INFORMS Journal on Computing
Articles in Advance, pp. 1–9
ISSN 1091-9856 (print) | ISSN 1526-5528 (online)

http://dx.doi.org/10.1287/ijoc.2013.0557
© 2013 INFORMS

Optimizing the Layout of Proportional Symbol Maps: Polyhedra and Computation

Guilherme Kunigami, Pedro J. de Rezende, Cid C. de Souza
Institute of Computing, University of Campinas, Campinas, SP 13083-852, Brazil
{kunigami@gmail.com, rezende@ic.unicamp.br, cid@ic.unicamp.br}

Tallys Yunes
School of Business Administration, University of Miami, Coral Gables, Florida 33124, tallys@miami.edu

Proportional symbol maps are a cartographic tool to assist in the visualization and analysis of quantitative data associated with specific locations, such as earthquake magnitudes, oil well production, and temperature at weather stations. As the name suggests, symbol sizes are proportional to the magnitude of the physical quantities that they represent. We present two novel integer linear programming (ILP) models to solve this computational geometry problem: how to draw opaque disks on a map so as to maximize the total visible border of all disks. We focus on drawings obtained by layering symbols on top of each other, also known as *stacking drawings*. We introduce decomposition techniques as well as several families of facet-defining inequalities, which are used to strengthen the ILP models that are supplied to a commercial solver. We demonstrate the effectiveness of our approach through a series of computational experiments using hundreds of instances generated from real demographic and geophysical data sets. To the best of our knowledge, we are the first to use ILP to tackle this problem, and the first to provide provably optimal symbol maps for those data sets.

Key words: computational geometry; integer programming; cartography; symbol maps
History: Accepted by Karen Aardal, Area Editor for Design and Analysis of Algorithms; received November 2011; revised November 2012; accepted March 2013. Published online in *Articles in Advance*.

1. Introduction

Proportional symbol maps (PSMs) are a cartographic tool to assist in the visualization and analysis of quantitative data associated with specific locations (e.g., earthquake magnitudes, oil well production, temperature at weather stations, etc.). At each location, a symbol is drawn whose size is proportional to the numerical data collected at that point on the map (Cabello et al. 2006, 2010). For our purposes, the symbols are scaled opaque disks, which are typically preferred by users (Griffin 1990), and we focus on drawings obtained by layering symbols on top of each other, also known as *stacking drawings*. Because of overlapping, a drawing of the disks on a plane will expose some of them (either completely or partially) and potentially obscure the others. Although there have been studies about symbol sizing, it is unclear how much the symbols on a PSM should overlap (Dent 1999, Slocum et al. 2003). The quality of a drawing is related to how easily the user is able to correctly judge the relative sizes of the disks. Intuitively, the accuracy of such a judgment is proportional to how much of the disk borders are visible. Figure 1 illustrates why it is better to consider visible border length rather than visible area. As a consequence, the objective function consists of maximizing one of two alternative measures of quality: the minimum visible border length of any disk (the *max-min* problem) or the total visible border length over all disks (the *max-total* problem).

For n disks, Cabello et al. (2006) show that the max-min problem can be solved in $O(n^2 \log n)$ time in general or in $O(n \log n)$ time if no point on the plane is covered by more than $O(1)$ disks. According to Cabello et al. (2006), the complexity of the max-total problem for stacking drawings is open; therefore, we focus on this version of the problem.

The contributions of this work are (i) identifying a new application of integer linear programming (ILP) in computational geometry and proposing two novel ILP formulations for the max-total problem; (ii) introducing decomposition techniques, as well as several families of facet-defining inequalities, for this problem; (iii) experimenting with ILP algorithms that demonstrate the effectiveness of our approach through a series of computational tests on hundreds of instances obtained from real geophysical data from NOAA's National Geophysical Data Center (NOAA Satellite and Information Service 2005); and (iv) providing, for the first time, provably optimal solutions to *all* of the max-total instances studied in Cabello et al. (2006, 2010). As a result, we find that the optimal

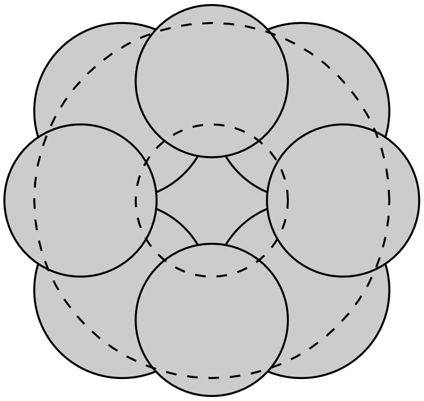

Figure 1 One Cannot Tell Whether the Bottom Disk is Small or Large

solutions are significantly superior to the best heuristic solutions obtained with the algorithm of Cabello et al. (2006, 2010).

The PSM has a clear application to the visualization of statistical data. Good understanding of such information is crucial to strategic decisions, which is at the heart of operations research. For example, knowing the intensity and location of earthquakes is crucial in deciding where to install emergency operations centers to handle such events.

An earlier and much shorter version of some of the results herein, not including our latest and best results, appeared as Kunigami et al. (2011). We are unaware of other attempts at using ILP to solve this problem.

In §2, we describe the problem more formally and introduce some basic terminology. We present two alternative ILP models for the problem in §3 and perform a polyhedral study of those formulations in §4. We describe new families of facet-defining inequalities in §5 and introduce decomposition techniques in §6. The computational results obtained with ILP algorithms based on each of the two formulations appear in §7. Finally, we conclude the paper and propose directions for future research in §8. The proofs of all theoretical results presented in this paper are included in its accompanying online supplement (available as supplemental material at http://dx.doi.org/10.1287/ijoc.2013.0557).

2. Problem Description and Terminology

Let $S = \{1, 2, \ldots, n\}$ be a set of disks with known radii and center coordinates on the Euclidean plane. Let $\mathcal{A}$ be their *arrangement,* defined as the planar subdivision induced by the borders of all the disks in S. In other words, $\mathcal{A}$ is a partition of the Euclidean plane into regions delimited by the borders of the disks in S. A point at which two or more disk borders intersect is called a *vertex* of $\mathcal{A}$. A portion of a disk border

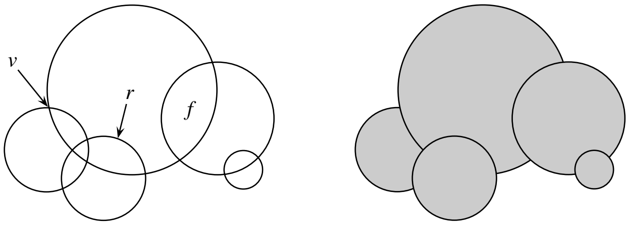

Figure 2 Arrangement with Vertex v, Arc r, and Face f (Left), and a Drawing (Right)

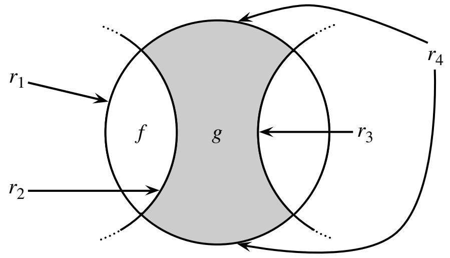

Figure 3 Three Single-Piece Canonical Arcs r_1, r_2, r_3, and a Multipiece Canonical Arc r_4

that connects two vertices, with no other vertices in between, is called an *arc*. A region of $\mathcal{A}$ that is delimited by arcs and does not have any arcs in its interior is called a *face*. A *drawing* of S is a subset of the arcs and vertices of $\mathcal{A}$ that is drawn on top of the filled interiors of the disks in S (see Figure 2). A set of arcs on the boundary of a face that belong to the same disk constitutes a *canonical arc*. In Figure 3, the boundary of face f is made up of canonical arcs r_1 and r_2. The boundary of face g is made up of three canonical arcs: r_2, r_3, and r_4. Note that canonical arc r_4 is composed of two pieces. For simplicity, we will use the term *arc* to mean canonical arc for the remainder of the paper, unless noted otherwise.

Given an arrangement, many drawings are possible, but not all of them represent a sensible, physically feasible, placement of symbols. A *stacking drawing* is obtained by assigning disks to levels (a stacking order) and drawing them, in sequence, from the lowest to the highest level. Such a drawing is made up of a set of arcs A and vertices V taken from $\mathcal{A}$. An arc $r \in \mathcal{A}$ belongs to A if all the disks that contain r in their interior are assigned to levels below the level of the disk containing r in its border. Note that a visible arc of A may be the concatenation of many arcs from $\mathcal{A}$. A vertex $v \in \mathcal{A}$ belongs to V (in the sense that it is a visible point in the drawing) if and only if there exists at least one arc in A that has v as one of its endpoints.

3. Two Alternative ILP Models

Let $G_S = (V, E)$ be an undirected graph with one vertex for every disk $i \in S$ (denoted $V(i)$) and one edge

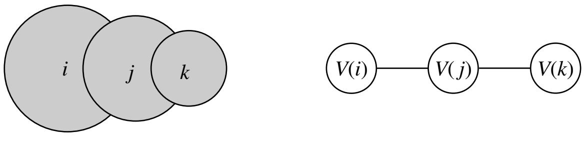

Figure 4 An Optimal Solution (Left) Needs to Assign the Three Disks to Levels 1, 2, and 3, Although the Largest Clique in G_S (Right) Has Size 2

for every pair of vertices whose corresponding disks overlap. Moreover, let $m-1$ be the length of the longest simple path in G_S, and let $\mathscr{K}$ be the set of all maximal cliques of G_S. Although G_S is a graph with special structure, the size of $\mathscr{K}$ can still be exponential in n, as is true for general graphs. This happens because a special case of these graphs are unit disk graphs, which are known to have exponentially many maximal cliques in the worst case (Gupta et al. 2005).

Proposition 1. *The max-total problem for stacking drawings has an optimal solution that uses at most m levels.*

At first, it might seem that the number of levels needed should be no greater than the size of the largest clique in G_S. However, consider the case when G_S is a simple path with $m>2$ vertices. The largest clique in G_S has size 2, but an optimal solution may need to use m levels. Figure 4 shows an example with $S=\{i,j,k\}$ and $m=3$.

From the set S, its arrangement $\mathscr{A}$ can be computed in $O(n^2)$ time. Having $\mathscr{A}$, the following data, which serve as input to our ILP models, can be calculated in polynomial time bounded by the total cardinality of all sets S_r^I, which does not exceed $O(n^3)$, in the worst case:

- $R \equiv$ set of all arcs in $\mathscr{A}$.
- $l_r \equiv$ length of arc $r \in R$ (total length if r has multiple pieces).
- $d_r \equiv$ disk that contains arc r in its border.
- $S_r^I \equiv$ set of disks that contain arc r in their interior.

Now we describe our first model: For each $r \in R$, let the binary variable x_r be equal to 1 if arc r is visible in the drawing, and 0 otherwise. Then, the objective is to maximize

$$\sum_{r\in R} l_r x_r. \tag{1}$$

We assume that $m \geq 2$ because it is trivial to find the optimal solution when $m=1$. For each disk $i \in S$, let the binary variable y_{ip} be equal to 1 if disk i is at level p ($1 \leq p \leq m$, with 1 being the bottom level and m being the top level), and equal to 0 otherwise. A stacking drawing has to satisfy the following constraints:

$$\sum_{p=1}^{m} y_{ip} \leq 1, \quad \forall\, i \in S, \tag{2}$$

$$x_r \leq \sum_{p=1}^{m} y_{d_r p}, \quad \forall\, r \in R, \tag{3}$$

$$\sum_{i:V(i)\in K} y_{ip} \leq 1, \quad \forall\, 1 \leq p \leq m,\ K \in \mathscr{K}, \tag{4}$$

$$\sum_{a=1}^{p} y_{d_r a} + \sum_{b=p}^{m} y_{ib} + x_r \leq 2,$$

$$\forall\, r \in R,\ i \in S_r^I,\ 1 \leq p \leq m, \tag{5}$$

$$x_r \in \{0,1\}, \quad \forall\, r \in R, \tag{6}$$

$$y_{ip} \in \{0,1\}, \quad \forall\, i \in S,\ 1 \leq p \leq m. \tag{7}$$

We refer to the convex hull of feasible integer solutions to (2)–(7) as P_1. Constraint (2) states that each disk is assigned to at most one level. Because of Proposition 1, and because assigning a disk to the lowest level never decreases the objective function value, any optimal solution to (2)–(7) can be converted to another solution with the same value and having all disks assigned to at most m levels. Hence, we use $\leq$ instead of $=$ in (2) to prevent P_1 from losing dimension, turning the study of the facial structure of this polytope (see §4) technically simpler. Constraint (3) states that a disk with a visible arc must be assigned to a level, and (4) says that overlapping disks cannot be at the same level. Although the latter constraints can be exponential in number, a compact formulation can easily be obtained by replacing them with simple constraints stating that a disk cannot be on two different levels simultaneously. However, we prefer to present the stronger form of constraints (4), which derives directly from previous studies on the independent set polytope (see Padberg 1973). Constraint (5) ensures that arc r is only visible if d_r is above all other disks that contain r.

Our second model is related to the partial order polytope (Müller 1996) (see §5 for further details). It uses the same x_r variables introduced in the first model but replaces variables y_{ip} with new binary variables w_{ij} for every pair of distinct disks $i, j \in S$. If $w_{ij}=1$, it means that disk i is placed above disk j. The constraints are as follows:

$$w_{ij} + w_{ji} \leq 1, \quad \forall\, i,j \in S,\ i<j, \tag{8}$$

$$x_r \leq w_{d_r j}, \quad \forall\, r \in R,\ j \in S_r^I, \tag{9}$$

$$w_{ij} + w_{jk} - w_{ik} \leq 1, \quad \forall\, i,j,k \in S,\ i \neq j \neq k \neq i, \tag{10}$$

$$x_r \in \{0,1\}, \quad \forall\, r \in R, \tag{11}$$

$$w_{ij} \in \{0,1\}, \quad \forall\, i,j \in S,\ i \neq j. \tag{12}$$

We refer to the convex hull of feasible integer solutions to (8)–(12) as P_2. Constraint (8) states that either i is above j or vice versa. Constraint (9) states that if arc r is visible, its disk d_r has to be above all other disks that contain r in their interior. Finally, (10) makes sure that the (partial) order imposed by the w_{ij} variables is transitive.

Note that both formulations accept partial drawings as feasible solutions. By a partial drawing we mean a set of arcs that is a subset of a drawing. However, because our objective is to maximize a linear function with nonnegative coefficients, any optimal solution to these models must be a complete drawing.

4. Polyhedral Results

In this section, we obtain the dimension of P_1 and $P_2$and determine which inequalities in their original formulations define facets.

4.1. Polyhedral Study of P_1

Proposition 2. *The dimension of P_1 is $nm+|R|$.*

Proposition 3. *Given an arc $r \in R$, the inequality $x_r \geq 0$ defines a facet of P_1, whereas the inequality $x_r \leq 1$ does not.*

Proposition 4. *Given a disk $i \in S$ and a level $1 \leq p \leq m$, the inequality $y_{ip} \geq 0$ defines a facet of P_1, whereas the inequality $y_{ip} \leq 1$ does not.*

Proposition 5. *Given a disk $i \in S$,* (2) *defines a facet of P_1.*

Proposition 6. *Given an arc $r \in R$,* (3) *defines a facet of P_1.*

Proposition 7. *Given $1 \leq p \leq m$ and $K \in \mathcal{K}$ with $|K| \geq 2$,* (4) *defines a facet of P_1.*

Proposition 8. *Given an arc $r \in R$, $i \in S_r^I$, and $1 \leq p \leq m$,* (5) *does not define a facet of P_1, but* (13) *does if $1 \leq p < m$:*

$$\sum_{a=1}^{p} y_{d_r a} + \sum_{b=p}^{m} y_{ib} + x_r \leq 1 + \sum_{a=1}^{m} y_{d_r a}. \tag{13}$$

4.2. Polyhedral Study of P_2

Proposition 9. *The dimension of P_2 is $n(n-1)+|R|$.*

Proposition 10. *Given an arc $r \in R$, the inequality $x_r \geq 0$ defines a facet of P_2, whereas the inequality $x_r \leq 1$ defines a facet of P_2 only when $S_r^I = \varnothing$.*

Proposition 11. *Given two distinct disks $i, j \in S$, the inequality $w_{ij} \geq 0$ defines a facet of P_2 if $S_r^I = \emptyset$ for all arcs r on the border of disk i. Moreover, the inequality $w_{ij} \leq 1$ does not define a facet of P_2.*

Proposition 12. *Given two disks $i, j \in S$ with $i < j$,* (8) *defines a facet of P_2.*

Proposition 13. *Given an arc $r \in R$ and a disk $j \in S_r^I$,* (9) *defines a facet of P_2.*

Proposition 14. *Given three distinct disks $i, j, k \in S$,* (10) *defines a facet of P_2.*

4.3. Relationship Between P_1 and P_2

We conclude this section with a formal comparison of our two ILP formulations. Let $\tilde{P}_1$ be the feasible set of the linear relaxation of (2)–(7) and let $\tilde{P}_2$ be the feasible set of the linear relaxation of (8)–(12). Moreover, let $\tilde{P}_1^x$ and $\tilde{P}_2^x$ be the projections of $\tilde{P}_1$ and $\tilde{P}_2$ respectively, onto the x-space. More specifically, $\tilde{P}_1^x = \{x \in \mathbb{R}^{|R|} \mid (y, x) \in \tilde{P}_1$ for some $y \in \mathbb{R}^{nm}\}$ and $\tilde{P}_2^x = \{x \in \mathbb{R}^{|R|} \mid (w, x) \in \tilde{P}_2$ for some $w \in \mathbb{R}^{n(n-1)}\}$.

Proposition 15. *$\tilde{P}_1^x \not\subseteq \tilde{P}_2^x$.*

Moreover, we conjecture that $\tilde{P}_2^x \subset \tilde{P}_1^x$.

5. Strengthening the ILP Formulations

The geometric nature of PSMs enables us to obtain new valid inequalities by observing that certain groups of arcs cannot be visible simultaneously because of physical impossibility. In the sequel, $\mathcal{A}$ is an arrangement of disks on a plane. We use the following additional data sets:

- $D_f \equiv$ set of disks that contain face f.
- $B_f \equiv$ set of arcs that form the boundary of face f. $B_f^+ = \{r \in B_f \mid d_r \in D_f\}$ and $B_f^- = B_f \setminus B_f^+$.
- $I_f \equiv$ set of disks whose borders contain an arc in B_f.
- $C_f \equiv$ set of disks that contain face f in their interior ($C_f = D_f \setminus I_f$).

Consider the arrangement in Figure 3. The boundary of face g is formed by arcs r_2, r_3, and r_4. We have $B_g = \{r_2, r_3, r_4\}$, $D_g = \{d_{r_4}\}$, $B_g^+ = \{r_4\}$, $B_g^- = \{r_2, r_3\}$, $I_g = \{d_{r_2}, d_{r_3}, d_{r_4}\}$, and $C_g = \emptyset$.

In the arrangement of Figure 5, the boundary of face f is formed by arcs r_1, r_2, and r_3. Therefore, we have $B_f = B_f^+ = \{r_1, r_2, r_3\}$, $D_f = \{d_{r_1}, d_{r_2}, d_{r_3}, d_{r_4}\}$, $I_f = \{d_{r_1}, d_{r_2}, d_{r_3}\}$, and $C_f = \{d_{r_4}\}$. If one of the arcs in B_f is visible in a drawing, the other two cannot be visible.

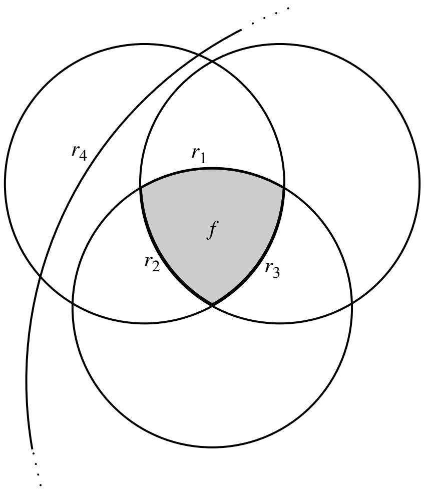

Figure 5 **Arcs r_1, r_2, and r_3 of Face f Cannot Be Visible Simultaneously**

Moreover, if d_{r_4} is assigned to the topmost level m, f will not be visible. This leads to the valid inequality $y_{d_{r_4}m} + x_{r_1} + x_{r_2} + x_{r_3} \leq 1$. In general, we have the following result:

Proposition 16. *Let f be a face of $\mathcal{A}$ with $|B_f^+| \geq 1$. If $|C_f| \geq 1$ or $|B_f^+| \geq 2$, then* (14) *defines a facet of P_1:*

$$\sum_{i \in C_f} y_{im} + \sum_{r \in B_f^+} x_r \leq 1. \tag{14}$$

Proposition 17. *Let f be a face of $\mathcal{A}$ with $|B_f^-| \geq 1$. For each $r \in B_f^-$,* (15) *defines a facet of P_1:*

$$\sum_{i \in D_f} y_{im} + x_r \leq 1. \tag{15}$$

Now let G_R be a graph with one node for every arc $r \in R$, denoted $V(r)$, and an edge between two nodes $V(r_1)$ and $V(r_2)$ if $d_{r_1} \in S_{r_2}^I$ and $d_{r_2} \in S_{r_1}^I$ (i.e., r_1 and r_2 cannot be visible simultaneously). Given a clique K of G_R, to simplify notation we will treat K as a set of arcs whose corresponding vertices induce a clique in G_R. Therefore, we can apply typical set operations to K, such as writing $r \in K$ to indicate that $V(r)$ is a node of the clique and writing $|K|$ to indicate the number of nodes in the clique.

In an attempt to find the counterpart of Proposition 16 for P_2, we obtained the following result.

Proposition 18. *Let K be a maximal clique in G_R with $|K| \geq 3$. Then* (16) *defines a facet of P_2:*

$$\sum_{r \in K} x_r \leq 1. \tag{16}$$

Because the w_{ij} variables define a partial order on the disks in S, P_2 can be viewed as a lifted partial order polytope (POP) with side constraints (Müller 1996). Therefore, one could use valid inequalities for the POP as a starting point for finding valid inequalities for P_2. The odd closed-walk inequality studied in Müller (1996) is one such example.

Proposition 19. *Let $D_S = (V, A)$ be a complete directed graph with one node in V for every disk in S. As before, $V(i)$ denotes the node corresponding to disk i, and an arc from $V(i)$ to $V(j)$ in D_S corresponds to the variable w_{ij}. Let $C = (V(i_1), \ldots, V(i_k), V(i_{k+1}))$, with $i_{k+1} = i_1$ and $i_{k+2} = i_2$, be an odd cycle of length k in D_S. Then* (17) *defines a facet of P_2:*

$$\sum_{a=1}^{k} w_{i_a i_{a+1}} - \sum_{a=1}^{k} w_{i_a i_{a+2}} \leq \frac{k-1}{2}. \tag{17}$$

A vertex of an arrangement is *nondegenerate* if it is an intersection point of exactly two disks or, equivalently, four arcs, as shown in Figure 6(i). Since each arc can be either visible or not, there are 16 potential assignments of values to their respective x variables. Because of our objective function (1), drawings that are candidates for optimal solutions can only include the five assignments shown in Figure 6(ii)–(vi) (dashed arcs are obscured). Assignments such as the ones shown in Figure 6(vii)–(viii) cannot be part of an optimal drawing. This observation gives rise to inequalities (18)–(21):

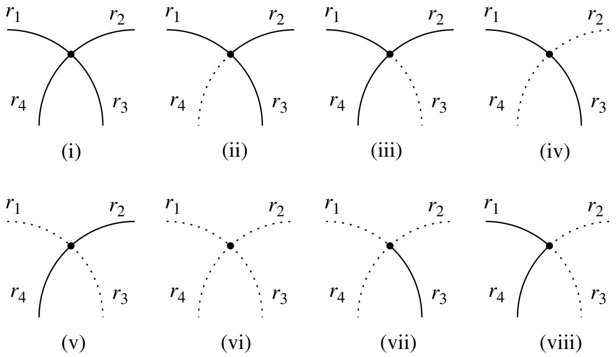

Figure 6 A Nondegenerate Vertex (i), Five Feasible Arc Configurations: (ii)–(vi), and Two Infeasible Ones: (vii) and (viii)

$$x_{r_1} \geq x_{r_3}, \tag{18}$$

$$x_{r_2} \geq x_{r_4}, \tag{19}$$

$$x_{r_3} + x_{r_4} \geq x_{r_1}, \tag{20}$$

$$x_{r_3} + x_{r_4} \geq x_{r_2}. \tag{21}$$

Because the definitions of P_1 and P_2 accept partial drawings as feasible solutions, (18)–(21) are not strictly valid for P_1 and P_2. Nevertheless, since optimal solutions are complete drawings that never violate (18)–(21), these inequalities can still be used to speed up the search.

6. Decomposition Techniques

To reduce the size of the ILP model, we introduce decomposition techniques that allow us to consider smaller sets of disks at a time.

Without loss of generality, we assume that G_S is connected. Otherwise, each of its connected components can be treated separately. A graph is 2-connected if it cannot be disconnected by removing fewer than two vertices. When the graph is connected but not 2-connected, the disconnecting vertices are known as a *cut vertices*, or *articulation points* (West 2001). If G_S is not 2-connected, we can decompose it around its articulation points. Consider the example in Figure 7(i), in which $S = \{a, b, c, d, e, v\}$. The node corresponding to disk v, i.e., $V(v)$, is an articulation point of G_S because its removal disconnects the graph into three connected components: $\{a, b\}$, $\{c, d\}$, and $\{e\}$. By adding v to each of these components,

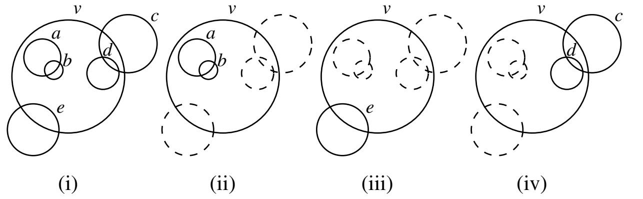

Figure 7 An Instance That Allows for Decomposition

we get instances (ii), (iii), and (iv) of Figure 7, which are solved independently. Those three optimal solutions can be combined into an optimal solution for the entire set S by preserving the relative order of the disks in each solution. Proposition 20 formalizes this idea.

Proposition 20. *Let S be a set of disks such that G_S is not 2-connected, and let v be a disk corresponding to an articulation point of G_S. Let S_k contain v plus the disk set of the k-th connected component obtained after the removal of $V(v)$ from G_S. The optimal solutions for each S_k can be combined into an optimal solution for S in polynomial time.*

If the graph of a connected component (G_{S_k}) is not 2-connected and has an articulation point, the previous procedure can be applied recursively.

From Figure 7(ii), it is clear that there exists an optimal solution in which a and b are drawn above v. Hence, we can consider a and b as a separate instance, and v as another. Proposition 21 formalizes this idea.

Proposition 21. *Let S be a set of disks and let H_S be a directed graph with one node for every disk in S and an arc from node i to j whenever a portion of the border of i's disk is contained in the interior of j's disk. Let S_k be the disk set of the k-th strongly connected component of H_S. The optimal solutions for each S_k can be combined into an optimal solution for S in polynomial time.*

7. Computational Experiments

Our experiments are performed on the same set of instances used in Cabello et al. (2006). Instances *City 156* and *City 538* represent the 156th and 538th largest American cities, respectively, in which the area of each disk is proportional to the city's population. Instances *deaths* and *magnitudes* represent the death count and Richter scale magnitude of 602 earthquakes worldwide, respectively. Disks are placed at the epicenters of each earthquake, and disk areas are proportional to the corresponding quantities (NOAA Satellite and Information Service 2005). When disks in an instance coincide, we replace them by a single disk whose border is the total border length of the original disks. This is possible because we can assume that such disks would occupy adjacent levels in an optimal solution. This preprocessing step reduces the number of disks in *deaths* and *magnitudes* to 573 and 491, respectively.

Table 1 Number of Components and Largest Component Before and After Decomposition

Instance	# disks	Connected	Strongly connected	2-connected
City 156	156	38 (57)	45 (56)	53 (29)
City 538	538	185 (98)	213 (94)	240 (53)
Deaths	573	134 (141)	317 (85)	333 (70)
Magnitudes	491	31 (155)	31 (155)	45 (116)

Part of the success of our approach depends on using our decomposition techniques to break down those large original instances into hundreds of smaller instances, as explained next.

In Table 1, column Connected shows the number of connected components in G_S for each instance, with the number of disks in the largest component in parentheses. Column Strongly connected shows the resulting number of components (and largest component) after we apply the decomposition of Proposition 21. Proposition 20 yields further decomposition, as shown under column 2-connected. The reductions in problem size are remarkable. *City 538* can now be solved by optimizing over sets of disks no larger than one-tenth of its original size. Before applying our decompositions, the largest instance in the original data set had 573 disks. As shown in the rightmost column of Table 1, the largest instance after decomposition has 116 disks. Solving the original instances is now equivalent to solving 671 significantly smaller instances.

We solved the strengthened versions of the two ILP models described in §3 with a commercial solver running its standard branch-and-cut algorithm. Thus, the cuts that are separated during the execution of the algorithm are restricted to those already implemented inside the solver. From now on, we refer to the first model ((2)–(7)) as model M_1, and we refer to the second model ((8)–(12)) as model M_2. We now present some implementation details for each case.

In model M_1, we implement (2) as SOS1, substitute (13) for (5), and add (14) and (18)–(21) at the root node. (Inequalities (15) did not help computationally.) Because $|\mathscr{K}|$ can be exponentially large, rather than including all of (4), we heuristically look for an edge covering of G_S by maximal cliques (Nemhauser and Sigismondi 1992). Alternatively, we also tried replacing (4) with $y_{ip} + y_{jp} \leq 1$ for each level p and all $(i, j) \in E$. Although theoretically weaker, the latter formulation performed better in our experiments. This might be explained by the sparser coefficient matrix of the weaker model, which typically yields easier-to-solve linear relaxations. Finally, instead of computing the exact value of m as in Proposition 1, which is NP-hard (Garey and Johnson 1979), we use $m = n$ in every run, because the exact m is equal to n in many of the large components.

In model M_2, we create a variable w_{ij} for all pairs $i, j \in S$ with $i \neq j$. This way, (10) prevents any cyclic orientation among disks. In addition, we implement (8) as an equality rather than an inequality because our computational experience suggests that the equality form yields better running times. Finally, we include (16) and (18)–(21) at the root node. For (16) in particular, instead of looking for all maximal cliques in G_R as described in Proposition 18, we only consider cliques corresponding to the arcs of B_f^+ for each face f of the arrangement. Although such cliques are not necessarily maximal, they can be found efficiently and already provide good results (see later).

Our implementation was done in C++, using CGAL for data extraction (Wein et al. 2007). We use XPRESS-Optimizer (Fair Isaac Corporation 2009) version 20.00 to solve each problem on a 2.4 GHz Intel® Core™2 Quad Processor with 4 GB RAM. Unless noted otherwise, we limit each run to five hours of CPU time.

For comparison purposes, we use the $O(n^2 \log n)$ heuristic from Cabello et al. (2006, 2010) to find good feasible solutions. Despite being a max-min heuristic, its solutions also perform well in terms of the max-total objective.

7.1. Results Obtained with Model M_1

Out of the 671 components obtained through decomposition, all but the five or six largest ones from each original instance are easily solved by our optimization algorithm. We will focus on them first.

For components with $|S_k| \leq 2$, the solution is trivial. For the remaining easy-to-solve components, we summarize our results in Table 2. Column Comp. $w/|S_k| > 2$ indicates how many easy components from the corresponding original instance have more than two disks. The next nine columns indicate the minimum, average, and maximum values of component size, followed by the number of search nodes and CPU time required to find an optimal solution, respectively. When compared to the heuristic solutions, the optimal solutions to the 67 problems from Table 2 are 13.2% better on average (min = 0% and max = 158.4%).

The results obtained with the five (or six) most challenging components of each original instance appear in Table 3. Component names are written as "α-β-γ (δ)," where α identifies the instance, β-γ indicates that this is the γ-th component generated by Proposition 20 when applied to the β-th component generated by Proposition 21, and δ is the number of disks. In Table 3, column Base value represents the total length of arcs r that are visible in any feasible solution ($S_r^I = \varnothing$). This value is equal to the length of the border of the region corresponding to the union of all disks and is subtracted from the solution values in the remaining columns. Columns Best feasible and Best UB are the best lower and upper bounds on the optimal value found within the time limit, respectively (optimal solutions appear in bold). Column

Instance *City 156* presented no difficulties, having all of its five largest components solved in less than eight minutes. We found optimal or near-optimal solutions to the first four largest components of *City 538*, with significant improvements in quality with respect to the heuristic solutions. The two largest components of *City 538* turned out to be more challenging, with sizable gaps remaining after five hours of computation. All but one of the largest earthquake and death components were solved to optimality. As was the case with component 538-24-0, the time limit was exhausted during the solution of death-2-0 even before branching started. The largest components obtained from the decomposition of earthquake magnitudes turned out to be the most challenging ones. Note that we do not have valid upper bounds for instances mag-1-0 and mag-7-0 because the time limit was not even enough to solve their first linear relaxation. Overall, we were able to find optimal solutions to 662 of the 671 components derived from our original four instances.

Cutting planes (14) and (18)–(21) were essential in achieving the results in Table 2 and Table 3. With those cuts, the number of search nodes was 54 times smaller on average, with some cases achieving reductions of almost three orders of magnitude. (Five of the 21 hardest components—six overall—would not have been solved to optimality by model M_1 without those cuts.) As a consequence, computation times were also drastically reduced.

Because of its direct relationship to the amount of overlapping between disks, the number of arcs in an instance/component is a better measure of difficulty

Table 2 Average Results with Model M_1 Over Smallest Nontrivial Components of Each Instance

		$\lvert S_k \rvert$			Nodes			Time (in sec.)		
Original instance	Comp. $w/\lvert S_k \rvert > 2$	Min	Avg	Max	Min	Avg	Max	Min	Avg	Max
City 156	11	3	5.3	14	1	20.8	213	0	3.5	38
City 538	20	3	5.4	12	1	11.9	145	0	0.4	5
Deaths	22	3	4.7	10	1	5.8	93	0	0.1	1
Magnitudes	14	3	4.7	10	1	1.8	7	0	0.1	1

Table 3 Results with Model M_1 on Largest Components from Each Original Problem Instance

Component	Base value	Best feasible	Best UB	% gap	% above heur.	Nodes	Time (*s*)
156-18-0 (7)	63.97	**12.91**	**12.91**	0	0	1	0
156-3-2 (8)	39.84	**40.99**	**40.99**	0	8.5	7	0
156-3-0 (14)	66.15	**71.17**	**71.17**	0	7.8	213	39
156-2-0 (26)	167.22	**138.05**	**138.05**	0	3.1	5,949	381
156-2-1 (29)	219.36	**153.85**	**153.85**	0	1.4	117	10
538-47-2 (17)	26.75	**25.27**	**25.27**	0	2.0	2,463	1,259
538-3-0 (26)	34.27	**39.19**	**39.19**	0	15.0	23,589	9,562
538-29-1 (26)	46.48	**36.40**	**36.40**	0	4.3	1,143	1,260
538-1-6 (29)	21.98	43.51	47.05	8.0	9.6	2,399	18,000
538-1-0 (51)	77.37	82.13	107.35	30.7	0.0	22	18,000
538-24-0 (53)	18.98	58.50	186.23	218.3	0.0	1	18,000
death-6-0 (12)	953.08	**60.16**	**60.16**	0	0.0	51	1
death-8-0 (14)	68.05	**39.65**	**39.65**	0	3.1	87	0
death-0-0 (24)	175.78	**145.74**	**145.74**	0	5.7	4,925	199
death-3-0 (24)	441.75	**323.18**	**323.18**	0	1.3	3,919	210
death-2-0 (70)	725.28	964.66	1,652.02	71.2	0.0	1	18,000
mag-5-1 (25)	214.92	**593.74**	**593.74**	0	3.7	965	9,609
mag-6-0 (26)	217.21	579.58	610.99	5.4	5.0	3,385	1,8000
mag-1-1 (39)	417.32	919.28	1,350.23	46.9	0.0	3	18,000
mag-5-0 (81)	601.79	1,741.24	2,317.66	33.1	0.0	1	18,000
mag-1-0 (113)	581.41	2,743.68	—	—	0.0	1	18,000
mag-7-0 (116)	700.37	2,622.46	—	—	0.0	1	18,000

than the number of disks. Model M_1 appears to be capable of handling about 600–700 arcs in five hours of CPU time, which, for our benchmark set, roughly corresponds to instances having between 24 and 26 disks. Table 4 contains more details about the size of our five largest components and how big their ILP formulation is before and after the inclusion of cuts. Because the number of cuts is small in model M_1, we add them at the root node of the search tree.

7.2. Results Obtained with Model M_2

As far as size is concerned, model M_2 is roughly equivalent to model M_1 in terms of the number of variables, but it can have fewer constraints than model M_1, depending on the problem instance. For the instances listed in Table 4, for example, model M_2 can have between 1.6 and 14.7 times fewer rows than model M_1. Moreover, the number of nonzero entries in the constraint matrices of those instances can be between 45 and 288 times smaller in model M_2.

For our set of problem instances, model M_2 turns out to be empirically superior to model M_1. On average, for the easy-to-solve instances (i.e., those summarized in Table 2), model M_2 solves 3.7 times faster (min = 0.95 times, max = 90 times) than model M_1 does and uses 9 times fewer (min = 1 times, max = 213 times) search nodes to reach optimality. However, the most impressive results are obtained on the larger, unsolved components of Table 3, as can be seen in Table 5. Model M_2 managed to solve all remaining instances to optimality, five of them quite easily. Although we allowed it to run for more than five hours in some cases, it is important to note that model M_1 would not have been able to solve those instances even if it had been given the same amount of additional time.

Table 4 Number of Arcs and Size of ILP Formulation for the Five Largest Components Using Model M_1

Component	No. of disks	No. of arcs	No. of cols.	No. of rows before cuts	No. of rows after cuts
538-24-0	53	3,753	6,562	3,026,565	3,035,839
death-2-0	70	1,366	6,266	620,970	624,115
mag-5-0	81	2,059	8,620	914,490	919,623
mag-1-0	113	4,318	17,087	3,733,407	3,744,116
mag-7-0	116	3,759	17,215	2,792,468	2,801,845

While comparing the root node upper bounds for the instances in Table 5 for which both models found a valid upper bound (i.e., excluding mag-1-0 and mag-7-0), we verified that the bounds obtained from model M_2 are always strictly better (10.4% better on average; 17.0% better if we ignore the base value) than those from model M_1. Although our implementation of constraints (4), (8), and (16) varies slightly from their strict definitions (see §7), the previous results, coupled with the superior empirical performance of model M_2 as well as Proposition 15, are supportive of the conjecture stated at the end of §4.

8. Conclusions

We propose two novel ILP formulations to optimize stacking drawings of PSMs with the objective of maximizing the total visible border of its symbols (opaque disks in our case). By studying structural and

Table 5 Results with Model M_2 on Components Not Solved to Optimality by Model M_1

Component	Base value	Best feasible	Best UB	% Gap	% Above Heur.	Nodes	Time (*s*)
538-1-6 (29)	21.98	**44.32**	**44.32**	0	11.7	1	5
538-1-0 (51)	77.37	**90.08**	**90.08**	0	9.7	1	19
538-24-0 (53)	18.98	**65.08**	**65.08**	0	11.2	453	84,308
death-2-0 (70)	725.28	**1,152.13**	**1,152.13**	0	19.4	1	61
mag-6-0 (26)	217.21	**579.58**	**579.58**	0	5.0	1	13
mag-1-1 (39)	417.32	**1,128.52**	**1,128.52**	0	22.8	1	48
mag-5-0 (81)	601.79	**1,914.28**	**1,914.28**	0	9.9	1	2,312
mag-1-0 (113)	581.41	**3,158.82**	**3,158.82**	0	15.1	1	34,306
mag-7-0 (116)	700.37	**2,916.17**	**2,916.17**	0	11.2	1	25,256

polyhedral aspects of these formulations, we devised effective decomposition techniques and new families of facet-defining inequalities that greatly reduce the computational effort required to solve the problem. These improvements enabled us to find the first provably optimal solutions to all of the real-world instances studied in Cabello et al. (2006, 2010). Because PSM instances are still challenging to solve when the number of arcs exceeds 2000 or so, we continue to study the PSM polyhedron in search of new families of cutting planes and/or alternative formulations.

Supplemental Material

Supplemental material to this paper is available at http://dx.doi.org/10.1287/ijoc.2013.0557.

Acknowledgments

The authors thank Marc van Kreveld for providing us with the data files used in Cabello et al. (2006, 2010), and the referees for their helpful comments. This work has been supported by CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico) [Grants 830510/1999-0, 301732/2007-8, 472504/2007-0, 483177/2009-1, 473867/2010-9, and 477692/2012-5], FAPESP (Fundação de Amparo à Pesquisa do Estado de São Paulo) [Grant 07/52015-0], and FAEPEX/Unicamp.

References

Cabello S, Haverkort H, van Kreveld M, Speckmann B (2006) Algorithmic aspects of proportional symbol maps. Azar Y, Erlebach T, eds. *Proc. 14th Eur. Sympos. Algorithms (ESA), Lecture Notes in Computer Science*, Vol. 4168 (Springer-Verlag, Berlin), 720–731.

Cabello S, Haverkort H, van Kreveld M, Speckmann B (2010) Algorithmic aspects of proportional symbol maps. *Algorithmica* 58:543–565.

Dent B (1999) *Cartography—Thematic Map Design*, 5th ed. (McGraw-Hill, Boston).

Fair Isaac Corporation (2009) *Xpress Optimizer Reference Manual*. Accessed June 22, 2013, http://www.fico.com/en/FIResourcesLibrary/Xpress-Optimizer-Refernce-Manual.pdf.

Garey MR, Johnson DS (1979) *Computers and Intractability* (Freeman, San Francisco).

Griffin T (1990) The importance of visual contrast for graduated circles. *Cartography* 19:21–30.

Gupta R, Walrand J, Goldschmidt O (2005) Maximal cliques in unit disk graphs: Polynomial approximation. *Proc. Second Internat. Network Optim. Conf. (INOC), Lisbon, Portugal*. Accessed November 1, 2012, http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.70.1687&rep=rep1&type=pdf.

Kunigami G, de Rezende PJ, de Souza CC, Yunes T (2011) Optimizing the layout of proportional symbol maps. Murgante B, Gervasi O, Iglesias A, Taniar D, Apduhan BO, eds. *Proc. ICCSA, Part III, 11th Internat. Workshop on Computational Geometry and Appl. (CGA), Lecture Notes Computer Science*, Vol. 6784 (Springer-Verlag, Berlin), 1–16.

Müller R (1996) On the partial order polytope of a digraph. *Math. Programming* 73:31–49.

Nemhauser GL, Sigismondi G (1992) A strong cutting plane/branch-and-bound algorithm for node packing. *J. Oper. Res. Soc.* 43:443–457.

NOAA Satellite and Information Service (2005) National geophysical data center. Accessed November 1, 2012, http://www.ngdc.noaa.gov.

Padberg MW (1973) On the facial structure of set packing polyhedra. *Math. Programming* 5:199–215.

Slocum TA, McMaster RB, Kessler FC, Howard HH (2003) *Thematic Cartography and Geographic Visualization*, 2nd ed. (Prentice Hall, Upper Saddle River, NJ).

Wein R, Fogel E, Zukerman B, Halperin D (2007) Advanced programming techniques applied to CGAL's arrangement package. *Computational Geometry* 38:37–63.

West D (2001) *Introduction to Graph Theory*, 2nd ed. (Prentice Hall, Upper Saddle River, NJ).