The Planar Multiple Obnoxious Facilities Location Problem: A Voronoi Based Heuristic*

Zvi Drezner and Pawel Kalczynski
Steven G. Mihaylo College of Business and Economics
California State University-Fullerton
Fullerton, CA 92834.
e-mail: zdrezner@fullerton.edu; pkalczynski@fullerton.edu

Said Salhi
Centre for Logistics & Heuristic Optimization
Kent Business School
University of Kent, Canterbury CT2 7PE, United Kingdom.
e-mail: S.Salhi@kent.ac.uk

Abstract

Consider the situation where a given number of facilities are to be located in a convex polygon with the objective of maximizing the minimum distance between facilities and a given set of communities with the important additional condition that the facilities have to be farther than a certain distance from one another. This continuous multiple obnoxious facility location problem, which has two variants, is very complex to solve using commercial nonlinear optimizers. We propose a mathematical formulation and a heuristic approach based on Voronoi diagrams and an optimally solved binary linear program.

As there are no nonlinear optimization solvers that guarantee optimality, we compare our results with a popular multi-start approach using interior point, genetic algorithm (GA), and sparse non-linear optimizer (SNOPT) solvers in Matlab. These are state of the art solvers for dealing with constrained non linear problems. Each instance is solved using 100 randomly generated starting solutions and the overall best is then selected. It was found that the proposed heuristic results are much better and were obtained in a fraction of the computer time required by the other methods.

The multiple obnoxious location problem is a perfect example where all-purpose non-linear non-convex solvers perform poorly and hence the best way forward is to design and analyze heuristics that have the power and the flexibility to deal with such a high level of complexity.

Key Words: Location; Obnoxious Facilities; Continuous Location; Voronoi Diagrams; Matlab; Heuristic; Binary Linear Program.

*This research has been supported in part by the Spanish Ministry of Economy and Competitiveness (MTM2015-70260-P), in part financed by the European Regional Development Fund (ERDF).

1 Introduction

Suppose that 100 communities are located in a 100 by 100 miles square. 20 obnoxious (e.g., noisy or polluting) factories or landfills need to be located in the area. These factories are required to be at least $D = 16$ miles from one another to avoid cumulative nuisance to the communities. The objective is to maximize the minimum distance between facilities and communities.

Figure 1: Configuration of 100 Communities and Some Highest hilltops

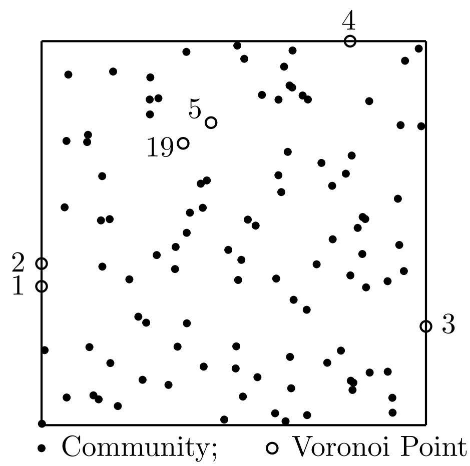
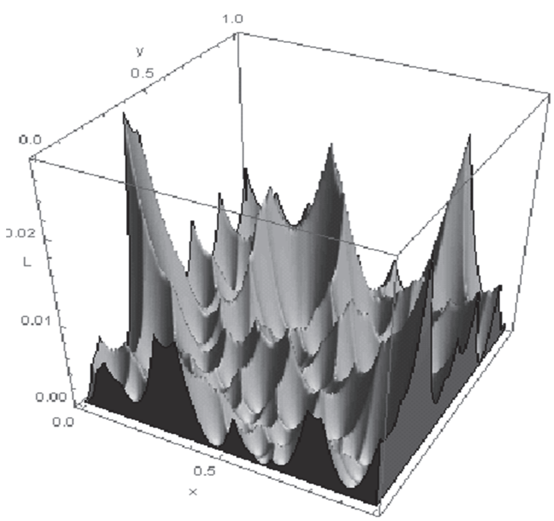


Figure 2: Surface of Distances to the Closest Community



To illustrate the problem consider the randomly generated example problem depicted in Figure 1. The surface of the shortest distance to the communities is depicted in Figure 2. There are 202 hilltops. In Figure 1 the five "tallest" hilltops are marked. If a standard non-linear optimization method is applied from a random starting solution, the process will likely end on hilltops depending on the starting solution. There are 2×10^{27} possible selections of different 20 hilltops. Intuitively, it is preferred to locate facilities on hilltops as long as the minimum distance of 16 miles is maintained. We therefore propose a heuristic that selects the best set of hilltops subject to the distance constraints. Note that if the locations of the facilities are restricted to hilltops, the heuristic solution is optimal. In Section 3.3 we show that the solutions for up to 5 facilities in this example are optimal.

1.1 Literature Review

Obnoxious location problems involve locating one or more facilities as far as possible from a set of communities. Most papers investigate the problem on networks or in discrete space [1, 4, 5, 6, 7, 16, 38]; location in the interior of a network [7, 14]; location on the plane [10, 20, 32, 39]; location on the sphere [12]. Applications may include nuisance generated by the facilities such as airports, pollution generating industrial facilities, prisons, and others affecting residents living in a set of communities. Another type of applications assume that the nuisance is generated by the communities and the facilities should be located at locations with minimum nuisance. For example, the location of schools or hospitals which require a low noise level caused by a set of points or locating a telescope as far as possible from light sources.. In most of these applications the nuisance propagates "by air" and not along nework links making the use of Euclidean distances appropriate.

Such models can be formulated in several ways. The most common way is to maximize the minimum distance between the facilities and the set of communities [4]. Hansen et al. [20] assume that the nuisance caused by communities declines by the square of the distance and suggested to minimize the sum of $\frac{1}{d^2}$ where d is the distance between a community and the facility. Church and Meadows [5] suggested to maximize the sum of distances from communities in a network environment. Colmenar et al. [6] solved the multiple facilities version of this problem.

Drezner and Wesolowsky [14] found a location in the interior of a planar network that maximizes the minimum distance between the facility and the links of the network. Drezner et al. [7] found the best location for a facility in the interior of a planar network minimizing the total nuisance generated by the links of the network.

The single facility problem is to find a location for one facility that maximizes the minimum distance to a set of n communities. The problem is equivalent to finding the center of the largest possible circle that has no communities in its interior. The facility must be located in a bounded region. Otherwise, the solution will be at infinity. Shamos and Hoey [32] showed how to optimally solve the problem in $O(n \log n)$ time using Voronoi Diagrams [30, 34, 37]. The idea of the Voronoi diagram is to partition the plane into polygons such that all the points inside a polygon are closest to one of the communities. The vertices of these polygons are equally distant to at least three communities (and closest to them) or to at least two communities if the Voronoi vertex is on an edge of the feasible region. The vertices of the feasible region are also Voronoi vertices that are at the minimum distance to at least one community. The circle centered at a Voronoi vertex with a radius equal to the distance to the closest community does not have communities in its interior. Therefore, the best location for the facility is on one of these vertices. Finding all the vertices is done in $O(n \log n)$ time and many computer codes are available for finding all the vertices, which are known as "Voronoi points" [29, 33].

The single facility location model suggested by Hansen et al. [20] was optimally solved by the "Big Square Small Square" global optimization method which was introduced in [20]. The problem was also solved by the effective global optimization method known as "Big Triangle Small Triangle" [9]. Another problem that aims to maximize the weighted sum of distances is a special case of minimizing the sum of weighted distances with positive and negative weights [13, 25, 36]. It can be efficiently solved by these global optimization methods, for example, [9].

Most of the papers mentioned above investigated single facility problems. The only paper that ivestigated the planar multi-facility obnoxious facility problem using Euclidean distances is [39]. They found the optimal solution by a branch and bound algorithm which can be applied to relatively small problems. They solved problems with up to five facilities and 120 demand points.

In this paper we heuristically solve two variants of the multiple obnoxious facility problem. The first variant is maximizing the minimum distance between facilities and communities subject to a required minimum distance between facilities. The second variant is maximizing all the distances between facilities and communities and between facilities. The distances between facilities can be multiplied by a factor to reflect a different weight to the two distance types in the objective function. Suppose that a number of communities are located in an area. A required number of obnoxious facilities (for example, noisy factories, landfills emmiting odor) need to be located in the area. The objective is to maximize the minimum distance between the communities and the facilities. The facilities are required to be at least a given distance from one another to avoid cumulative nuisance to the communities. Note that if no separation distance is imposed, the optimal solution is to locate all facilities at the center of the largest circle without any communities. Alternatively, the distance between facilities is required to be at least the minimum distance between communities and facilities multiplied by a given factor.

The aim of the study is three fold:

(i) To heuristically solve the two variants of the multiple obnoxious problem in the plane.

(ii) To effectively incorporate the power of Voronoi vertices with optimally solving a binary linear program in a recursive manner.

(iii) To explore the interior point, GA and SNOPT [17] non-linear optimization solvers in Matlab [19] for comparison purposes.

The rest of the paper is organized as follows. The continuous multiple obnoxious facility location problem is presented and its formulation is provided in the next section. This is followed in Section 3 by our Voronoi-based heuristic using solutions to binary linear programs. The computational results are presented in Section 4. A case study of locating obnoxious facilities in Colorado, U.S.A. is presented and solved in Section 5 and we conclude the paper with a summary of the results.

2 The Multiple Obnoxious Facilities Location Problem

The multiple obnoxious facility location problem is to locate p obnoxious facilities in a convex polygon among a set of communities [11, 21, 38, 39]. Additional restrictions are required, otherwise, the solution would be to locate all facilities at the optimal single facility location. We wish the facilities not to be close to one another because the facilities may affect each other negatively. For example, when locating schools or hospitals as far as possible from nuisance causing communities, these facilities need to be spread out. When locating airports it does not make sense to locate two airports next to one another. In addition, the location of the facilities must be restricted to a finite area, otherwise, the solution would be to locate all facilities at infinity.

Let $A_i = (a_i, b_i)$ for $i = 1, \dots n$ be the locations of communities, and $X_j = (x_j, y_j)$ for $j = 1, \dots, p$ be the unknown locations of the p facilities. We assume that the propagation of the nuisance declines as the Euclidean distances increase.

Two problems are investigated in this paper:

Maximin1: Maximize the minimum squared distance between facilities and communities subject to a given minimum distance D between facilities [2]. The non-linear programming formulation is:

$$\begin{aligned} & \max\{\ L\ \} \\ \text{Subject to:} & \\ & (x_j - a_i)^2 + (y_j - b_i)^2 \geq L \quad \text{for } \ i = 1, \dots n; j = 1, \dots p \\ & (x_i - x_j)^2 + (y_i - y_j)^2 \geq D^2 \quad \text{for } \ 1 \leq i < j \leq p \end{aligned} \tag{1}$$

In addition we need constraints that restrict the facilities' locations to a convex polygon or any region. This formulation has $2p+1$ variables and $np + \frac{p(p-1)}{2}$ constraints in addition to the constraints restricting the locations to a region in the plane such as a square.

Note that it is more convenient to apply squared Euclidean distances in the formulation.

Maximin2: Maximize the minimum of all distances both between the facilities and communities and between facilities [39]. The distances between facilities are equal to the distances between communities and facilities multiplied by a given factor α (the squared distance is multiplied by α^2). Welch et al. [39] allowed for different weights for different facilities. Such a modification can be easily accommodated. This problem is formulated as

$$\begin{aligned} & \max\{\ L\ \} \\ \text{Subject to:} & \\ & (x_j - a_i)^2 + (y_j - b_i)^2 \geq L \quad \text{for } \ i = 1, \dots n; j = 1, \dots p \\ & (x_i - x_j)^2 + (y_i - y_j)^2 \geq \alpha^2 L \quad \text{for } \ 1 \leq i < j \leq p \end{aligned} \tag{2}$$

As in Maximin1, we also need constraints that restrict the facilities' locations to a convex polygon or any region. The size of this formulation is the same as the one given in (1).

The factor α allows flexibilility when facilities are more (or less) obnoxious to each other than to communities. A special case of this problem is using $\alpha = 1$ which entails equal importance given to all distances. In Maximin1 the minimum distance between facilities is imposed rather than being dependent on the minimum distance between facilities and communities.

2.1 Relationship to the p-Dispersion Problem

The multiple obnoxious facility problems are related to the p-dispersion problem [3, 15, 22]. In the p-dispersion problem, a set of potential locations for facilities is given and the objective is to select p points out of the potential locations that maximize the minimum distance between facilities. There are no communities in the p-dispersion model.

The p-dispersion location problem in an area is finding locations for p facilities in the area such that the minimum distance between pairs of facilities is maximized. The formulation is similar to (1) without the first type constraints, because there are no communities in the problem formulation, with the objective of maximizing D. The p-dispersion problem in an area [8, 24, 26, 28, 35] is equivalent to packing p circles in an area. Heuristic solution approaches by solving the non-linear program using all purpose solvers are suggested in [8, 26]. Optimal branch and bound algorithms are proposed in [24, 28, 35]. Most results are for circle packing in a square. The best known solutions for the p-dispersion in a square are given in http://www.packomania.com/ which reports proven optimal solutions for $p \leq 30$ and $p = 36$. Suppose that the optimal solution to the p-dispersion problem is D^*. For these values of p, there cannot be a feasible solution to problem Maximin1 if D exceeds D^*. Furthermore, if D is close to D^*, there are very few feasible solutions and distances to communities become almost irrelevant. This is not in the spirit of obnoxious facilities applications where distances to communities are the focus of the problem. We are therefore interested in D being considerably smaller than D^* for the problem to have practical applicability.

The value of D^* in a unit square is $\sqrt{2}$ for $p = 2$ and D^* declines to about 0.287 for $p = 20$. It is equal to $\frac{1}{q-1}$ for $p = q^2$ for an integer q up to $p = 36$. We therefore selected $D = \frac{1}{\sqrt{2p}}$ and $D = \frac{1}{\sqrt{p}}$ which is below D^*, see Table 1.

3 A Voronoi Based Heuristic Solution Approach

The problems can be heuristically solved by a multi-start approach solving the non-linear non-convex formulations by an optimization software such as those available in Matlab [19]. However, these problems have numerous local maxima and it is difficult to escape such local maxima. For $p = 20$ facilities and $n = 1000$ communitiess there are 4×10^{47} local maxima (some of them are infeasible). Even the smallest problem tested in this paper (locating two facilities among 100 communities) has over 20,000 local maxima. Generating a starting solution close to the "correct"

Table 1: p-dispersion Optimal Solutions

p	D^*	$\frac{1}{\sqrt{2p}}$	Ratio	$\frac{1}{\sqrt{p}}$	Ratio
2	1.414214	0.5000	0.3536	0.7071	0.5000
3	1.035276	0.4082	0.3943	0.5774	0.5577
4	1.000000	0.3536	0.3536	0.5000	0.5000
5	0.707107	0.3162	0.4472	0.4472	0.6325
6	0.600925	0.2887	0.4804	0.4082	0.6794
7	0.535898	0.2673	0.4987	0.3780	0.7053
8	0.517638	0.2500	0.4830	0.3536	0.6830
9	0.500000	0.2357	0.4714	0.3333	0.6667
10	0.421280	0.2236	0.5308	0.3162	0.7506
11	0.398207	0.2132	0.5354	0.3015	0.7572
12	0.388730	0.2041	0.5251	0.2887	0.7426
13	0.366096	0.1961	0.5357	0.2774	0.7576
14	0.348915	0.1890	0.5416	0.2673	0.7660
15	0.341081	0.1826	0.5353	0.2582	0.7570
16	0.333333	0.1768	0.5303	0.2500	0.7500
17	0.306154	0.1715	0.5602	0.2425	0.7922
18	0.300463	0.1667	0.5547	0.2357	0.7845
19	0.289542	0.1622	0.5603	0.2294	0.7923
20	0.286612	0.1581	0.5517	0.2236	0.7802

local maximum is very unlikely.

Maximin1 is equivalent to finding p empty circles such that the distance between any two circles' centers is at least D with the objective of maximizing the radius of the smallest circle. Maximin2 is similar but the value of D depends on the smallest distance between facilities and communities. We propose a heuristic approach that found much better results than solving (1) and (2) directly by a non-linear non-convex available procedure, in a much shorter run time. This approach is based on selecting p points out of the set of V Voronoi points as potential facilities' locations.

It is important to note that even though the heuristic procedures were tested using Euclidean distances, they can be used for any distance measure once a Voronoi diagram is available for that distance measure.

We first define and prepare the following structure:

- All V Voronoi points, intersection points with the sides of the convex polygon and its vertices are generated.
- The distance to the closest community is calculated for each Voronoi point.
- The Voronoi points are sorted by decreasing order of these distances.

- The sorted list of Voronoi points is $\{V_i\}$ for $i = 1, \ldots V$, with a distance d_i between V_i and its closest community, such that $d_1 \geq d_2 \geq \ldots \geq d_V$.
- The distance between Voronoi points i and j is D_{ij}.

It is well known (see for example Okabe et al. [30]) that the number of Voronoi points V is around $2n$.

The idea is to find p locations out of the V Voronoi points so that the distances between the chosen p Voronoi points are feasible, and the minimum distance to communities is maximized. Suppose that the vector of the V Voronoi points is sorted by the distance to the closest community (the peaks in Figure 2). Maximizing the shortest distance is equivalent to finding the p feasible Voronoi points whose p^{th} index in the sorted vector of distances is minimized. Define the optimal p^{th} index as K^* with the optimal objective function d_{K^*}.

Suppose that the first $p \leq K \leq V$ Voronoi points are selected. If $K \geq K^*$ there is a feasible solution to the problem based on these K Voronoi points. On the other hand, if $K < K^*$ no feasible solution exists.

3.1 Solving Maximin1 Heuristically

The first $p \leq K \leq V$ Voronoi points are selected. K binary variables x_i for $i = 1, \ldots, K$ are defined with x_i equals 1 if Voronoi point i is selected and zero otherwise.

The following binary linear program solves the problem for a given K. When a feasible solution for this K exists, the solution is optimal to selecting p out of the V Voronoi points. Otherwise, if there is no feasible solution, the K need to be increased. Define the constants $\Delta_i = d_1 - d_i$. Note that $\Delta_i \geq 0$ because d_1 is the maximum distance.

Formulation BLP

$$\text{Maximize } \{L\}$$

subject to:

$$\sum_{i=1}^{K} x_i = p \tag{3}$$

$$x_i + x_j \leq 1 \text{ when } D_{ij} < D \tag{4}$$

$$L + x_i \Delta_i \leq d_1 \text{ for } i = 1, \ldots, K \tag{5}$$

$$x_i \in \{0, 1\}$$

When Voronoi point i is not selected ($x_i = 0$), the constraint $L + x_i\Delta_i \leq d_1$ is $L \leq d_1$ which is always satisfied. When Voronoi point i is selected ($x_i = 1$), the constraint reduces to $d_i \geq L$ and maximizing L results in the combination of p Voronoi points whose minimum distance to communities is maximized.

We suggest two approaches that employ BLP solutions. In Algorithm 1 we solve the problem using $K = V$. In Algorithm 2 we attempt to shorten the run time by solving a sequence of problems with smaller values of K in order to reduce the number of constraints in the BLP formulation.

Algorithm 1: Solve the BLP problem using $K = V$.

Algorithm 2:

1. Select $K = K_{\min}$.
2. Solve the BLP problem.
3. If there is a solution, stop.
4. If there is no feasible solution, increase K by q and go to Step 2

In our implementation we used $K_{\min} = 2p$ and $q = p$.

3.2 Solving Maximin2 Heuristically

For Maximin2, solving BLP for a given K means replacing D by αd_K in the BLP formulation. We first present and prove the following two properties.

Property 1: *If there is a feasible solution for BLP using $K = K_1$, i.e. using $D = \alpha d_{K_1}$, there will be a feasible solution for every $K \geq K_1$.*

Proof: Since $K \geq K_1$, $\alpha d_K \leq \alpha d_{K_1}$ and the feasible solution using $K = K_1$ is also feasible for $K \geq K_1$. □

Property 2: *If there is no feasible solution for BLP using $K = K_1$, there is no feasible solution for every $K \leq K_1$.*

Proof: If there was a feasible solution for $K \leq K_1$, there would have been a feasible solution for $K = K_1$ by Property 1. □

We conclude, by Properties 1 and 2, that up to a certain value of K there are no feasible solutions and for all greater values of K there is a feasible solution. The solution is obtained by the solution for the smallest possible K, defined as K^*, that has a feasible solution. The objective function, which is the minimum distance between facilities and communities, is d_{K^*} and the distances between facilities are all at least αd_{K^*}.

We create a range $K_{\min} \leq K \leq K_{\max}$ such that for $K = K_{\min}$ there is no feasible solution and for $K = K_{\max}$ there is a feasible solution. The range is decreased every iteration and once $K_{\max} = K_{\min} + 1$, the feasible solution for $K_{\max}$ is the best (or tied for the best) possible set of p locations from the set of V Voronoi points.

In Algorithm 3 we perform a bisection search on the whole range $p \leq K \leq V$. In algorithm 4 we attempt to reduce the run times by narrowing the range for the bisection search and avoid solving unnecessarily BLP problems with large values of K. The power and usefullness of incorporating neighborhood reduction in the search is shown to be a promising way forward in heuristic search design in general [31].

Algorithm 3:

1. Set $K_{\min} = p - 1$ and $K_{\max} = V$. We select $K_{\min} = p - 1$ because it is possible that for $K = p$ there is a feasible solution.
2. Set $K = \frac{1}{2}(K_{\min} + K_{\max})$ rounded down and solve the BLP problem replacing D by αd_K.
3. If there is no feasible solution, set $K_{\min} = K$ and go to Step 4. Otherwise,
   (a) Save this solution and set $K_{\max} = K$.
   (b) Find $\overline{K}$, the largest index among the Voronoi points in the solution.
   (c) If $\overline{K} \leq K_{\min}$ go to Step 4.
   (d) If $\overline{K} = K$ go to Step 4. If $\overline{K} < K$ solve the BLP problem for $K = \overline{K}$ replacing D by $\alpha d_{\overline{K}}$ and go to Step 3.
4. If $K_{\max} - K_{\min} > 1$ go to Step 2. Otherwise, choose the solution for $K = K_{\max}$ and stop.

Note that Step 3c is valid because there is no feasible solution for $\overline{K}$ by Property 2

Algorithm 4: Tightening Scheme for Initial $K_{\min}$ and $K_{\max}$

1. Select $K = 2p$.

2. Solve the BLP problem for this K replacing D by αd_K.

3. If there is no feasible solution, increase K by p and go to Step 2.

4. Otherwise, set $K_{\max} = K$, $K_{\min} = K - p$ ($K_{\min} = p - 1$ if $K = 2p$), and apply Algorithm 3 from Step 2 with these values of $K_{\min}$ and $K_{\max}$.

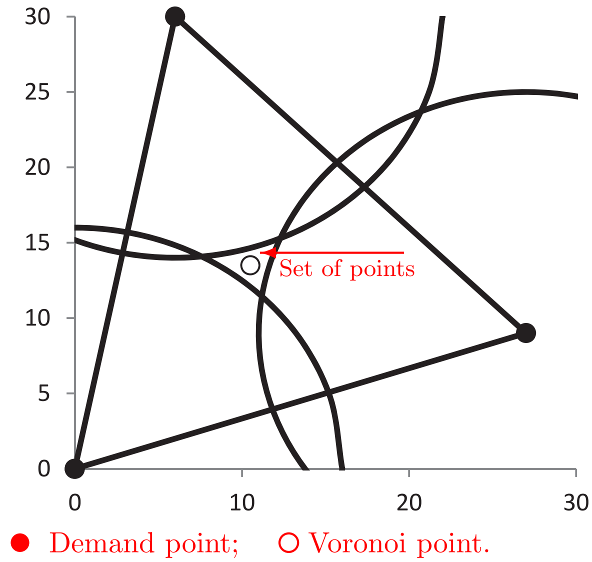
3.3 Properties of the Heuristic Solution

This approach is a heuristic because the optimal facilities' locations are not necessarily on hilltops. When there are no distance constraints, then the solution is to locate all the facilities at the top of the tallest hilltop. When D is very small, the solution is a cluster of p facilities near the tallest hilltop as long as the minimum distance to communities does not decrease by much. When D is moderately increased, then it is not possible to locate more than one facility in a disk centered at any hilltop because the hill is not large enough. Even if the hill is large enough, the facilities must be located near the bottom of the hill and the minimum distance to communities (the minimum height of the two facilities' locations) is likely to be too small. It is important to remember that the solution is the height of the lowest facility and facilities that are located at higher locations can be moved without affecting the value of the objective function.

It is possible, however, that the optimal solution is not at hilltops. Suppose that two tall hilltops, are slightly closer than D to one another. The heuristic procedure cannot select both hilltops (it may select only one of them). However, it may be possible to move the facilities located at these two hilltops in opposite directions thus attaining a distance of D between them (and not violating the distance constraints with other facilities) while reducing slightly the minimum distance between the translated two facilities and their closest community (sliding downhill but not by much). This may result in a better solution if the minimum distance is still above the heuristic objective function. Such a scenario is possible but not likely. Even if it occurs, the heuristic solution will not deteriorate much. For example, the heuristic solution for locating $p = 20$ facilities includes the 53^{rd} highest hilltop. If the minimum distance is relaxed and we select among the 46 highest hilltops, the objective on hilltops is 2% higher, but the 16 miles distance constraints are violated. If the facilities are moved from the hilltops to accommodate the 16 miles distance requirement, the objective function cannot improve and may well be worse than the heuristic objective function.

To illustrate this point we considered a hilltop based on three demand points, see Figure 3. The demand points are located at (0, 0), (6, 30), and (27, 9). The Voronoi point inside the triangle is

Figure 3: Points Near the Top of a Hill Exceeding a Height of 16



located at (10.5, 13.5) which is at distance $\sqrt{292.5} = 17.103$ from all three demand points. The area near the Voronoi point for which the height on the hill is greater than 16 is depicted in Figure 3. Suppose that the distance to another Voronoi point is slightly lower than D and the heuristic solution is less than 16. It may be possible to move these two points downhill slightly farther from one another to achieve a distance of D between them and if the points remain in the interior of the area, the value of the objective function may exceed 16. Such a solution is not on Voronoi points. Since the "hills" are steep, see Figure 2, the area of exceeding a lower value is usually small as is observed in Figure 3.

Consider the only instance that SNOPT found a better solution than the Voronoi heuristic. The location of $p = 4$ facilities among $n = 100$ demand points using $D = 0.5$ reported in Table xx. In Table 3 the distances between the first five Voronoi points are reported. When $D \leq 0.419$ is used, the solution is points 1 (or 2) and points 3,4,5. The objective function for this selection is 0.150887. However, this solution violates the $D \geq 0.5$ constraints. A better solution was found by SNOPT by moving slightly Voronoi points 4 and 5 thus obtaining a distance of 0.5 between them. The objective function was reduced by 17% to 0.124590, but this solution is still better than using Voronoi point #19 (see Figure 1), which satisfies the $D \geq 0.5$ constraints, with the objective of 0.114609. This observation may suggest ways to attempt and improve the heuristic approach. Such options are discussed in the conclusions section as ideas for future research.

It may also be possible that a large area contains no communities and there is only one Voronoi point in the area while two or more facilities that are far enough from one another may be located there. Note that our algorithm is suited for $p << n$ which is the case in most practical applications. For example, if $p > V$, there must be facilities located at points which are not Voronoi points. However such instances require a very small value of D which is not practical.

Note that in an optimal solution to the original problem a facility cannot be located inside disks (on hills) whose d_i (height) is smaller than d_{K^*}. If a starting solution for the non-linear optimization procedure has facilities on such hills, standard non-linear optimization software will not be able to escape to another hill due to the extreme non-convexity of the surface (see for example Figure 2). In other words, the procedure has to cross deep "valleys", which it is not designed to do, and eventually will result in an inferior value of the objective function.

Table 2: The First Five Voronoi Points

i	x	y	d_i
1	0	0.361453	0.166317
2	0	0.420781	0.158368
3	1	0.257239	0.154282
4	0.802745	1	0.151738
5	0.440903	0.787825	0.150887

Table 3: The Values of D_{ij}

	1	2	3	4	5
1	0.000	0.059	1.005	1.026	0.613
2	0.059	0.000	1.013	0.990	0.574
3	1.005	1.013	0.000	0.769	0.771
4	1.026	0.990	0.769	0.000	0.419
5	0.613	0.574	0.771	0.419	0.000

See for example the randomly generated problem with $n = 100$ communities used in the computational experiments. The 100 communities in a square are depicted in Figure 1. We also show in the same figure the top five Voronoi points which are also given in Table 2. The distances D_{ij} between the five Voronoi points are given in Table 3. The values of D used for the $p = 2, 3, 4$ instances are $D = \frac{1}{\sqrt{2p}} = 0.5, 0.408, 0.354$, respectively.

In Figure 2, the five highest Voronoi points are visible. Compare it also with Figure 1. The first two Voronoi points are on the left wall very close to one another, the third one is on the right (not so clearly visible), the fourth one is in the far right, and the fifth one is close to the middle.

Since the distance D_{12} is small, the heuristic solutions for $2 \leq p \leq 4$ include either Voronoi point 1 or Voronoi point 2 but not both. Voronoi point 3 is added to the $p = 2$ heuristic solution, Voronoi points 3 and 4 are added for $p = 3$ and Voronoi points 3,4,5 for the $p = 4$ heuristic solution. The heurisic objectives for these p's are indeed d_3, d_4, and d_5 and K^* are 3,4, and 5 (see Table 4). These solutions are optimal for the original problems because we cannot "separate" Voronoi points 1 and 2 far enough so that the distance between them will not be less than D. For example, for the $p = 2$ instance the points in the square that are at least a distance d_3 from all communities, which is the heuristic objective, are Voronoi point 3 and small areas surrounding Voronoi points 1 and 2 (intersection of the exteriors of circles centered at communities with a radius d_3). See also Figure 3 for an illustration. All other points in the square are closer than d_3 to at least one community. If there is a solution with a distance greater than d_3, the area where facilities can be located does not include Voronoi point 3 but the facilities must be located in the interior of the small areas around Voronoi points 1 and 2 and none of these points are at least a distance $D = \frac{1}{2}$ from one another.

Theorem 1: *The heuristic solution based on Voronoi points is a local maximum.*

Proof: Consider the heuristic solution that includes the Voronoi point V_{K^*} which determines its objective value d_{K^*}. There may be several Voronoi points tied for this distance. By the construction of the Voronoi points, an infinitesimal change in the location of V_{K^*} cannot increase its minimal distance to the closest community. Infinitesimal changes in other Voronoi points which are part of the heuristic solution, V_K for $K \leq K^*$, cannot improve the value of the objective function as well. Therefore, any combination of such infinitesimal changes cannot improve the value of the objective function even if some Voronoi points are at exactly a distance D from one another. □

By Theorem 1 it is clear that if the heuristic solution is used as a starting solution to a non-linear optimization software, such software cannot improve it. For illustration purposes, empirical experiments were conducted using Matlab showing that the heuristic solutions could not be improved when they were used as a starting solution, supporting Theorem 1.

4 Computational Experiments

All experiments were run on a virtual server with 16 vCPUs and 128 GB of vRAM. Algorithms 1-4 were implemented with the OPL and run on IBM's CPLEX Optimization Studio 12.4 environment. We used the default CPLEX MIP solver settings for all four algorithms.

The non-convex quadratically-constrained (QCP) versions of Maximin1 and Maximin2 problems were implemented in Matlab R2016b and solved using the interior-point method and SNOPT [17] starting from 100 random solutions. The GA method provided similar but poorer results and therefore results using GA are not reported. Unlike in the case of CPLEX, the default settings of QCP interior point solver resulted in poor quality solutions and long processing times, so the following changes were made: (i) analytical gradients and Hessians were specified for the objective function and all non-linear constraints, (ii) scaling was applied to the objective function and all constraints and (iii) the maximum number of function evaluations was increased to 50000. The first two changes significantly improved the quality of the solutions and solver's efficiency (run time) and the last one prevented the solver from exiting prematurely.

We experimented with $n = 100$ and 1000 with $p = 2, 3, \ldots, 20$ for each problem for a total of 76 instances, each solved by two algorithms and QCP for comparison purposes. We generate random locations for communities in a square (for details see the Appendix) that can be easily replicated for future comparisons with other methods. The interior point and SNOPT solvers were applied in a multi-start approach repeating the process from 100 random starting solutions and the best result is reported. We also experimented with $n = 100$ instances and 1000 starting solutions. The results were only slightly better but run times were about 10 times longer and thus these results are not reported. For Maximin1 we applied $D = \frac{1}{\sqrt{2p}}$ and for Maximin2 we applied $\alpha = 2$. The value of V is 202 for $n = 100$ and 2002 for $n = 1000$.

In tables 4-7 we report:

1. for the heuristic algorithms: the value of the objective function which is the minimum distance between facilities and communities,

2. for Maximin1: the number of pairs of Voronoi points for which $D_{ij} < D$ (constraints of type (4)),

3. for the heuristic algorithms: the value of K at the optimal solution,

4. for Maximin2: the number of BLP applications,

5. the clock time in seconds,

6. for QCP we also report the value of the objective function and the percentage of the QCP objective below the heuristic objective for both the interior point solver and SNOPT.

Table 4: Results for Maximin1 $n = 100$ Instances Using $D = \frac{1}{\sqrt{2p}}$

	Heuristic		Alg. 1		Alg. 2		Interior Point			SNOPT		
p	Objective	†	K	Time (sec.)	K	Time (sec.)	Objective	Time (sec.)	% below Heuristic	Objective	Time (sec.)	% below Heuristic
2	0.154282	9,593	3	1.60	3	0.57	0.111489	23.51	27.7%	0.154282	2.66	0.0%
3	0.151738	6,957	4	1.63	4	0.90	0.110094	23.04	27.4%	0.133824	2.52	11.8%
4	0.150887	5,495	5	2.15	5	1.15	0.108818	26.00	27.9%	0.102189	3.50	32.3%
5	0.111488	4,558	21	2.38	21	2.18	0.092668	31.07	16.9%	0.102189	4.77	8.3%
6	0.111488	3,884	21	2.54	21	2.12	0.092668	37.96	16.9%	0.102189	7.15	8.3%
7	0.110668	3,432	22	2.58	22	2.23	0.095394	56.61	13.8%	0.094258	17.27	14.8%
8	0.108818	3,007	25	2.10	25	2.26	0.081280	48.48	25.3%	0.095395	20.20	12.3%
9	0.106636	2,720	26	2.87	26	2.18	0.081276	54.47	23.8%	0.092658	23.33	13.1%
10	0.102189	2,477	32	2.34	32	1.82	0.081271	67.72	20.5%	0.081767	79.07	20.0%
11	0.101100	2,271	36	2.07	36	2.27	0.081278	68.32	19.6%	0.081281	76.49	19.6%
12	0.100538	2,071	39	2.29	39	2.21	0.075754	76.60	24.7%	0.075618	100.21	24.8%
13	0.100538	1,913	39	1.65	39	1.43	0.081280	84.58	19.2%	0.081407	135.11	19.0%
14	0.096482	1,789	46	2.63	46	2.13	0.055635	93.30	42.3%	0.078265	206.78	18.9%
15	0.096482	1,688	46	3.85	46	2.16	0.058284	112.99	39.6%	0.071777	255.66	25.6%
16	0.096482	1,596	46	2.60	46	1.10	0.027046	135.21	72.0%	0.067402	489.99	30.1%
17	0.096482	1,515	46	2.63	46	1.37	0.050588	208.13	47.6%	0.072201	553.38	25.2%
18	0.095394	1,436	49	3.51	49	1.99	0.027046	199.49	71.6%	0.063215	502.35	33.7%
19	0.094537	1,365	51	2.87	51	1.62	0.027045	209.96	71.4%	0.066939	446.24	29.2%
20	0.094259	1,303	53	3.07	53	1.63	0.050265	212.04	46.7%	0.059051	581.34	37.4%

† Constraints of Type (4) for $K = V$

4.1 Maximin1 Results

In Table 4 we report results for $n = 100$ and in Table 5 for $n = 1000$ by Algorithms 1 and 2, interior point and SNOPT for $2 \leq p \leq 20$. In two cases the value of K at the heuristic solution is not the same for both algorithms. This is due to ties between some values of d_i resulting in different sorted vectors of d_i. The value of the objective function is the same by applying both algorithms.

We note that for $n = 100$, the interior point $p = 2$ best objective is equal to d_{21}, $p = 3$ objective is equal to d_{22}, and the $p = 14, 16, 18, 19, 20$ objectives are equal to d_{197} almost at the bottom of the

Table 5: Results for Maximin1 $n = 1000$ Instances Using $D = \frac{1}{\sqrt{2p}}$

p	Heuristic Objective	†	Alg. 1		Alg. 2		Interior Point			SNOPT		
			K	Time (sec.)	K	Time (sec.)	Objective	Time (sec.)	% below Heuristic	Objective	Time (sec.)	% below Heuristic
2	0.060413	978,875	5	176.37	5	23.44	0.032961	166.01	45.4%	0.043710	20.78	27.6%
3	0.048334	718,783	12	71.32	12	22.21	0.032971	250.88	31.8%	0.040716	43.25	15.8%
4	0.048334	569,500	12	89.24	12	21.24	0.031792	466.65	34.2%	0.038923	79.44	19.5%
5	0.048099	473,078	14	49.04	14	23.74	0.027037	880.24	43.8%	0.035011	924.34	27.2%
6	0.048099	405,490	14	47.72	14	21.15	0.021501	1143.16	55.3%	0.039285	1793.50	18.3%
7	0.044364	355,455	29	74.95	29	23.30	0.013620	1477.08	69.3%	0.030422	1693.56	31.4%
8	0.044364	316,529	29	39.43	29	21.66	0.024378	1869.33	45.0%	0.027604	2736.92	37.8%
9	0.044324	284,873	30	37.19	30	21.77	0.010598	2621.13	76.1%	0.027575	5933.92	37.8%
10	0.043385	259,320	36	50.68	36	20.99	0.019381	2881.38	55.3%	0.026487	4878.93	38.9%
11	0.041560	238,634	44	32.04	43	23.22	0.014727	3535.14	64.6%	0.024715	5576.01	40.5%
12	0.041552	220,422	45	60.26	45	22.40	0.012497	4425.06	69.9%	0.023707	5660.69	42.9%
13	0.041193	204,874	49	29.53	49	22.26	0.010614	5601.43	74.2%	0.025258	8140.24	38.7%
14	0.041193	191,432	49	33.50	49	22.32	0.010578	6416.49	74.3%	0.020032	19105.39	51.4%
15	0.039729	179,592	69	29.95	70	22.02	0.010614	9233.43	73.3%	0.023445	1608.35	41.0%
16	0.039664	169,334	72	62.55	72	22.77	0.003796	10958.50	90.4%	0.020939	5546.08	47.2%
17	0.039647	160,236	74	26.84	74	22.80	0.003797	13542.01	90.4%	0.016033	8355.69	59.6%
18	0.039664	152,157	72	38.14	72	24.07	0.010576	15234.95	73.3%	0.021174	11513.74	46.6%
19	0.039647	144,796	73	30.62	73	22.77	0.003797	16269.15	90.4%	0.020433	11902.36	48.5%
20	0.040239	138,098	65	51.09	65	24.30	0.003796	17831.12	90.6%	0.016277	18924.10	59.5%

† Constraints of Type (4) for $K = V$

list of 202 Voronoi points. One of the facilities is located at the top right corner (1,1) (see Figure 1). The best QCP solutions have at least one facility at the top of a low height hill (see Figure 2). Note that it is enough that one facility is “stuck” in a “bad” region. All other facilities’ locations become irrelevant to the value of the objective function.

We suspect that the interior points and SNOPT are not designed to effectively solve such extreme non-convex problems. To get good solutions one must be “lucky” when selecting the starting solution. In fact, when the heuristic solution was used as a starting solution for QCP, the result remained the same. This is expected in view of Theorem 1. Run times by QCP are much longer. Note that Algorithms 1 and 2 have no random component and replicating them will yield the same solution which is optimal for the BLP.

For $n = 100$ run times are comparable for Algorithms 1 and 2. For $n = 1000$, run times required by Algorithm 2 are stable and do not vary much for different values of p. On the other hand, run times by Algorithm 1 decrease as p increases. Since D decreases as p increases, the number of

Table 6: Results for Maximin2 $n = 100$ Instances

p	Heuristic Objective	Algorithm 3 K	Algorithm 3 BLP Runs	Algorithm 3 Time (sec.)	Algorithm 4 K	Algorithm 4 BLP Runs	Algorithm 4 Time (sec.)	QCP (Interior Point) Objective	QCP (Interior Point) Time (sec.)	QCP (Interior Point) % below Heuristic
2	0.154282	3	3	1.66	3	4	1.94	0.111489	17.84	27.7%
3	0.151738	4	3	2.36	4	4	1.79	0.108817	19.58	28.3%
4	0.150887	5	3	2.13	5	5	1.75	0.095811	26.21	36.5%
5	0.128668	14	12	9.01	14	9	5.68	0.092664	31.57	28.0%
6	0.111488	21	6	4.14	21	9	3.94	0.091351	35.57	18.1%
7	0.110668	22	4	3.72	22	10	5.42	0.083075	54.85	24.9%
8	0.108818	25	6	4.49	25	10	5.24	0.081279	48.44	25.3%
9	0.106636	26	5	5.07	26	7	4.35	0.081279	51.75	23.8%
10	0.102189	32	9	6.57	32	10	6.30	0.079098	71.50	22.6%
11	0.101100	36	8	9.42	36	11	6.56	0.076545	81.25	24.3%
12	0.100538	39	10	10.98	39	11	9.22	0.079096	77.60	21.3%
13	0.098631	43	13	11.56	43	10	6.77	0.078138	88.80	20.8%
14	0.096482	46	10	10.28	46	11	7.37	0.066097	98.15	31.5%
15	0.095394	49	10	10.17	49	11	6.89	0.066088	107.45	30.7%
16	0.094259	53	11	10.17	53	14	10.07	0.068306	117.49	27.5%
17	0.094259	53	9	9.08	53	11	7.38	0.056740	138.84	39.8%
18	0.094258	54	8	9.97	54	9	8.41	0.056742	146.93	39.8%
19	0.094012	55	8	8.99	55	9	9.95	0.056744	155.11	39.6%
20	0.093847	56	9	10.89	56	10	9.39	0.048573	173.89	48.2%

constraints of type (4) decreases as p increases leading to shorter run times. OPL using CPLEX was able to solve such problems with almost a million constraints and two thousand variables in a short run time. Both algorithms required very short run times and Algorithm 2 is clearly preferred for small values of p.

The quality of the heuristic solutions is much better than those of the QCP. For $n = 100$, the interior point objectives were below the heuristic objectives by 13%-72% and the SNOPT objective was the same for $p = 2$ but was up to 37% below the heuristic solution for larger values of p. For $n = 1000$, the interior point solutions were 32%-90% below the heuristic objectives and the SNOPT solutions were 16%-59% below the heuristic solutions. In some cases the heuristic objective was more than 10 times better! Run times required by Matlab are much longer. The largest problem was solved heuristically in 24 seconds while it required about five hours by the interior point and SNOPT.

Table 7: Results for Maximin2 $n = 1000$ Instances

p	Heuristic Objective	Algorithm 3			Algorithm 4			QCP (Interior Point)		
		K	BLP Runs	Time (sec.)	K	BLP Runs	Time (sec.)	Objective	Time (sec.)	% below Heuristic
2	0.060413	5	6	15.61	5	6	8.56	0.032961	184.28	45.4%
3	0.052838	7	5	16.92	7	6	10.04	0.032971	320.24	37.6%
4	0.050160	9	5	18.45	9	8	11.74	0.031792	527.52	36.6%
5	0.048652	10	4	16.21	10	6	10.21	0.027037	897.87	44.4%
6	0.048334	12	5	17.86	12	6	11.74	0.021501	1321.51	55.5%
7	0.048099	14	7	17.62	14	6	10.42	0.013620	1656.54	71.7%
8	0.047801	16	7	19.99	16	7	12.00	0.024378	2348.02	49.0%
9	0.044977	24	12	22.96	24	6	11.98	0.010598	2799.99	76.4%
10	0.044977	25	6	18.24	25	8	11.61	0.019381	3371.73	56.9%
11	0.044364	29	12	29.76	29	6	18.15	0.014727	3913.02	66.8%
12	0.044324	30	10	22.64	30	8	11.86	0.012497	4564.89	71.8%
13	0.044324	31	6	27.82	31	9	12.64	0.010614	5467.57	76.1%
14	0.043973	32	10	22.10	32	9	12.36	0.010578	6450.50	75.9%
15	0.043710	33	10	22.40	33	10	13.10	0.010614	7515.70	75.7%
16	0.043710	34	6	18.91	34	10	13.21	0.003796	8120.26	91.3%
17	0.043487	35	10	22.94	35	10	13.52	0.003797	9550.26	91.3%
18	0.043385	36	10	21.63	36	8	13.03	0.010576	10591.68	75.6%
19	0.042742	40	12	24.63	40	10	13.63	0.003797	11294.25	91.1%
20	0.041560	43	12	22.56	43	10	13.68	0.003796	11861.28	90.9%

4.2 Maximin2 Results

In Table 6 we report results for $n = 100$ and in Table 7 for $n = 1000$ by Algorithms 3 and 4 and interior point for $2 \leq p \leq 20$. The SNOPT solver failed to find even one feasible solution in 100 runs for 20 of the 38 instances. We also solved the $n = 100$ instances 10,000 times. There were more feasible solutions and the results are slightly better (much worse than the heuristic results) and run times are 100 times longer. We thereofore do not report the SNOPT results for the Maximin2 instances. The performance, except SNOPT, and conclusions are very similar to those obtained for Maximin1.

Both algorithms are very efficient for $n = 100$ and performed about equally well. For $n = 1000$, run times required by Algorithm 4 are generally lower than those required by Algorithm 3.

The quality of the heuristic solutions is much better than those obtained by the interior point solver. For $n = 100$, the interior point objectives were below the heuristic objectives by 18%-48%. For $n = 1000$, they were 37%-91% below the heuristic objectives. Run times of the QCP are much

Table 8: Results for Maximin1 $n = 100$ Instances Using $D = \frac{1}{\sqrt{p}}$

p	Heuristic	Alg. 1		Alg. 2		Interior Point			SNOPT		
	Obje-ctive	K	Time (sec.)	K	Time (sec.)	Obje-ctive	Time (sec.)	% below Heuristic	Obje-ctive	Time (sec.)	% below Heuristic
2	0.154282	3	0.59	3	0.18	0.111486	3.81	27.7%	0.154283	3.07	0.0%
3	0.151738	4	0.57	4	0.16	0.108816	3.49	28.3%	0.150887	3.24	0.6%
4	0.114609	19	0.59	19	0.20	0.108818	5.30	5.1%	0.124591	5.13	-8.7%
5	0.111488	21	0.51	21	0.29	0.092665	7.54	16.9%	0.110669	7.39	0.7%
6	0.110668	22	0.47	22	0.19	0.081278	9.72	26.6%	0.095301	9.26	13.9%
7	0.108818	25	0.52	25	0.20	0.073011	23.61	32.9%	0.101620	22.99	6.6%
8	0.102189	32	0.48	32	0.24	0.073038	54.59	28.5%	0.095395	52.58	6.6%
9	0.102189	32	2.18	32	0.32	0.069585	61.80	31.9%	0.086117	59.57	15.7%
10	0.095394	49	0.55	49	0.24	0.055803	81.38	41.5%	0.093217	91.84	2.3%
11	0.095394	49	0.50	49	0.33	0.065434	74.35	31.4%	0.079606	72.08	16.6%
12	0.095169	50	0.50	50	0.35	0.027046	79.35	71.6%	0.079318	77.03	16.7%
13	0.094401	52	0.75	52	0.24	0.059649	151.55	36.8%	0.073015	147.33	22.7%
14	0.081280	78	0.72	78	0.40	0.035561	134.87	56.2%	0.072998	131.88	10.2%
15	0.081407	77	2.78	77	0.45	0.065433	302.97	19.6%	0.070529	294.69	13.4%
16	0.081280	78	0.61	78	0.29	0.055881	237.83	31.2%	0.067402	232.79	17.1%
17	0.075380	91	0.52	91	0.52	0.027045	490.82	64.1%	0.073015	483.18	3.1%
18	0.073080	103	0.53	103	0.52	0.027046	221.06	63.0%	0.069189	245.86	5.3%
19	0.075380	91	0.56	91	0.37	0.070528	268.33	6.4%	0.056615	299.86	24.9%
20	0.076831	88	0.57	88	0.38	0.027046	481.49	64.8%	0.058138	521.96	24.3%

longer in some cases by a factor of almost one thousand.

5 Case Study: Locating Obnoxious Facilities in Colorado

There are 271 municipalities in Colorado and we wish to build p obnoxious facilities such as pollution generating industrial facilities to be as far as possible from these municipalities. The problems were solved by the Voronoi based heuristic algorithms (that need to be solved only once) as well as by Matlab, using interior point method and using SNOPT, reporting the best solution obtained from 100 randomly generated starting solutions.

The locations for $2 \le p \le 20$ by Maximin1 requiring $D = 80$ miles between facilities and Maximin2 with $\alpha = 2$ are depicted in Tables 11 and 12. The results clearly show that the Voronoi based heuristic performed much better than the Matlab procedures on these 38 instances. The best value of the objective function obtained by the Matlab procedures was betwwn 6% and 57% lower than the results obtained by the Voronoi heuristic. Run times by the Voronoi based heuristic are

Table 9: Results for Maximin1 $n = 1000$ Instances Using $D = \frac{1}{\sqrt{p}}$

p	Heuristic Objective	Alg. 1		Alg. 2		Interior Point			SNOPT		
		K	Time (sec.)	K	Time (sec.)	Objective	Time (sec.)	% below Heuristic	Objective	Time (sec.)	% below Heuristic
2	0.060413	5	120.93	5	13.65	0.032968	151.78	45.4%	0.060414	28.85	0.0%
3	0.048099	14	100.05	14	13.89	0.032959	342.97	31.5%	0.038076	70.84	20.8%
4	0.048099	14	90.64	14	13.74	0.029457	523.79	38.8%	0.036250	146.11	24.6%
5	0.044364	29	82.76	29	15.14	0.027045	848.14	39.0%	0.032971	355.05	25.7%
6	0.044364	29	86.52	29	13.44	0.019380	1274.65	56.3%	0.031137	373.78	29.8%
7	0.043710	34	74.77	34	13.73	0.027044	1753.83	38.1%	0.029046	2959.65	33.5%
8	0.041560	43	79.85	43	14.26	0.026407	2518.62	36.5%	0.028746	1745.14	30.8%
9	0.039729	69	84.82	69	14.23	0.019382	2975.67	51.2%	0.027386	1653.56	31.1%
10	0.039729	69	76.05	69	14.15	0.019381	3750.88	51.2%	0.024644	1304.80	38.0%
11	0.038075	115	66.48	115	15.24	0.010609	5058.97	72.1%	0.023472	2954.71	38.4%
12	0.039123	87	68.38	87	14.63	0.009643	7469.20	75.4%	0.024104	3310.78	38.4%
13	0.038075	115	75.59	115	14.91	0.009863	11551.76	74.1%	0.023842	8746.06	37.4%
14	0.038075	115	84.43	115	14.29	0.010608	13240.33	72.1%	0.020034	10845.53	47.4%
15	0.037049	133	67.23	133	14.59	0.010615	19198.95	71.3%	0.020288	6800.51	45.2%
16	0.037049	133	64.99	133	15.31	0.009875	21075.77	73.3%	0.021321	3716.66	42.5%
17	0.035941	147	110.77	147	14.47	0.003795	30142.08	89.4%	0.020564	7650.78	42.8%
18	0.034271	203	82.50	203	17.11	0.003794	34586.89	88.9%	0.016961	4403.85	50.5%
19	0.035288	167	122.84	167	15.06	0.003796	40564.65	89.2%	0.020070	6593.26	43.1%
20	0.033267	222	132.45	222	17.47	0.010612	45396.36	68.1%	0.020019	8934.54	39.8%

more than 1,000 times faster for large values of p. Interior point performed better than SNOPT.

The solution for locating 20 obnoxious facilities by the maximin1 model is depicted in Figure 4. The heuristic minimum distance between facilities and communities is about 16.5 miles (see Table 11). Interior point's best solution is about 15.6 miles while SNOPT's is about 10.8 miles.

6 Conclusions

We formulated and solved two multiple obnoxious facilities problems. A given number of facilities are to be located in a convex polygon with the objective of maximizing the minimum distance between facilities and a given set of communities. The facilities has to be farther than a certain distance from one another. The proposed heuristic solution approaches are based on generating the Voronoi points of Voronoi diagrams [30, 34]. A binary linear program (BLP) was constructed and the solution approaches applied this BLP iteratively. Run times are very short producing excellent results.

Table 10: Upper Bounds For the Heuristic Results

	$n = 100$			$n = 1,000$		
p	Heuristic	U.B.	% above	Heuristic	U.B.	% above
2	0.154282	0.158368	2.6%	0.060413	0.065301	8.1%
3	0.151738	0.154282	1.7%	0.048099	0.064855	34.8%
4	0.150887	0.151738	0.6%	0.048099	0.063223	31.4%
5	0.111488	0.150887	35.3%	0.044364	0.060413	36.2%
6	0.111488	0.150845	35.3%	0.044364	0.055291	24.6%
7	0.110668	0.148404	34.1%	0.043710	0.052838	20.9%
8	0.108818	0.135640	24.6%	0.041560	0.050315	21.1%
9	0.106636	0.134780	26.4%	0.039729	0.050160	26.3%
10	0.102189	0.134754	31.9%	0.039729	0.048652	22.5%
11	0.101100	0.133824	32.4%	0.038075	0.048627	27.7%
12	0.100538	0.133587	32.9%	0.039123	0.048334	23.5%
13	0.100538	0.132914	32.2%	0.038075	0.048158	26.5%
14	0.096482	0.128668	33.4%	0.038075	0.048099	26.3%
15	0.096482	0.126415	31.0%	0.037049	0.047881	29.2%
16	0.096482	0.124170	28.7%	0.037049	0.047801	29.0%
17	0.096482	0.124036	28.6%	0.035941	0.047774	32.9%
18	0.095394	0.117843	23.5%	0.034271	0.047660	39.1%
19	0.094537	0.114609	21.2%	0.035288	0.046943	33.0%
20	0.094259	0.113482	20.4%	0.033267	0.046500	39.8%

For comparison purposes we solved the problem by a multi-start approach applying the non-convex quadratically-constrained (QCP) method in Matlab based on Matlab's default interior point and SNOPT solvers. The best results obtained by Matlab are worse by at least 13% than the heuristic results. In some cases the heuristic results are better by a factor greater than 10. This means that the minimum distance between communities and facilities in the heuristic solution is more than ten times greater than the minimum distance in the best solution found by Matlab! For example, suppose that 1000 communities are located in a 100 by 100 miles square in locations corresponding to our test problem. 20 noisy factories need to be located in the area. These factories are required to be at least 16 miles from one another to avoid cumulative nuisance to the communities. By Matlab using the interior point method the minimum distance between a community and a factory is 0.38 miles (see Table 5). SNOPT found a solution of 1.6 miles. By our heuristic result each community is at least 4 miles away from any factory. When the distances between factories are required to be at least twice the minimum distance to the communities (see Table 7), the minimum distance by the interior point method is the same 0.38 miles, SNOPT failed

Table 11: Results for Colorado Municipalities: Maximin1 Objective

	Heuristic	Alg. 1		Alg. 2		Interior Point			SNOPT		
p	Objective	K	Time (sec.)	K	Time (sec.)	Objective	Time (sec.)	% below Heuristic	Objective	Time (sec.)	% below Heuristic
2	39.694395	4	3.53	4	1.06	36.511967	41.96	8.0%	33.728213	10.90	15.0%
3	38.333642	6	3.26	6	1.28	35.181856	54.72	8.2%	33.728213	17.07	12.0%
4	37.388116	8	3.59	8	1.11	29.416538	83.51	21.3%	33.728213	57.46	9.8%
5	35.771728	16	3.25	16	1.34	26.652446	103.13	25.5%	33.728213	492.77	5.7%
6	35.181855	18	3.68	18	1.12	26.600475	143.14	24.4%	31.924244	1169.13	9.3%
7	33.728213	20	3.25	20	1.34	26.600475	171.81	21.1%	25.416189	1957.45	24.6%
8	30.134736	32	3.65	32	1.17	22.520097	210.84	25.3%	22.520097	2322.08	25.3%
9	29.742869	33	3.37	33	1.36	22.520097	290.29	24.3%	20.820193	3175.30	30.0%
10	29.528622	36	3.63	36	1.14	20.126060	335.87	31.8%	22.132283	2938.46	25.0%
11	29.416537	37	3.31	37	1.33	18.718547	452.46	36.4%	19.287761	2987.33	34.4%
12	29.057271	39	3.65	39	1.12	22.113030	561.33	23.9%	17.393139	3999.68	40.1%
13	28.629458	42	3.35	42	1.33	19.079799	689.59	33.4%	16.998044	3119.47	40.6%
14	28.265609	47	3.62	47	1.16	19.575039	873.61	30.7%	15.655950	5069.37	44.6%
15	24.835288	88	3.40	88	1.54	17.559717	1128.02	29.3%	15.255855	5696.77	38.6%
16	22.132283	118	3.79	118	1.64	16.309250	1343.76	26.3%	13.932383	6009.23	37.0%
17	20.949743	133	3.43	133	1.84	18.718547	1736.98	10.7%	13.870493	3498.17	33.8%
18	19.418680	154	3.71	154	1.97	13.261295	2133.28	31.7%	14.485326	4566.82	25.4%
19	19.145613	159	3.46	159	2.23	16.608329	2663.39	13.3%	11.821557	5510.11	38.3%
20	16.535647	215	3.82	215	2.70	15.565481	3009.58	5.9%	10.829746	4801.80	34.5%

to find a feasible solution, and our heuristic found a solution with a minimum distance of 4.16 miles. Run times required by Matlab employing the interior point method or SNOPT solvers are much longer. The largest problem was solved heuristically in 24 seconds while it required about five hours by Matlab. We do not expect to get much better results by using other non-linear non-convex solvers because there are so many local maxima (4×10^{47} local maxima, some infeasible, for the largest tested problem) and the result depends on the initial random solution because it is unlikely to move from one local maximum to another (see Figure 2).

We also solved a case study of locating obnoxious facilities in Colorado among 271 municipalities. The Voronoi heuristic performed much better than Matlab for this case study as well. By inspecting Figures 1 and 4, it seems that solutions tend to be close to the periphery of the convex polygon. It is possible, for example in the Colorado case study, that communities outside the state may be affected and should be considered in the model. In such cases the Voronoi points should be created considering also points outside the convex polygon but restricted to the convex polygon. This can be accomplished by creating a Voronoi diagram based on all points, selecting as Voronoi points the

Table 12: Results for Colorado Municipalities: Maximin2 Objective

p	Heuristic Objective	Alg. 3 K	Alg. 3 Time (sec.)	Alg. 4 K	Alg. 4 Time (sec.)	Interior Point Objective	Interior Point Time (sec.)	Interior Point % below Heuristic	SNOPT Objective	SNOPT Time (sec.)	SNOPT % below Heuristic
2	39.694395	4	1.85	4	1.08	36.511967	44.00	8.0%	33.728213	10.95	15.0%
3	38.333642	6	1.74	6	1.02	35.181856	56.67	8.2%	33.728213	18.43	12.0%
4	37.388116	8	2.08	8	1.04	29.416538	84.32	21.3%	33.728213	62.07	9.8%
5	37.203216	9	2.53	9	1.11	26.652446	108.30	28.4%	33.728213	539.84	9.3%
6	35.181855	18	2.03	18	1.16	26.600475	150.18	24.4%	31.924244	1262.25	9.3%
7	33.728213	20	2.08	20	1.17	26.600475	181.87	21.1%	25.416189	2092.06	24.6%
8	33.119803	22	3.71	22	1.34	22.520097	219.15	32.0%	22.520097	2504.32	32.0%
9	32.110293	24	1.93	24	1.10	22.520097	298.95	29.9%	20.820193	3378.49	35.2%
10	30.134736	32	2.02	32	1.43	20.126060	355.58	33.2%	22.132283	3118.91	26.6%
11	29.416537	37	2.01	37	1.40	18.718547	473.52	36.4%	19.287761	3167.01	34.4%
12	29.057271	39	2.00	39	1.47	22.113030	588.50	23.9%	17.393139	4429.27	40.1%
13	28.902359	40	1.96	40	1.39	19.079799	729.14	34.0%	16.998044	3297.04	41.2%
14	28.629458	42	2.01	42	1.21	19.575039	939.98	31.6%	15.655950	4924.29	45.3%
15	28.359140	46	2.18	46	1.37	17.559717	1134.59	38.1%	15.255855	5216.74	46.2%
16	28.265609	47	2.12	47	1.27	16.309250	1373.46	42.3%	13.932383	5698.57	50.7%
17	27.093737	65	2.40	65	2.04	18.718547	1786.32	30.9%	13.870493	3111.48	48.8%
18	26.388220	72	2.59	72	2.17	13.261295	2193.20	49.7%	14.485326	3921.14	45.1%
19	26.086154	73	2.71	73	2.07	16.608329	2723.78	36.3%	11.821557	4800.02	54.7%
20	25.266566	80	2.37	80	1.65	15.565481	3072.09	38.4%	10.829746	4197.52	57.1%

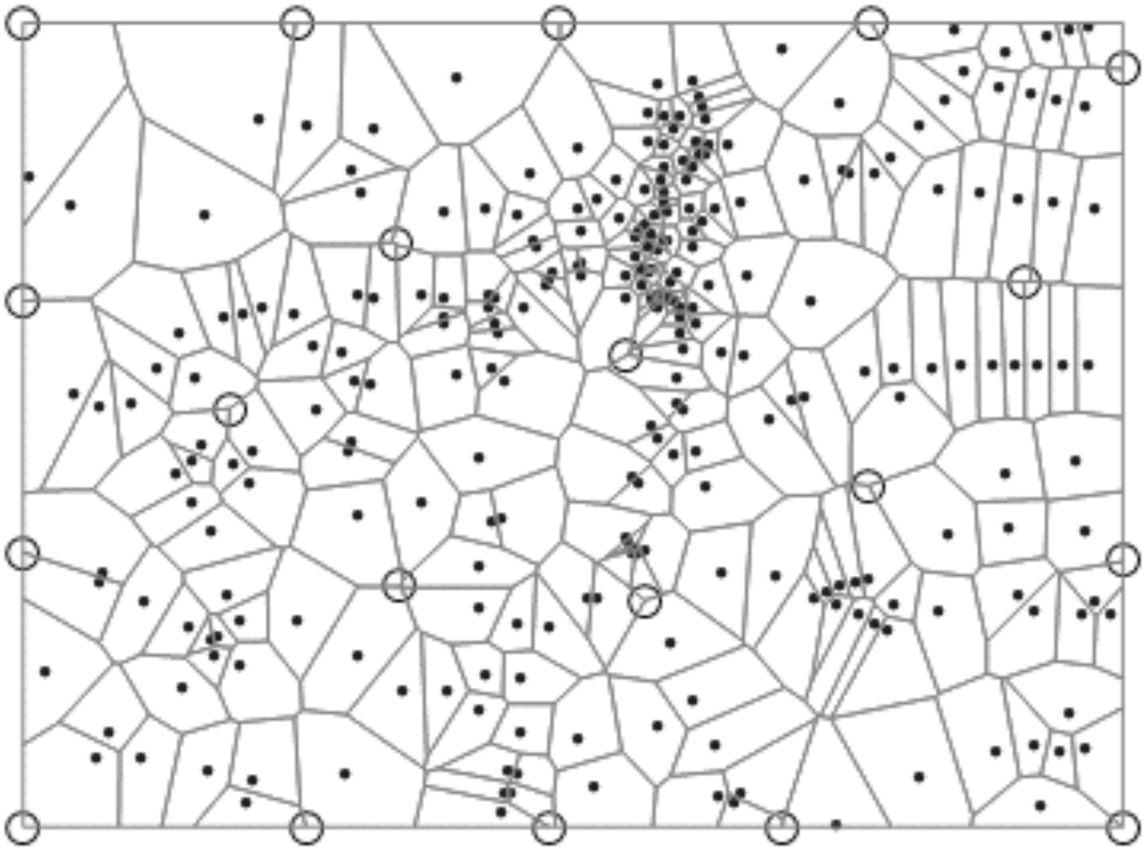
Voronoi points in the convex polygon and the intersection points between the Voronoi edges and the boundary of the convex polygon.

The problem can also be defined in a cube in three dimensions or on the globe. The heuristic approach requires three-dimensional Voronoi vertices [18] or spherical Voronoi vertices [27]. Non linear optimization procedures such as QCP in Matlab can be implemented in a multi-start approach but from the experience based on the results presented in this paper we do not expect that high quality solutions will be found this way.

6.1 Suggestions for Future Research

The discussion in Section 3.3 suggests other solution algorithms based on the Voronoi heuristic. There are a few possible approaches. For example, require a lower value of D and apply the Voronoi heuristic. Presumably, some constraints for the original value of D are violated. Apply an optimization procedure from this solution subject to the original D constraints. Some solution points may slide a bit from hilltops and a better solution may possibly be found. Constructing,

Figure 4: Locating 20 Obnoxious Facilities in Colorado



analyzing, and testing such approaches will constitute a full fledged new paper.

Appendix: Generating Random Configurations

We follow the idea presented in [23] for generating random numbers. We generate a sequence of integer numbers in the open range (0, 100,000). A starting seed r_1, which is the first number in the sequence, is selected. The sequence is generated by the following rule for $k \geq 1$:

- Set $\theta = 12219 r_k$.
- Set $r_{k+1} = \theta - \lfloor \frac{\theta}{100000} \rfloor \times 100000$, i.e., r_{k+1} is the remainder of dividing θ by 100000. It is also the last five digits of θ.

For the x coordinates we used $r_1 = 97$ and for the y-coordinates we used $r_1 = 367$. The first 100 points in a square (we divide the coordinates by 100000 so the points are in a unit square) are depicted in Figure 1.

These sequences return to the first point after 5000 generations. Note that even though there are 99,999 numbers between 1 and 99,999, even numbers and numbers divisible by 5 are not obtained in the sequences. We could get longer sequences if 100,000 or 99,999 were prime numbers. The sequences suggested in [23] exploit the fact that $2^{31} - 1$ is a prime number. Note that the number 12,219 can be replaced by many other numbers.

References

[1] Batta, R. and Chiu, S. (1988). Optimal obnoxious paths on a network: Transportation of hazardous materials. *Operations Research*, 36:84–92.

[2] Berman, O. and Huang, R. (2008). The minimum weighted covering location problem with distance constraints. *Computers & Operations Research*, 35:356–372.

[3] Chandra, B. and Halldórsson, M. M. (2001). Approximation algorithms for dispersion problems. *Journal of algorithms*, 38:438–465.

[4] Church, R. L. and Garfinkel, R. S. (1978). Locating an obnoxious facility on a network. *Transportation Science*, 12:107–118.

[5] Church, R. L. and Meadows, B. (1979). Location modelling using maximum service distance criteria. *Geographical Analysis*, 11:358–373.

[6] Colmenar, J. M., Greistorfer, P., Martí, R., and Duarte, A. (2016). Advanced greedy randomized adaptive search procedure for the obnoxious p-median problem. *European Journal of Operational Research*, 252:432–442.

[7] Drezner, T., Drezner, Z., and Scott, C. H. (2009). Location of a facility minimizing nuisance to or from a planar network. *Computers & Operations Research*, 36:135–148.

[8] Drezner, Z. and Erkut, E. (1995). Solving the continuous p-dispersion problem using non-linear programming. *Journal of the Operational Research Society*, 46:516–520.

[9] Drezner, Z. and Suzuki, A. (2004). The big triangle small triangle method for the solution of non-convex facility location problems. *Operations Research*, 52:128–135.

[10] Drezner, Z. and Wesolowsky, G. (1983a). Location of an obnoxious facility with rectangular distances. *Journal of Regional Science*, 23:241–248.

[11] Drezner, Z. and Wesolowsky, G. (1985). Location of multiple obnoxious facilities. *Transportation Science*, 19:193–202.

[12] Drezner, Z. and Wesolowsky, G. O. (1983b). Minimax and maximin facility location problems on a sphere. *Naval Research Logistics Quarterly*, 30:305–312.

[13] Drezner, Z. and Wesolowsky, G. O. (1991). The Weber problem on the plane with some negative weights. *INFOR*, 29:87–99.

[14] Drezner, Z. and Wesolowsky, G. O. (1996). Obnoxious facility location in the interior of a planar network. *Journal of Regional Science*, 35:675–688.

[15] Erkut, E. (1990). The discrete p-dispersion problem. *European Journal of Operational Research*, 46:48–60.

[16] Erkut, E. and Neuman, S. (1989). Analytical models for locating undesirable facilities. *European Journal of Operational Research*, 40:275–291.

[17] Gill, P. E., Murray, W., and Saunders, M. A. (2005). SNOPT: An SQP algorithm for large-scale constrained optimization. *SIAM Review*, 47:99–131.

[18] Golin, M. J. and Na, H.-S. (2003). On the average complexity of 3d-Voronoi diagrams of random points on convex polytopes. *Computational Geometry*, 25:197–231.

[19] Hanselman, D. and Littlefield, B. C. (1997). *Mastering MATLAB 5: A comprehensive tutorial and reference*. Prentice Hall PTR.

[20] Hansen, P., Peeters, D., and Thisse, J.-F. (1981). On the location of an obnoxious facility. *Sistemi Urbani*, 3:299–317.

[21] Karkazis, J. and Karagiorgis, P. (1987). The general problem of locating obnoxious facilities in the plane. In *Proceedings of the llth IFORS Conference, Buenos Aires*, pages 703–717.

[22] Kuby, M. (1987). Programming models for facility dispersion: the p-dispersion and maxisum dispersion problems. *Geographical Analysis*, 19(4):315–329.

[23] Law, A. M. and Kelton, W. D. (1991). *Simulation modeling and analysis*. McGraw-Hill, New York, second edition.

[24] Locatelli, M. and Raber, U. (2002). Packing equal circles in a square: a deterministic global optimization approach. *Discrete Applied Mathematics*, 122:139–166.

[25] Maranas, C. D. and Floudas, C. A. (1993). A global optimization method for Weber's problem with attraction and repulsion. In Hager, W. W., Hearn, D. W., and Pardalos, P. M., editors, *Large Scale Optimization: State of the Art*, pages 259–293. Kluwer, Dordrecht.

[26] Maranas, C. D., Floudas, C. A., and Pardalos, P. M. (1995). New results in the packing of equal circles in a square. *Discrete Mathematics*, 142:287–293.

[27] Na, H.-S., Lee, C.-N., and Cheong, O. (2002). Voronoi diagrams on the sphere. *Computational Geometry*, 23:183–194.

[28] Nurmela, K. J. and Oestergard, P. (1999). More optimal packings of equal circles in a square. *Discrete & Computational Geometry*, 22:439–457.

[29] Ohya, T., Iri, M., and Murota, K. (1984). Improvements of the incremental method of the Voronoi diagram with computational comparison of various algorithms. *Journal of the Operations Research Society of Japan*, 27:306–337.

[30] Okabe, A., Boots, B., Sugihara, K., and Chiu, S. N. (2000). *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*. Wiley Series in Probability and Statistics. John Wiley.

[31] Salhi, S. (2017). *Heuristic Search: The Emerging Science of Problem Solving*. Springer.

[32] Shamos, M. and Hoey, D. (1975). Closest-point problems. *Proceedings 16th Annual Symposium on the Foundations of Computer Science*, pages 151–162.

[33] Sugihara, K. and Iri, M. (1994). A robust topology-oriented incremental algorithm for Voronoi diagram. *International Journal of Computational Geometry and Applications*, 4:179–228.

[34] Suzuki, A. and Okabe, A. (1995). Using Voronoi diagrams. In Drezner, Z., editor, *Facility Location: A Survey of Applications and Methods*, pages 103–118. Springer, New York.

[35] Szabo, P. G., Markot, M., Csendes, T., and Specht, E. (2007). *New Approaches to Circle Packing in a Square: With Program Codes*. Springer, New York.

[36] Tellier, L. N. and Polanski, B. (1989). The 1-median problem: Frequency and different solution types and extension to repulsive forces and dynamic processes. *Journal of Regional Science*, 29:387–405.

[37] Voronoï, G. (1908). Nouvelles applications des paramètres continus à la théorie des formes quadratiques. deuxième mémoire. recherches sur les parallélloèdres primitifs. *Journal für die reine und angewandte Mathematik*, 134:198–287.

[38] Welch, S. and Salhi, S. (1997). The p obnoxious facility network location problem with facility interaction. *European Journal of Operational Research*, 102:302–319.

[39] Welch, S. B., Salhi, S., and Drezner, Z. (2006). The multifacility maximin planar location problem with facility interaction. *IMA Journal of Management Mathematics*, 17:397–412.