Reverse facility location problems

Sergio Cabello[*] J. Miguel Díaz-Báñez[†] Stefan Langerman[‡] Carlos Seara[§] Inma Ventura[¶]

1 Introduction and problem statement

In the *Nearest Neighbor problem* (NN), the objects in the database that are nearer to a given query object than any other objects in the database have to be found. In the conceptually inverse problem, *Reverse Nearest Neighbor problem* (RNN), objects that have the query object as their nearest neighbor have to be found. Reverse Nearest Neighbors queries have emerged as an important class of queries for spatial and other types of databases. The concept has been introduced by Korn et al. [10, 11], where a large number of applications in marketing and decision support system are given. See [17] for a recent survey on the current state-of-art and open geometric problems in another application area.

The RNN problem itself has several variants, namely, the monochromatic, bichromatic, static or dynamic versions. In the monochromatic case, all points have the same color. In the bichromatic case, the point set consists of both red and blue points and the problem is that given a query point of one color, one needs to find the points of the other color for which the query point is a bichromatic nearest neighbor. In the static version of the problem, the distances between the points in the set remain unchanged whereas in the dynamic problem they may change. Some related work includes [4, 12, 14, 16].

This paper considers the static bichromatic variant in which the data points are of two categories. In particular, we define *RNN facility location problems*, in which some points are designated as facilities and others as customers. In this setting, a *reverse nearest neighbor query* asks for the set of customers affected by the opening of a new facility at some point, assuming all customers go to their nearest facility. We study optimization problems that arise when considering various optimization criteria: maximizing the number of potential customers for the new facility (MAXCOV criterion); minimizing the maximum distance to the associated clients (MINMAX criterion); and maximizing the minimum distance to the associated clients (MAXMIN criterion). The MAXCOV and MINMAX criteria deal with the location of an *attractive* facility, while the MAXMIN criterion seeks the best location for a new *obnoxious* facility. Finally, observe that the MAXCOV criterion can also be seen as a greedy step in a discrete version of the Voronoi game [2].

In the sequel, unless otherwise specified, we use the L_2 metric and $d(p, q)$ denotes the Euclidean distance between the points p and q. Let $S = \{p_1, \ldots, p_N\}$ be a set of points in the plane. Given a point b in the plane, the *reverse nearest neighbor set* of b is defined as $RNN(b) = \{p_i \in S : d(p_i, b) \leq d(p_i, p_j), \forall p_j \in S \setminus \{p_i\}\}$. For the bichromatic case, assume we have a nonempty set $R = \{r_1, \ldots, r_n\}$ of n red points (clients) and a nonempty set $B = \{b_1, \ldots, b_m\}$ of m blue points (facilities), with $n \geq m \geq 2$. Given a new query blue point $b \notin B$, the *bichromatic reverse nearest neighbor set* is defined as $BRNN(b) = \{r_i \in R : d(r_i, b) \leq d(r_i, b_j), \forall b_j \in B\}$. Notice that the monochromatic and bichromatic settings differ on, for example, the size of the output of the queries, as it is stated in the following result.

Lemma 1 [15] *For any query point, the set $RNN(b)$ has at most 6 points, but the size of $BRNN(b)$ may be arbitrarily large.*

Notice that for any blue point $b \notin B$, it holds that $0 \leq |BRNN(b)| \leq n$. Notice also that if $r_i \in BRNN(b)$, then (by definition) the open disk centered at r_i and radius $d(r_i, b)$ is empty of blue points. We formalize the optimization problems as follows.

The MAXCOV problem *Given a set $S = R \cup B$, compute the value MAXCOV(S)= $\max\{|BRNN(b)| : b \in \mathbb{R}^2 \setminus B\}$, i.e., compute the maximum number of points that $BRNN(b)$ may have for a new point $b \notin B$, and find a witness placement b_0 such that $|BRNN(b_0)|$ =MAXCOV(S).*

In the MAXCOV problem, we are interested in computing the locus $\mathcal{L}_S$ of all points b satisfying $|BRNN(b)| = \text{MAXCOV}(S)$. More generally, for any positive integer k, we are interested in the level set $L(k) = \{b \in \mathbb{R}^2 : |BRNN(b)| \geq k\}$. Observe that

[*]Dept. of Mathematics, Institute for Mathematics, Physics and Mechanics, partially supported by the European Community Sixth Framework Programme under a Marie Curie Intra-European Fellowship, cabello@imfm.uni-lj.si

[†]Dept. de Matemática Aplicada II, Universidad de Sevilla, partially supported by project BFM2003-04062, dbanez@us.es

[‡]Chercheur qualifié du FNRS, Dept. d'Informatique, Université Libre de Bruxelles, stefan.langerman@ulb.ac.be

[§]Dept. de Matemàtica Aplicada II, Universitat Politècnica de Catalunya, partially supported by projects MCYT-FEDER-BFM2003-00368, Gen-Cat-2001SGR00224 and MCYT HU2002-0010, carlos.seara@upc.edu

[¶]Dept. de Matemáticas, Universidad de Huelva, partially supported by project BFM2003-04062, iventura@us.es

$L(MAXCOV(S)) = \mathcal{L}_S$, and $L(1) = \{b \in \mathbb{R}^2 : BRNN(b) \neq \emptyset\}$.

The MINMAX problem *Given $S = R \cup B$ and a region $X \subseteq L(1)$, compute the value $MINMAX(S) = \min_{b \in X} \max\{d(b,x) : x \in BRNN(b)\}$, and find a witness placement $b_0 \in X$ such that $\max\{d(b_0,x) : x \in BRNN(b_0)\} = MINMAX(S)$.*

The MAXMIN problem *Given $S = R \cup B$ and a region $X \subseteq L(1)$, compute the value $MAXMIN(S) = \max_{b \in X} \min\{d(b,x) : x \in BRNN(b)\}$, and find a witness placement $b_0 \in X$ such that $\min\{d(b_0,x) : x \in BRNN(b_0)\} = MAXMIN(S)$.*

Notice that for both the MINMAX and MAXMIN problems we add the additional constraint that the new point b has to be placed in a given region X with $X \subseteq L(1)$, as otherwise we could always place b such that $BRNN(b) = \emptyset$. We assume that X can be described using $O(n)$ pieces of constant complexity. Typically, we would consider X to be a level set $L(k)$ for some value k. Although for some values k, the level set $L(k)$ may have quadratic complexity in n, we will see that we can handle this type of sets within the same asymptotic bounds. Note that the MAXCOV and MAXMIN/MINMAX criteria are of completely different nature: the MAXCOV criterion maximize the number of points in a set, which is a discrete measure; the MAXMIN/MINMAX criteria optimize a distance, which a is continuous measure.

2 The MAXCOV problem

In this section we provide exact and approximate algorithms for the MAXCOV problem, as well as a hardness result for the exact problem.

For every red point $r_i \in R$, we denote by $b(r_i)$ the nearest blue point. Let R_i be the *red disk* with radius $d(r_i, b(r_i))$ centered at point r_i. The set of n disks $\{R_1, \ldots, R_n\}$ can be computed in $O(n \log m)$ time as follows: compute the Voronoi diagram of B and preprocess it for point location; after $O(m \log m)$ time, a point location query can be replied in $O(\log m)$ time. By locating each $r_i \in R$ in the Voronoi diagram, we get the points $b(r_1), \ldots, b(r_n)$ in $O(n \log m)$, which is enough information to construct the disks $\{R_1, \ldots, R_n\}$.

Let $\mathcal{A}$ be the arrangement produced by the set of n red disks $\{R_1, \ldots, R_n\}$. The idea of the algorithm is to associate a label l_c to each cell c of $\mathcal{A}$ with the number of disks from $\{R_1, \ldots, R_n\}$ that contain it, and then look for the cells in $\mathcal{A}$ with maximum label. This works because if a cell c has label k, it means that a blue point b inside this cell c is contained in exactly k red circles, which means that the point b is the closest point of the k red points corresponding to the red circles. The arrangement $\mathcal{A}$, together with the labels l_c for each cell $c \in \mathcal{A}$ can be constructed using the following result.

Theorem 2 [5, 13] *The arrangement of n circles with different radii can be computed by an incremental algorithm with $O(n\lambda_4(n)) = O(n^2 2^{\alpha(n)})$ worst-case running time or by a randomized incremental algorithm with $O(n^2)$ expected running time.*

Once we have computed the arrangement $\mathcal{A}$, we can construct the dual graph G of the arrangement containing a node for each cell $c \in \mathcal{A}$ and an edge between two cells whenever their closure intersects. If two faces $c, c' \in \mathcal{A}$ are adjacent in G, it is easy to compute the label $l_{c'}$ from the label l_c. Thus, making a traversal in the dual graph G, we can compute the labels l_c for all faces $c \in \mathcal{A}$. With this information, it is possible to compute l_c for all the edges and vertices $c \in \mathcal{A}$. Special care has to be paid if the arrangement is degenerate, that is, if some circles in $\{R_1, \ldots, R_n\}$ are tangent; details are standard and omitted. After computing l_c for all cells $c \in \mathcal{A}$, we can find the value MAXCOV(S) using that MAXCOV(S)= $\max\{l_c \mid c \in \mathcal{A}\}$ and report the locus $\mathcal{L}_S$ of all optimal placement using that $\mathcal{L}_S = \bigcup_{\{c \in \mathcal{A} : l_c = \text{MAXCOV}(S)\}} c$.

Theorem 3 *The value MAXCOV(S) and the set of all optimal placements $\mathcal{L}_S$ can be computed in $O(n^2 2^{\alpha(n)})$ worst-case running time or in $O(n^2)$ expected time.*

Notice that we can construct any of the level sets $L(k)$ in the same running time. The level set $L(1)$ is exactly the union of the n disks $R_1, \ldots, R_n$, which can be described in linear space and constructed in near-linear time [9]. Once we have a level set $L(k)$ under the MAXCOV criterion, we may be interested in one that optimizes the MAXMIN or MINMAX criteria. We will show below how to deal with this. We next pass to hardness of computing the optimum and approximation algorithms.

Theorem 4 *Computing MAXCOV(S) is 3SUM hard.*

The proof of this theorem is similar to the one used in [3]. For a set L of n lines with integer coefficients and distinct slopes, it is 3SUM hard to determine if three lines of L intersect in a common point [7]. We reduce this problem to the problem of computing the value MAXCOV(S).

In some applications, it may be that a quadratic time algorithm is not affordable and we would be satisfied with a placement for the new facility that is suboptimal but such that the number of clients it gets, is close to the optimal placement; in other words, and approximation algorithm for the optimization problem. We have seen that computing MAXCOV(S) is equivalent to finding the maximum depth in $\mathcal{A}$. It also follows that if we find a point b whose depth in $\mathcal{A}$ is d, then $|BRNN(b)| = d$, and so MAXCOV(S)$\geq d$. A probabilistic algorithm to find a point that $(1-\varepsilon)$-approximates the maximum depth in an arrangement of n disks is given by Aronov and Har-Peled [3], and it readily leads to the following result.

Theorem 5 *Given a parameter $\varepsilon > 0$, an $(1-\varepsilon)$-approximation of the value MAXCOV(S) and a witness placement can be computed in $O(n\varepsilon^{-2}\log n)$ expected time. The result is correct with high probability.*

3 The MINMAX and MAXMIN problems

We are given a bichromatic set $S = B \cup R$ formed by m blue points B (facilities) and n red points R (clients), $n \geq m \geq 2$, and a constraint region $X \subseteq L(1)$.

According to the MINMAX criterion we are interested in finding a new blue point $p \in X$ such that the maximum distance to the points in $BRNN(p)$ is minimized. Consider the cost function $Cost : L(1) \to \mathbb{R}$ that tells for each point $p \in L(1)$ the cost, according to the MINMAX criterion, of placing the new blue point, or facility, at p; it holds that $Cost(p) = \max\{d(p,x) : x \in BRNN(p)\}$. Consider the graph of the function $Cost$ in $3D$. In the following, we are going to give a combinatorial description of this graph.

Embed the plane containing R, B in the plane $z = 0$ in $3D$, that is, consider R, B as embedded in the xy-plane in 3D. For a point $r_i = (x_i, y_i) \in R$, consider the cylinder $Cyl_i = \{(x,y,z) \in \mathbb{R}^3 \mid (x-x_i)^2 + (y-y_i)^2 \leq (d(r_i, b(r_i)))^2\}$, which is the vertical, solid cylinder through the disk centered at r_i with radius $d(r_i, b(r_i))$, and consider the (surface) cone $Con_i = \{(x,y,z) \in \mathbb{R}^3 \mid (x-x_i)^2 + (y-y_i)^2 = z^2, z \geq 0\}$ with apex at point $(x_i, y_i, 0) \in R$. Finally, let Σ_i be the portion of the surface Con_i contained in Cyl_i. Observe that Σ_i is a surface patch with constant complexity.

The reason for considering Σ_i for each point r_i is the following: $\rho = (x,y,t) \in \mathbb{R}^3$ is a point vertically above (resp. below) Σ_i if and only if $r_i \in BRNN(x,y)$ and $d((x,y), r_i) \leq t$ (resp. $d((x,y), r_i) \geq t$). To see the validity of this claim, observe that ρ has a vertical above/below relation with r_i if and only if $\rho \in Cyl_i$. Moreover, by the way the cone Con_i is defined, it holds that $\rho = (x,y,t)$ is above (resp. below) Con_i if and only if $d((x,y), r_i) \leq t$ (resp. $d((x,y), r_i) \geq t$). Let U be the upper envelope of surfaces $\Sigma_1, \dots, \Sigma_n$. Using the discussion above we readily obtain the following property.

Lemma 6 *The upper envelope U is the graph of the function $Cost$.*

We are interested in finding a point $p \in X$ that minimizes $Cost$, and therefore the problem reduces to finding the lower point in the envelope U restricted to the region X. Let U_X be the portion of U defined over X. If X has complexity $O(n)$ we can argue that U_X has complexity $O(n^{2+\varepsilon})$ as follows, where the complexity of an envelope U_X is defined as its number of vertices, edges, and faces. For each boundary arc $a \in X$, we consider a vertical wall $W_a = a \times \mathbb{R}$ in 3D. Since X has $O(n)$ complexity, we have $O(n)$ surfaces of the type W_a.

The upper envelope U_W of the surfaces $\Sigma_1, \dots, \Sigma_n$ together with the walls W_a for arcs a in the boundary of X can be computed and described in $O(n^{2+\varepsilon})$ time, for any fixed $\varepsilon > 0$ [1]. However, since we have introduced the vertical walls W_a, the domain of each patch of U_W is either fully contained in X or fully outside X. It follows that the restriction U_X of U to X can be constructed in $O(n^{2+\varepsilon})$ time.

It remains to find the lower point of U_X. Observe that this point does not necessarily have to be a vertex. However, finding the lower point of U_X can be done by checking each piece of U_X, that is, each vertex, edge, and face. For a vertex and an edge in U_X, the lower point can be found in constant time, while for each face in U_X we can find the minimum in time proportional to its complexity. Using that the complexity of U_X is $O(n^{2+\varepsilon})$, we conclude the following.

Theorem 7 *The MINMAX problem can be solved in $O(n^{2+\varepsilon})$ time, for any fixed $\varepsilon > 0$.*

Using the same approach, the MAXMIN problem can be solved by computing the lower envelope L of $\Sigma_1, \dots, \Sigma_n$, considering its restriction L_X to a given set X, and finding the highest point in L_X.

Theorem 8 *The MAXMIN problem can be solved in $O(n^{2+\varepsilon})$ time, for any fixed $\varepsilon > 0$.*

4 MINMAX/MAXMIN criteria for optimal MAXCOV solutions

Here we describe how to find the best location b within $L(k)$ according to the MINMAX criterion, i.e., we want a smallest circle centered at $b \in L(k)$ while containing $BRNN(b)$. Let U be the upper envelope of the surface patches $\Sigma_1, \dots, \Sigma_n$. We want to find the lower point of U restricted to the locus $L(k)$, for some value k. Let U_k be the restriction of the upper envelope U to $L(k)$. We next argue that U_k has complexity $O(n^{2+\varepsilon})$ and can be constructed in $O(n^{2+\varepsilon})$ time. For each circle R_i, consider the cylinder $C_i = R_i \times \mathbb{R}$ in $\mathbb{R}^3$. The upper envelope U' of the surfaces $\Sigma_1, \dots, \Sigma_n, C_1, \dots, C_n$ has complexity $O(n^{2+\varepsilon})$ and can be constructed in $O(n^{2+\varepsilon})$ time [1]. Because we have included $C_1, \dots, C_n$ in the set of surfaces, the domain of each patch of U' is contained in a cell in the arrangement $\mathcal{A}$. The restriction of U_k to a cell of $c \in L(k)$ is the same as the restriction of U' to the same cell. We conclude that the envelope U_k has complexity $O(n^{2+\varepsilon})$, and we can find the lower point in U' using $O(n^{2+\varepsilon})$ time by checking each piece of U_k independently.

Theorem 9 *According to the MINMAX criterion, the best location in the set of placements in a level set $L(k)$ can be computed in $O(n^{2+\varepsilon})$ time, for any fixed $\varepsilon > 0$.*

Clearly, the same result applies if we replace the MINMAX criterion by the MAXMIN criterion.

5 Working with L_1 and L_∞-metrics

We consider now the L_∞ metric. For the MAXCOV criterion, the ideas described above directly apply, but they yield better running times. Like before, let R_i be the disk (square) with radius $d_\infty(r_i, b(r_i))$ centered at point r_i, and define the arrangement $\mathcal{A}$ induced by $\{R_1, \ldots, R_n\}$. We have to compute the maximum depth of $\mathcal{A}$. Although $\mathcal{A}$ may have quadratic complexity, the maximum depth in an arrangement of n rectangles can be found in $O(n \log n)$ time. This corresponds to a maximum clique in the intersection graph of squares [8]; alternatively, we may use a sweep-line algorithm maintaining a segment tree describing the depth of the line in the arrangement. Since the same argument applies to the L_1 metric, this leads to the following result.

Theorem 10 *In the L_∞ and L_1 metrics, we can compute the value MAXCOV(S) and a witness placement in $O(n \log n)$ worst-case running time.*

Observe that the description of all the optimal placements may take $\Omega(n^2)$, since it may consist of the union of many cells from $\mathcal{A}$. Of course, the 3SUM-hardness proof does not carry to the L_∞ or L_1 metric, and it does not make sense to consider approximation algorithms.

For the MINMAX criterion, the same ideas as described for the L_2 metric apply. For each point r_i, we consider the square cylinders $Cyl_i = R_i \times \mathbb{R}$, and the polyhedral cones Con_i such that its section at $z = t$ corresponds a square centered at r_i and side length $2t$. Notice that Σ_i is a surface consisting of 4 triangles, that is, 4 piece-wise linear patches. Like before, we want to compute the upper envelope of these linear patches, which can be done in $O(n^2\alpha(n))$ time [6]. The rest of the analysis carries out like before, and we obtain the following improved bound.

Theorem 11 *In the L_∞ and L_1 metrics, the MINMAX problem can be solved in $O(n^2\alpha(n))$ time.*

Acknowledgements

These problems were posed during the *Second Spanish Workshop on Geometric Optimization*, July, 2004, El Rocío, Huelva, Spain. The authors would like to thank the other participants for helpful comments.

References

[1] P.K. Agarwal, O. Schwarzkopf, M. Sharir. The overlay of lower envelopes and its applications. *Discrete Comput. Geom.*, 15, 1996, pp. 1–13.

[2] H.-K. Ahn, S.-W. Cheng, O. Cheong, M. Golin, R. van Oostrum. Competitive facility location: the Voronoi game. *Theoretical Comp. Sci.*, 310, 2004, pp. 457–467.

[3] B. Aronov, S. Har-Peled. On approximating the depth and related problems. *SODA*, 2005.

[4] R. Benetis, C.S. Jensen, G. Karčiauskas, S. Šaltenis. Nearest neighbor and reverse nearest neighbor queries for moving objects. *Symposium on Database Engineering & Applications*, 2002, pp. 44–53.

[5] H. Edelsbrunner, L. Guibas, J. Pach, R. Pollack, R. Seidel, M. Sharir. Arrangements of curves in the plane–topology, combinatorics, and algorithms. *Theoretical Comp. Sci.*, 92, 1992, pp. 319–336.

[6] H. Edelsbrunner, L. Guibas, M. Sharir. The upper envelope of piecewise linear functions: algorithms and applications. *Discrete Comput. Geom.*, 4, 1989, pp. 311–336.

[7] A. Gajentaan, M.H. Overmars. On a class of $O(n^2)$ problems in computational geometry. *Comput. Geom. Theory and Applications*, 5, 1995, pp. 165–185.

[8] H. Imai, T. Asano. Finding the connected components and a maximum clique of an intersection graph of rectangles in the plane. *J. Algorithms*, 4, 1983, pp. 310–323.

[9] K. Kedem, R. Livne, J. Pach, M. Sharir. On the union of Jordan regions and collision-free translational motion amidst polygonal obstacles. *Discrete Comput. Geom.*, 1, 1986, pp. 59–71.

[10] F. Korn, S. Muthukrishnan. Influence sets based on reverse nearest neighbor queries. *International Conference on Management of Data*, SIGMOD, Vol. 29.2, 2000, pp. 201–212.

[11] F. Korn, S. Muthukrishnan, D. Srivastava. Reverse nearest neighbor aggregates over data streams. *28th VLDB Conference*, 2002.

[12] A. Maheshwari, J. Vahrenhold, N. Zeh. On reverse nearest nighbor queries. *CCCG'02*, 2002.

[13] M. Sharir, P.K. Agarwal. *Davenport-Schinzel sequences and their geometric applications*, Cambridge University Press, 1995.

[14] A. Singh, H. Ferhatosmanoglu, A. Aman Tosun. High dimensional reverse nearest neighbor queries. *International Conference on Information and Knowledge Management*, 2003, pp. 91–98.

[15] M. Smid. Closest point problems in computational geometry. *Handbook on Computational Geometry, Elsevier*, 1997.

[16] Y. Tao, D. Papadias, X. Lian. Reverse *k*NN search in arbitrary dimensionality. *30th VLDB Conference*, 2004.

[17] G.T. Toussaint. Geometric proximity graphs for improving nearest neighbor methods in instance-based learning and data mining. *International Journal of Computational Geometry and Applications*, Vol. 15, No. 2, 2005, pp. 101–150.