

Optimal Covering Points and Related Problems

Justin Tzou Brian Wetton *

September 17, 2014

Abstract

We consider the optimal covering of the unit square by N circles. By optimal, we mean the covering that can be done with N circles of minimum radius. Equivalently, we study the problem of the optimal placement of N points such that the maximum over all locations in the square of the distance of the location to the set of points is minimized. We propose a new algorithm that can identify optimal coverings to high precision. Numerical predictions of optimal coverings for $N = 1$ to 16 agree with the best known results in the literature. We use the optimal designs in approximations to two novel, related problems involving the optimal placement of curves.

1 Introduction

We consider in this work the problem of optimally distributing N points inside a domain so that, when the Euclidean distance to the nearest design point is computed for all locations on the domain, the maximum of these distances is minimized. Equivalently, we seek to distribute the N points in such a way that equal-radius circles centred at these points required to cover the entire domain are as small as possible. In this work, we call this the optimal covering problem. We restrict our discussion to the unit square domain. We propose a

*Mathematics Department, University of British Columbia, wetton@math.ubc.ca. Corresponding author.

numerical algorithm to identify optimal designs to high precision. The use of these optimal designs in approximations to two related problems involving the optimal placement of curves is also investigated.

The covering problem arises in statistical applications, and is discussed in [6]. Here, the idea is to identify N experiments that give the best coverage of a two-dimensional parameter range. The problem also arises in the context of operations research in the optimal placement of service centres [4, 1, 17] or wireless transmitters [7]. The literature on the subject is disjoint and the current authors cannot claim to have identified all relevant material. The first time this problem was considered was in [15] in which conjectured optimal coverings of the square for N from one to ten were proposed based on numerical computation. These stand as optimal coverings except for $N = 6$. In this case, a small improvement was found [8]. Proofs for the optimality of some small N cases can be found in [5] with little subsequent theoretical progress. Candidates for optimal coverings of the square up to $N = 30$ based on numerical computation can be found in [11] and for $N = 31$ to 100 in [16]. In this last article, references to other covering problems (sphere surface by circles, triangles by circles, etc.) can be found.

The related problem of circle packing has received much more attention. Early proofs for optimal packing in a square with small numbers of circles can be found in [12]. Numerically computed candidates for optimal solutions can be found in [10] for up to fifty circles in a square. Candidates of optimal packings of up to several thousand circles are kept up to date on a web site [14]. It is known that for the circle packing problem, free points (“rattlers”) can occur that can lie anywhere within a certain region of the domain. This happens in the square for seven circles, for example [12]. For our covering problem in the square, we have not observed an optimal design that allows any point to be moved while still preserving the optimality of the design. This has also been observed by others [15, 8, 11, 16] and we conjecture that for the covering problem in the square, rattlers do not occur. This influenced the numerical optimization strategy used in this article and by others for this covering problem. In the current work, a final numerical step involves a structure in which there are segments of the minimal length that lock the design points into the optimal configuration. For domains other than the square, the covering problem can have rattlers (consider the $N = 2$ covering problem in a circular domain, for example). The methods in the literature and

our own method can be extended to other domain shapes, but would have difficulty with problems with rattlers.

In another related problem, the optimal covering does not minimize the *maximum* distance of any location in the domain to the N points but rather the *average* distance over all locations. This problem can be considered easier than our covering problem since there is more regularity. It is known that as $N \rightarrow \infty$ optimal solutions lie on a hexagonal lattice [9] in the domain interior. This result relies on the fact that edge effects become negligible for large N when the average is taken.

In the current work, we present an algorithm for computing candidate optimal covering designs in the more difficult case where the maximum distance is minimized. The algorithm has two stages. In the first stage, the N points are placed randomly and moved heuristically towards an optimal candidate. Then, a Newton iteration is used to compute optimal locations to near machine precision. Our algorithm is described in more detail below. This iterative refinement idea is different from other approaches in the literature. In [15] the equivalent to our second stage of the algorithm used a mechanical analogy to shrinking, tensioned bars with frictionless pin joints to obtain high accuracy solutions. This was also the second stage strategy used in [11]. In that work, the first stage to identify approximate candidates was not done with a heuristic search but rather an optimization procedure to move the points to reduce the area not covered by the current design. In [16] a similar optimization idea is used without a second stage. For both these competitive algorithms, the iterative second stage we propose could be used to improve final convergence to high accuracy solutions.

The motivating problem for the authors is more difficult. It is to find a simple curve \mathcal{C} in a given simply connected and bounded domain. The curve has given length and given starting and ending points and is such that the maximum distance of the curve to any point in the domain is minimized over the set of all such curves. It is not known whether this problem has well defined minimizers and there may be a high degree of degeneracy in locally optimal curves. It arises in the hydrogen fuel cell application, in the design of reactant gas flow channels [13]. Here, the starting and ending points are the channel inlet and outlet port locations. The channel should be laid out to supply reactant gas evenly to all parts of the fuel cell area, hence the desire to minimize distance to the channel path \mathcal{C} . In the case where

the guarantee of maximum distance to \mathcal{C} is critical but the exact value of curve length is not, sub-optimal solutions of this problem can be found by connecting optimal point coverings to start and end points optimally. The resulting problem is the travelling salesman problem, known to be NP hard but for which many heuristic solution methods are known. A related problem is that of finding a connected network of given length that optimally covers a domain. This removes the requirement of given start and end locations and allows branch points in the network. This problem has been studied in the context of optimal transportation networks [3]. Some numerical results for such optimal networks in the square can be found in [2]. Here, the related problem of optimizing *average* rather than *maximum* distance is studied. For the maximum distance case, sub-optimal solutions that can guarantee a given maximal distance can be constructed by connecting optimal N point coverings with a network of shortest length. This is the Steiner tree problem, also known to be NP-hard but with many effective heuristic algorithms.

In section 2 below, our two-stage algorithm for the optimal covering problem is described. Some results of the method are shown in section 3. The use of the optimal covering points to find approximate solutions to the optimal curve problems is discussed in section 4.

2 Optimal Covering Algorithm

The algorithm has two stages with iterations at each stage. We call the N points P_j to be placed *design points*. At each iteration of the first stage, the point Q in the domain that is farthest away from its nearest design points is found. Then, design points nearest to Q are moved towards it. In the second stage, an optimization problem is solved with Newton iterations that can find design points to machine precision. As with many optimization algorithms, the algorithm may find a local minimum rather than the true optimum. It is therefore intended that a set of solutions be computed from various random starting configurations with enough trials that the optimal configuration is included within this set. The number of such trials needed generally increases with N .

2.1 Computing δ

We define δ as the largest of the distances computed from every point in the domain to the nearest design point(s). We also let the theoretical minimum of δ be called δ^* so that $\delta \geq \delta^*$, with equality holding only when the configuration is optimal. To compute δ for a particular arrangement of design points, we admit the following locations in the domain as candidates for being one of the points Q_i : the four corners, centres of circles situated on the domain boundary passing through two design points, and centres of circles situated strictly inside the domain passing through three design points. The centres of the M circles with radii equal to the largest of all the radii are taken as points $Q_i, i = 1, \dots, M$. Of course, to be an actual candidate, no design points can be strictly inside the circle. It is fairly easy to see that this finite set of possible Q does produce the largest distance δ in the square to the set of design points.

At each step of the algorithm, the value of δ is calculated systematically checking the cases listed above, with the centre(s) of the maximal circles denoted by Q_i .

2.2 Stage 1: Decreasing δ By Moving Design Points

Once the locations of Q_i have been identified, the next step might be to move at least one of the nearest design points toward them by a specified step size h , thereby possibly decreasing δ . However, this method performed poorly, especially for larger N . The reason for this may be that this method is more of a local search, converging only to the “nearest minimum,” a suboptimal configuration for which no design point can be moved locally without increasing δ . In general, the more design points there are, the more local minima.

To promote a wider search and avoid local minima, we instead admit for points Q_i the centres of all circles that have radius within a given tolerance (proportional to the step length h) of δ . We also set the step size relatively large at the start and decrease it gradually. The result is that more design points are moved farther from their original locations. Note that this combination of characteristics does not guarantee a decrease in δ from one step to the next, and so while the algorithm favours downhill moves, uphill transitions are allowed. The following steps summarizes the first stage (the coarse search por-

tion) of the algorithm, starting with a relatively large initial step length h :

1. Compute δ .
2. Make a list of all candidates for δ that are within a predetermined tolerance of δ (the tolerance proportional to the step length).
3. Of this set of circles, move every design point on the boundary of a circle towards the centre. The magnitude of the move is h .
 - if one design point at P_j lies on the boundary of $M > 1$ circles of candidate radii δ_i within tolerance centred at Q_i , $i = 1, 2, \dots, M$, the direction in which to move point j is computed as

$$\vec{v}_j = \sum_{i=1}^M \delta_i (Q_i - P_j),$$

so that directions associated with larger distances are weighted more heavily. For a step size of h , the resulting position of point j would be $P'_j = P_j + h \frac{\vec{v}_j}{|\vec{v}_j|}$.

4. Ensure all design points are still inside the square - if not, decrease the step size by a factor of two and move all points in the same direction. Repeat this step until all points are inside the square and reset the step size to its previous value.
5. Repeat steps 1 through 4 until the value of δ begins to oscillate about a mean value for a predetermined number of times
6. Decrease step size by a geometric factor, and return to step 1. Once the step size has been decreased to a predetermined value, end stage 1.

2.3 Stage 2: High Precision Solutions

The above describes an algorithm that is capable of finding the vicinity of an optimal configuration, but is incapable of locating it to high precision. We proceed on the assumption that, when the configuration is optimal (or even just locally optimal), there are enough maximal locations in the domain for which the nearest design point is the same distance δ (δ^* if optimal) away that the design point positions are fixed. As mentioned in the introduction, this idea relies on the conjecture that no rattlers appear in the optimal covering problem for the square.

This assumption allows for a more algebraic minimization. We seek to make all the links \vec{c} from circle centres Q to any design points P on its circle's edge to be of equal length (the optimal δ^* if the coarse algorithm result is close enough to the optimal configuration). Variables in the optimization are the coordinates of the circle centres Q , the coordinates of the design points P , and δ . Since the number of links is usually different from the number of variables, a straight solve in the manner of equal number of equations and unknowns is not possible in general. We instead minimize, over the space of all variables, the following function for the case of K links:

$$H_a = \sum_{i=1}^K \left(\|P - Q\|_i^2 - \delta^2 \right)^2 + a\delta^2. \quad (1)$$

We make two notes here about Eq. 1. First, the subscript i indexes the i th link, between a design point P and circle centre (maximal distance point) Q . Second, the last term $a\delta^2$ is included to ensure that we are in fact, by minimizing H_a , converging to a configuration for which δ is a minimum. It also serves to prevent a singular Hessian matrix during the Newton iteration for the minimization of H_a (solving of $\nabla H_a = 0$) when there are fewer links than unknowns. We have found that a should start out roughly on the order of the final step size employed in the coarse search algorithm, and then be decreased during the Newton iteration.

A graphical depiction of the method can be seen from the converged $N = 2$ optimal solution shown in figure 1. Here, there are $K = 8$ links, shown as dashed lines. The function H_a is minimized as $a \rightarrow 0$ over 6 unknowns: four for the coordinates of the two design points P and two more for the single coordinates of the two edge maximal distance points Q . The four corners are also maximal distance points Q but are fixed, not variables. During the course of the optimization, the structure of the problem can vary (a corner Q moving along an edge for example). The algorithm tracks these changes automatically.

For this procedure to converge to the optimal solution, it is imperative that the coarse algorithm identifies the “correct links” to set to equal lengths. The lengths of all the correct links must be longer than those of the extraneous links. To achieve this, the final step size h_f used in the coarse search must be small enough. Once that is arranged, the task then is to separate the correct links from the extraneous ones. This is done as follows: we sort the L links (correct

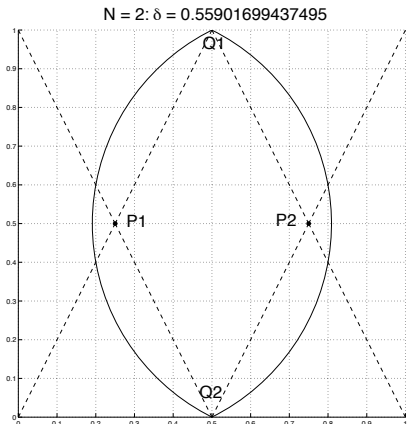


Figure 1: Optimal Configuration for $N = 2$ describing stage 2 of the algorithm.

plus extraneous) according to their lengths so that $\delta_i < \delta_j$ for $i < j$ and compute $R_i = \frac{\delta_{max} - \delta_i}{h_f}$. If the optimal configuration has K correct links, $R_{L-K} - R_{L-K+1}$ is significantly larger than all other differences between adjacent values of R . As part of the algorithm, we have a thresholding technique to identify K . As mentioned above, the two stage method must be run starting from a sample of random initial conditions to be able to find optimal solutions. Past small N , there is no proof of optimality of the results.

The iterations of this second stage can fail to converge or lead to a local but not globally optimal solution. Convergence failure is normally due to an incorrect identification of the links used in the Newton step. This identification becomes increasingly difficult as N increases and the algorithm as a whole is suitable only for moderate N . However, if the links are correctly identified, the Newton procedure is robust and finds potentially optimal solutions quickly and to high accuracy. Thus, we believe our Newton approach is the strongest contribution of this work and can be combined with other stage 1 strategies.

3 Results for various N

We present some of our results for various N in figure 2. Circles centred around each design point have a radius δ , and the dashed lines represent the links in the stage 2 process. Our results for $N = 2-5$ and 7 are proven optimal [5]. The $N = 6$ has a close nearly optimum solution that is often found by the algorithm, first reported in [15]. The optimal case was first seen in [8]. It can be seen that as N increases, the optimal configurations near the boundaries become very subtle.

4 Related Problems

We proceed to the two curve optimization problems discussed in the Introduction. Recall that the motivating problem is: given a closed and bounded domain Ω (the unit square in our computations) find a curve \mathcal{C} of given length L in Ω such that

$$\delta = \max_{\mathbf{x} \in \Omega} \text{dist}(\mathbf{x}, \mathcal{C})$$

is minimized with one of two additional conditions:

A: \mathcal{C} is a simple curve beginning at given point \mathbf{x}_0 (inlet) and ending at given point \mathbf{x}_1 (outlet).

B: \mathcal{C} is connected.

One could also consider the “inverse problem” of finding a curve \mathcal{C} with a given value of δ of smallest length. We provide sub-optimal solutions to these problems based on the $N = 16$ solution shown in figure 2. By sub-optimal, we mean curves that have a value δ below a given tolerance (from application criteria) but may not have the smallest length that can achieve that tolerance. Taking the inlet to be the lower left hand corner and the outlet to be the upper right for condition **A** above, we compute the travelling salesman problem to find the curve of minimum length between the points. This curve is guaranteed to be no more than a distance $\delta = 0.169427051598$ from every point in the domain by the property of the $N = 16$ interior points. However, it is not the simple, continuous curve of the smallest length that has this property. Also shown is the heuristic solution to the connected curve (allowing branches) with this property (condition **B** above). This is found by applying a heuristic Steiner algorithm to

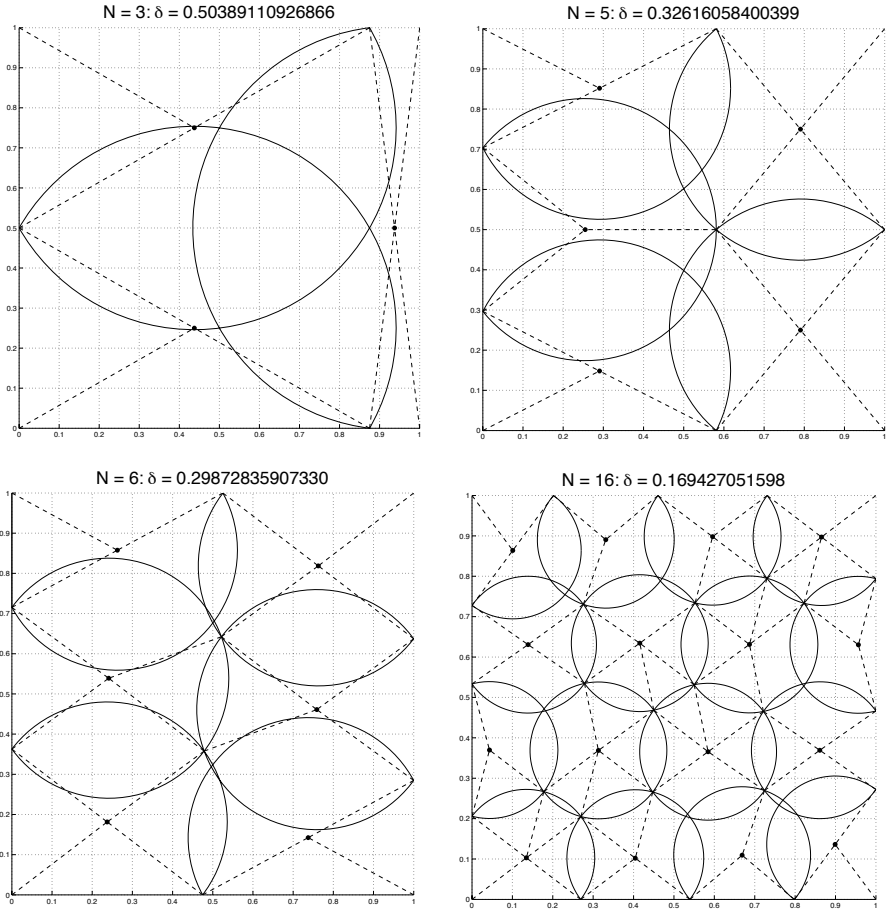


Figure 2: Optimal configurations for $N = 3, 5, 6,$ and 16 points. The configurations for 6 and 16 points are the best reported but not proven optimal.

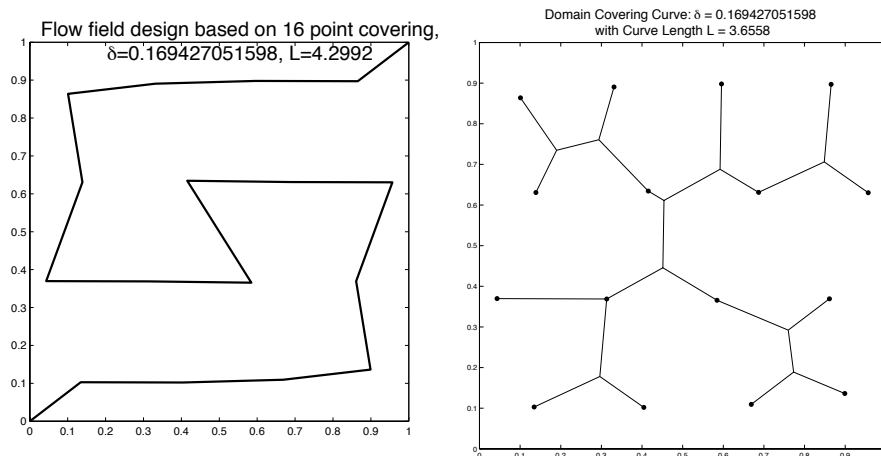


Figure 3: Curves with $\delta = 0.169427051598$ for the simple curve problem (left) with the inlet at lower left and outlet upper right; for the connected network (right). These curves have a guaranteed δ but sub-optimal length.

the configuration for $N = 16$ design points. This curve network is also guaranteed to be no more than a distance $\delta = 0.169427051598$ from every point in the domain. We do not claim that this is the curve with this property that has the shortest length. However, it is known that such optimal curves in the average distance problem can have branch points that must occur at Steiner angles [2, 3].

5 Summary

We present a new algorithm for finding optimal covering points. It has an optimization second stage with a carefully constructed objective function that can find optimal designs to machine precision. Our second stage could be used in other methods that have heuristics that tend to approximate optimal configurations. The optimal designs are used to find approximate solutions to two related optimal curve problems, one of which is of interest to an aspect of hydrogen fuel cell design.

Acknowledgements

The first author would like to thank the NSERC USRA programme for the funding that allowed him to work on this problem. The second author acknowledges NSERC Canada research funding and the contributions to the literature search by another USRA student, Cheng Zhang.

References

- [1] Berman and Drezner, “A new formulation for the conditional p-median and p-venter problem,” *Operations Research Letters* **36**, 481-483 (2008).
- [2] Buttazzo, Oudet and Stepanov, “Optimal Transportation Problems with Free Dirichlet Regions,” conference proceedings of “Variational Methods for Discontinuous Structures,” Cernobio 2001, in *Progress in Nonlinear Differential Equations* **51**, Birkhäuser Verlag, Basel (2002), 41-65.
- [3] Buttazzo, Stepanov, Pratelli, and Solimini, *Optimal Urban Networks via Mass Transportation*, Lecture Notes in Mathematics #1961, Springer, Berlin (2009).
- [4] Drezner, “The p-Centre Problem – Heuristic and Optimal Algorithms,” *Journal of the Operational Research Society* **35**, 741-748 (1984).
- [5] Heppes and Melissen, “Covering a rectangle with equal circles,” *Period. Math. Hung.* **34**, 6581 (1997).
- [6] Johnson, Moore, and Ylvisaker, “Minimax and maximin distance designs,” *Journal of Statistical Planning and Inference* **26**, 131-148 (1990).
- [7] Ke, Liu, and Tasi, “Efficient Algorithm for Constructing Minimum Size Wireless Sensor Networks to Fully Cover Critical Square Grids,” *IEEE Transactions on Wireless Communications* **10**, 1154-1164 (2011).
- [8] Melissen and Schuur, “Improved coverings of a square with six and eight equal circles,” *Electronic Journal of Combinatorics* **3** (#R32), 1-10 (1996).

- [9] Morgan and Bolton, "Hexagonal Economic Regions Solve the Location Problem," *The American Mathematical Monthly* **109**, 165-172 (2002).
- [10] Nurmela and Östergård, "Packing up to 50 Equal Circles in a Square," *Discrete and Computational Geometry* **18**, 111-120 (1997).
- [11] Nurmela and Östergård, "Covering a square with up to 30 circles," Research report, Helsinki University of Technology, Laboratory for Theoretical Computer Science (2000).
- [12] Schaer, "The densest packing of 9 circles in a square," *Canadian Mathematics Bulletin* **8**, 273-277 (1965).
- [13] D. Spornjaka, A. K. Prasada and S. G. Advani, "Experimental investigation of liquid water formation and transport in a transparent single-serpentine PEM fuel cell," *Journal of Power Sources* **170**, 334-344 (2007).
- [14] Specht, <http://hydra.nat.uni-magdeburg.de/packing/csq/csq.html>, accessed November 1, 2012.
- [15] Tarnai and Gaspar, "Covering a square by equal circles," *Elem. Math.* **50**, 167-170 (1995).
- [16] Stoyan and Patsuk, "Covering a compact polygonal set by identical circles," *Comput. Optim. Appl.* **46**, 75-92 (2010).
- [17] Xavier and de Oliviera, "Optimal Covering of Plane Domains by Circles Via Hyperbolic Smoothing," *Journal of Global Optimization* **31**, 493-504 (2005).