A note on the inflating enclosing ball problem

Nikola K. Blanchard [1] and *Siargey Kachanovich* [2]

[1] IRIF, Université Paris Diderot, France & Loria, Université de Lorraine
[2] Université Côte d'Azur, INRIA Sophia-Antipolis, France

Extended Abstract

1 Introduction

Our goal in this paper is, given a connected set of balls, to select and inflate one ball to cover the whole set with the minimal radius. More formally, we are given an abstract metric space and a path-connected set of balls with given centres $c_1, c_2, \ldots, c_n$ and radii $r_1, r_2, \ldots, r_n$. We want to choose one of the centres and create a ball of radius R around it to cover the whole set of balls with minimal R. By using arguments from graph theory, we show that $R \leq r_a + \sum_j r_j$, where r_a is the mean of the two biggest radii among the r_i. This bound is tight. Finally, we show that in the usual complexity models, computing this centre requires $\Theta(n^2)$ operations.

This problem is closely related to the smallest enclosing ball problem, which has been studied since the early 1980s [Meg83, Wel91, MNV13]. The main difference is that, instead of computing the smallest enclosing ball without constraints on the centre, our centre is imposed from a small set, and we are concerned with precise bounds on the radius. Our motivation comes from facility location [GK99, CP04, Vyg05] and dynamic clustering problems [EMS14, ANS15], where fractional solutions to the linear programs generally come in the form of such structures. Being able to replace a connected set of balls by a single optimal ball in such problems could be a step towards solutions that have better approximation ratios and are easier to analyse, especially in the dynamic metric setting where some bounds are still far from being tight [BS17].

2 Upper and lower bounds

Definitions

We work in a general metric space E endowed with a distance $d(.,.)$ and the natural topology generated by open balls. Paths on E are well-defined as continuous functions $[0,1] \to E$.

We consider a set of n points $c_1, c_2 \ldots c_n$ in the metric space E, and to each point c_i associate a closed ball B_i of radius r_i (corresponding to all the points in E that are at distance at most r_i from c_i). The subspace corresponding to the union of all the balls will be written as S. We have one constraint on the the set of balls: the union S is assumed to be *path-connected*, meaning that there is a path from x to y through S, for any $x, y \in S$.

Problem

We seek to cover S by increasing the radius of a single ball, and have two questions. Which center c_i should we choose, and how can we minimise the ratio of the new radius R compared to the sum of the previous radii? That is, how do we minimise

$$\frac{R}{\sum_j r_j}.$$

The immediate answer is to find the ball closest to the "center" of S and open it with a sufficient radius, but that center might not be easy to manipulate in some abstract metric spaces and does not immediately give the best bound. Before proving the tight, bound we need an elementary lemma.

Lemma 1. *If a shortest path P between two arbitrary points $x, y \in S$ through S is of length d, then the sum of radii $\sum_j r_j$ is at least $d/2$.*

Proof. Without loss of generality, let $B_1, \ldots, B_k$ be the sequence of balls that the path P follows. Note that by the triangle inequality, the path P passes through each ball once — otherwise there exists a shorter path between x and y than P, which contradicts P being the shortest. Write $e_1, \ldots, e_k$ to be the "entry" points on the path P into the balls $B_1, \ldots, B_k$ respectively. We will define by convention $e_1 = x$ and $e_{k+1} = y$. We have $d(e_1, e_2) + d(e_2, e_3) + \ldots + d(e_k, e_{k+1}) = d$. By the triangle inequality with the centres $c_1, c_2, \ldots, c_k$, we finally get:

$$d = \sum_{i=1}^{k} d(e_i, e_{i+1}) \leq \sum_{i=1}^{k} (d(e_i, c_i) + d(c_i, e_{i+1})) \leq \sum_{i=1}^{k} 2r_i \leq 2 \sum_{i=1}^{n} r_i.$$

□

From Lemma 1 it follows that we can easily have $\frac{R}{\sum_j r_j} \leq 2$ by opening any facility with a big enough radius to cover everyone. Suppose that we take a centre c_i and a point x farthest from i, opening i with radius $d(i, x)$ is enough to cover S (because a point not inside that new ball would have to be at distance higher than $d(i, x)$). But the cost of the initial solution was at least $d(i, x)/2$, hence the result.

The previous result can be extended to arbitrary paths as long as their total length inside each ball is at most $2r_i$. If there is a path of length d satisfying such a condition, $\sum_j r_j \geq d/2$.

We also need a theorem from graph theory:

Theorem 2 ([WC04]). *For any tree graph T, we have:*

$$2 \times radius - \max_{i,j}(d(c_i, c_j)) \leq diameter.$$

Our goal is now to prove the following:

Theorem 3. *We can always cover S by a ball with radius $R \leq r_a + \sum_j r_j$, where r_a is the mean of the two biggest radii among the r_i.*

Proof. Let us consider a weighted complete graph K_n with $c_1, \ldots, c_n$ as nodes, where the weight of an edge (c_i, c_j) is equal to $d(c_i, c_j)$. Let T be a minimum spanning tree of the graph G. We expand T to a second tree T' in the following way. For each node c_i of T, we add a leaf l_{ij} for each c_j with $j \neq i$ (thus making the number of nodes in T' equal to n^2). The weight of an edge (l_{ij}, c_i) is set to be equal to the maximum distance between the centre c_j and a point in B_i.

We observe that, if we define the ball centred at a central node of T' with radius R equal to the radius of T', then this ball covers S entirely. Let us now take a path corresponding to the diameter of the tree T'. We know that its length d is less than $2\sum_j r_j$. We also know from Theorem 2 that $d \geq 2 \times R - \max_{i,j} d(c_i, c_j)$. However, $d(c_i, c_j)$ is at most the sum of the two biggest radii among the r_j, which we will denote by $2r_a$. By combining this with Lemma 1, we get that:

$$R \leq \frac{d}{2} + r_a \leq r_a + \sum_j r_j$$

□

Lemma 4. *This bound is tight.*

Proof. Consider two balls of radius r in the Euclidean space that touch in a single point. Any of these two balls has to be inflated with radius $3r$ to cover their union. This proves the lower bound. □

Remark 5. We can consider another situation in the Euclidean space, to show that this bound applies to instances of arbitrary sizes. Suppose that we have an even number $2n$ of identical balls of radius r in a line touching one another and a ball of a high radius r' on each side of the middle of the line. The optimal solution is then to inflate one the two big balls with sufficient radius to cover everyone other ball. The radius of the inflated ball in this case is $2n \times r + 3r_a$, while the sum of radii is $2n \times r + 2r_a$.

Remark 6. If we don't look at the values of the radii but only consider a multiplicative bound, this means that we can get examples with $\frac{3}{2} \leq \frac{R}{\sum_j r_j}$ (as in the first previous example), and the theorem becomes $\frac{R}{\sum_j r_j} \leq \frac{3}{2}$.

3 Algorithmic considerations

Using the same notations as before, we now look at the following algorithmic question: in how many operations can we find a centre $c^* \in \{c_1, \ldots, c_n\}$ such that we can cover all balls $B_1, \ldots, B_n$ by a ball centred at c^* with minimal possible radius R. The answer to this question depends on the choice of the model of how the input data is accessed. We will now investigate multiple models and give the lower bounds on the worst-case complexities for each of them.

Different input methods. We consider three models. The first and most general is: given any two centres c_1 and c_2, we have access to an oracle that outputs the distance between c_1 and c_2 in $O(1)$ time. This corresponds to a matrix representation of the complete distance graph.

An alternative is to have an adjacency list, where for each ball we can access in $O(1)$ time to its radius as well as to the list of all centres within the ball – potentially sorted by distance to the ball's centre.

General algorithm. We will now describe a simple algorithm that finds c^*. We compute for each centre c_i, the radius of the smallest ball centred at c_i that covers S. We will leave for now the complexity of finding this radius as a parameter k. To find c^*, we pick the centre with the minimal such radius. As we have n different centres, the complexity of the algorithm is thus $O(nk)$.

If we have access to the distance between c_i and c_j in $O(1)$ operations – or access to the list of distances between c_i and all c_j – we can check each center in $O(n)$, leading to an $O(n^2)$ algorithm. As shown in the following lemma, we cannot do better than this in the general case.

Lemma 7. *With the matrix representation, $\Omega(n^2)$ queries can be needed, making the previous algorithm asymptotically optimal.*

Proof. We will use an adversary argument, with a game for two players on a complete graph K_n, where the n nodes correspond to the centres $c_1, \ldots, c_n$. All edge weights in the graph K_n can be either 1 or $1 + \varepsilon$ for some fixed $\varepsilon \in (0, 1)$. Note that any assignment of the weights on the edges in the graph K_n satisfies the triangle inequality, hence the weighted graph K_n defines a metric space.

The first player (the user) chooses an edge in the graph and the second player (the adversary) chooses the weight of this edge from 1 and $1 + \varepsilon$. The goal of the user is to efficiently decide whether there exists a node in the graph K_n such that all its incident edges have weight 1, with the adversary trying to make the user lose as much time as possible. We will now show that the adversary has a strategy to make the user pick $\Omega(n^2)$ edges in K_n to decide.

Let e be an edge picked by the first player with two endpoints c_i and c_j. If e is not the last picked edge for either c_i or c_j, then the adversary assigns to the edge e the weight 1. Otherwise, the adversary assigns to e the weight $1 + \varepsilon$, unless c_i or c_j is the last node with no distance equal to $1+\varepsilon$. In this case, the adversary arbitrarily sets the distance to be either 1 or $1+\varepsilon$.

Recall that the user needs to check for each node if there is an incident edge with weight $1+\varepsilon$. When an edge with weight $1+\varepsilon$ is detected, this check is complete for the two endpoints of the edge. Therefore, at least $\lceil n/2 \rceil$ edges with weight $1+\varepsilon$ need to be queried. Moreover, by construction, for each such edge, at least $n-2$ edges of weight 1 were queried previously. We counted each edge of weight 1 at most twice, leaving us with at least $n(n-2)/4$ edges of weight 1 required to query. In total, the user is therefore required to pick $O(n^2)$ edges.

□

Remark 8. Although it requires an involved and technical proof, it can be shown that this instance can be embedded in $\mathbb{R}^n$. Any weight assignment to the edges of K_n satisfies the triangle inequality, corresponding to a sub-constrained problem that can be satisfied in $\mathbb{R}^n$.

4 Discussion

This paper leaves open multiple algorithmic questions. First, can better algorithms be found in the general case, using the number of edges as a parameter instead of the number of centres? For example, it could be possible to get $\tilde{O}(m)$ complexity by extending the constructive proof of Theorem 3, building a spanning tree and finding its centre in $O(m \log^* m)$. However, this requires finding a way to avoid the quadratic increase in size due to the construction of T'.

A different possible direction is to look not for optimal solutions but for satisficing ones, where any center with $R \le r_a + \sum_j r_j$ is considered a valid solution. In such a case, one could get rid of dense parts of the graph by only looking at the top three furthest neighbours and absorbing them into the ball currently considered. Most importantly, could the methods shown in this paper improve the bounds or give access to simpler proofs of the results for clustering and facility location problems, such as the ones in [CP04]?

Acknowledgements. This work was supported partly by the french PIA project "Lorraine Université d'Excellence", reference ANR-15-IDEX-04-LUE.

References

[ANS15] Hyung-Chan An, Ashkan Norouzi-Fard, and Ola Svensson. Dynamic facility location via exponential clocks. In *SODA*, pages 708–721, 2015.

[BS17] Nikola K. Blanchard and Nicolas Schabanel. Dynamic Sum-Radii Clustering. In Sheung-Hung Poon, Md. Saidur Rahman, and Hsu-Chun Yen, editors, *WALCOM: Algorithms and Computation: 11th International Conference and Workshops – WALCOM*, pages 30–41. Springer International Publishing, 2017.

[CP04] Moses Charikar and Rina Panigrahy. Clustering to minimize the sum of cluster diameters. *Journal of Computer and System Sciences*, 68(2):417–441, 2004.

[EMS14] David Eisenstat, Claire Mathieu, and Nicolas Schabanel. Facility location in evolving metrics. In *ICALP*, pages 459–470, 2014.

[GK99] Sudipto Guha and Samir Khuller. Greedy strikes back: Improved facility location algorithms. *Journal of Algorithms*, pages 228–248, 1999.

[Meg83] Nimrod Megiddo. Linear-time algorithms for linear programming in r^3 and related problems. *SIAM journal on computing*, 12(4):759–776, 1983.

[MNV13] Boris Mordukhovich, Nguyen Mau Nam, and Cristina Villalobos. The smallest enclosing ball problem and the smallest intersecting ball problem: existence and uniqueness of solutions. *Optimization Letters*, 7(5):839–853, 2013.

[Vyg05] Jens Vygen. Approximation algorithms for facility location problems. Lecture Notes, 2005.

[WC04] Bang Ye Wu and Kun-Mao Chao. *Eccentricities, diameters, and radii*, chapter 7, pages 165–173. CRC Press, 2004.

[Wel91] Emo Welzl. Smallest enclosing disks (balls and ellipsoids). In Hermann Maurer, editor, *New Results and New Trends in Computer Science*, pages 359–370, Berlin, Heidelberg, 1991. Springer Berlin Heidelberg.