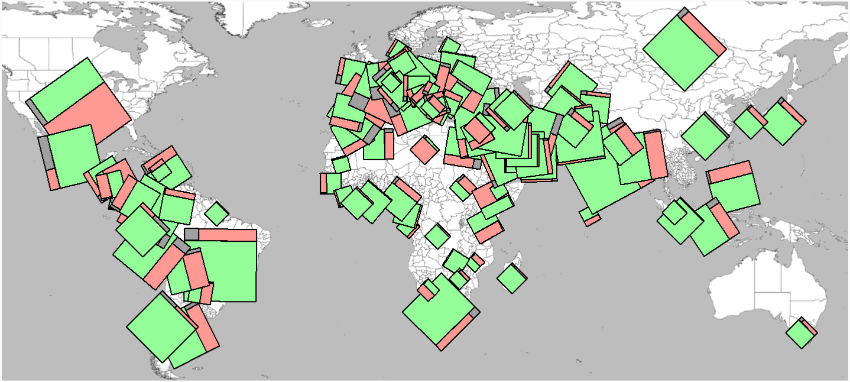






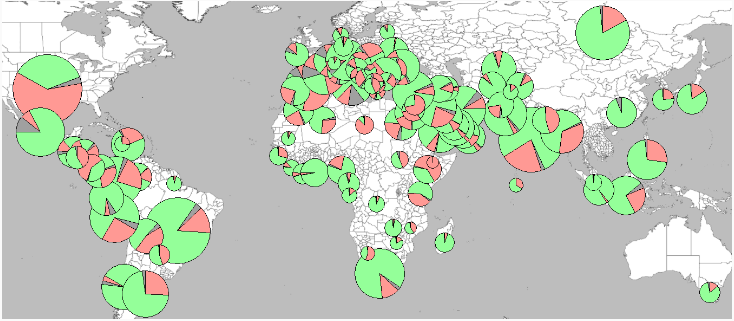
algorithm	covered	minDist (rel)	minDistAvg (rel)	minDistAvg (abs)
Painter+random	80	0.0 (0)	1.01	24.266
random+heuristic	29	0.0 (0)	1.587	42.946
RightToLeft	18	0.017 (0)	1.621	43.407
Painter+ heuristic	5	0.022 (0)	1.706	41.719
our Stacking	0	0.271	1.765	44.452

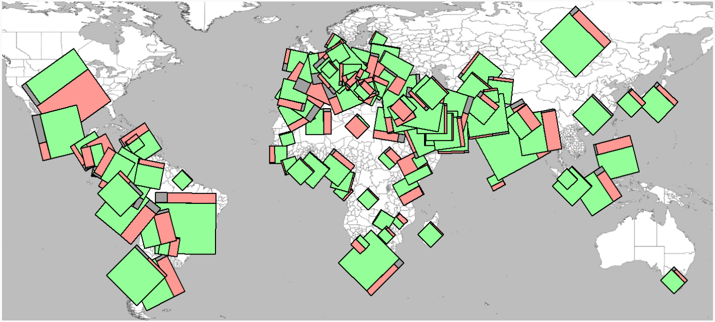
Table 5: date: 22.08.2020, $M = 50$, $S = 500$ and $\mathrm{MnC} = 5000$



algorithm	covered	minDist
random Stacking+random rotations	87	0.235 (0)
Painter+random rotations	41	0.58 (0)
random Stacking+heuristic rotations	56	0.027 (0)
Painter+heuristic	19	0.052 (0)
our Stacking	13	0.052 (0)

Table 6: date: 22.08.2020 , $M = 50$, $S = 500$ and MnC $= 5000$





Exploration in App

Exploration of the data

[Switch to app and play!]

Conclusion and Outlook

Summary

- Four glyphs were shown, with two new approaches.
- NP-hardness of new approaches was outlined.
- Heuristics and greedy approach usually are good choices.
- Square/pie approach can be interpreted as discrete version of the relative visibility.
- All of this was verified on the most recent COVID-19 data,
- and experimentally demonstrated.