Chapter 4
Big Triangle Small Triangle Method for the Weber Problem on the Sphere

Atsuo Suzuki

4.1 Introduction

We propose the Big Triangle Small Triangle (BTST) Method for solving the Weber problem on the sphere (WPS). It can also be applied to other single facility location problems on the sphere. The WPS is a variation of the Weber problem which is a classic and well-studied location problem. We assume that the demand points are distributed on the surface of a sphere, and our problem is to find the location of a facility so as to minimize the sum of the distances from demand points to the facility. The distance is measured by the great circle distance. The objective function of the WPS is not convex, and the Weiszfeld-type algorithm is not guaranteed to find the facility's optimal location. The BTST type algorithm divides the surface of the sphere into spherical triangles and applies a branch-and-bound method. We show that it finds the optimal solution within a short computational time.

As trade and services become global, there is an increased interest in location problems on a sphere. Most location problems, that are usually formulated on the plane, can also be formulated on the sphere. While planar distances are convex, great circle distances on the sphere are not. Therefore, even planar convex problems are not convex, in general, when formulated on the sphere. In order to find the global optimum on the sphere, global optimization techniques need to be utilized. One effective global optimization algorithm that was formulated on the plane is the "Big Triangle Small Triangle" (BTST) algorithm (Drezner and Suzuki 2004). We develop the spherical equivalent of the BTST algorithm.

The chapter is organized as follows: In Sect. 4.2 we review the BTST algorithm. In Sect. 4.3, we introduce the WPS, and provide a short literature review. In Sect. 4.4

A. Suzuki (✉)
Department of Systems and Mathematical Sciences, Nanzan University, Nagoya, Aichi, Japan
e-mail: atsuo@nanzan-u.ac.jp

H. A. Eiselt, V. Marianov (eds.), *Contributions to Location Analysis*, International Series in Operations Research & Management Science 281,
https://doi.org/10.1007/978-3-030-19111-5_4

we formulate the problem and list the properties we use for our algorithm. In Sect. 4.5 we describe the algorithm in detail. In Sect. 4.6 we derive the lower bound. In Sect. 4.7 we present the results of numerical experiments. We close the paper with the conclusion in Sect. 4.8. The derivation of a lower bound is detailed in the Appendix.

4.2 The Big Triangle Small Triangle Method

The Big Triangle Small Triangle method was proposed by Drezner and Suzuki (2004). The method solves optimally location problems with non-convex objective functions. Drezner and Suzuki (2004) solved the Weber problem with attraction and repulsion (WAR) and an obnoxious location problem. These problems have non-convex objective functions and unless we utilize the BTST method, it is difficult to obtain a guaranteed optimal solution.

The outline of the BTST method for a given relative accuracy ϵ is as follows:

1. Triangulation: The feasible region of the problem is divided into triangles. We utilize the Delaunay triangulation which is a dual graph of the Voronoi diagram generated by the demand points.
2. Initial upper bound *UB*: We evaluate the objective function at the centroid of each triangle and set the minimum of them as the initial upper bound.
3. Initial lower bound *LB* for each triangle: We evaluate a lower bound for each triangle.
4. Branch-and-bound
   (a) All triangles whose *LB* is greater than $UB(1-\epsilon)$ are discarded.
   (b) The triangle with the minimum *LB* (the "big triangle") is divided into four similar triangles ("small triangles") and the *UB* and *LB* in each triangle are evaluated. *UB* may be updated.
   (c) If the minimum of *LB* of all triangles is greater than $UB(1-\epsilon)$, then *UB* is the minimum of the objective function within a given relative accuracy ϵ. The solution is the centroid of the triangle which attains the *UB*.

The BTST method was motivated by the Big Square Small Square (BSSS) method proposed by Hansen et al. (1981, 1985). They solved variations of the Weber problem. The BTST method utilizes the triangulation of the feasible region into many triangles, while the BSSS method divides the feasible region, embedded in a big square, into small squares. The BSSS method is very simple and effective; however, the BTST method is also effective and is superior to the BSSS method in the following two points.

The first point is about the case when the feasible region is a polygon. In this case, the BSSS method divides the rectangle which includes the feasible region into rectangles which may have non-feasible areas. The BSSS method needs an additional procedure to avoid non-feasible solutions. On the other hand, the BTST method divides the polygon into triangles and all triangles and their interiors are feasible and the union of the triangles includes all feasible points.

The second point is that the triangles of the BTST method do not include demand points in their interior. A demand point in the interior of a triangle may complicate the derivation of an upper bound in a triangle. For example, in the Weber problem, the partial derivatives of the objective function are infinite at the demand points. In the BSSS method, some of the squares may include demand points and an upper bound may be ineffective. The triangles in the BTST method do not include demand points in their interior if we utilize the Delaunay triangulation described below.

The BTST method utilizes the Delaunay triangulation (Lee and Schachter 1980) which is a dual graph of the Voronoi diagram (Sugihara and Iri 1992; Okabe et al. 2000; Suzuki and Okabe 1995) to triangulate the feasible region. As described above, it is one of the reasons that BTST is better than BSSS. Voronoi diagrams and the Delaunay triangulation have been studied in the field of the computational geometry (Ohya et al. 1984; Sugihara and Iri 1992). The FORTRAN source program is available to the public on the WEB site of Kokichi Sugihara. Figure 4.1 is an example of the Voronoi diagram generated by randomly distributed 256 points on the plane.

Drezner and Suzuki studied various location problems using the Voronoi diagram. Examples include: the continuous p-center problem (Suzuki and Drezner 1996), the continuous hub model (Suzuki and Drezner 1997), the equitable radius circle problem (Suzuki and Drezner 2009), and the covering problem (Drezner and Suzuki 2010). They used the Delaunay triangulation (Lee and Schachter 1980) to triangulate the feasible region effectively in the application of the BTST algorithm.

Drezner utilized the BTST method for various location problems. They are listed in the introduction of Drezner (2007). After that Drezner has continued to apply the BTST method to various location problems, such as the multi-Weber problem

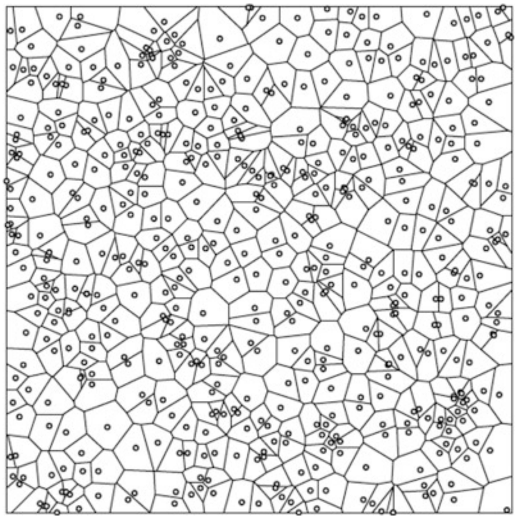


Fig. 4.1 Voronoi diagram for 256 generators by the program of Ohya et al. (1984)

(Suzuki and Drezner 2013), Huff based competitive location model (Drezner and Drezner 2004), and the network nuisance location problem (Drezner et al. 2009). It is almost impossible to obtain a guaranteed optimal solution for these problems without applying the BTST method.

4.3 The WPS Problem

The WPS problem is a variation of the Weber problem. The objective function of the classic Weber problem is convex, and by the algorithm first proposed by Weiszfeld (1936) and improved by Drezner (1996, 2015), the optimal solution is obtained. On the other hand, as the objective function of the WPS is not convex, the Weiszfeld-type algorithm cannot guarantee that the solution found is optimal. The WPS has been studied by many researchers, and many heuristics have been proposed for its solution. For example, Drezner and Wesolowsky (1978) proposed a heuristic based on the Weiszfeld algorithm. Other heuristic algorithms are surveyed by Hansen et al. (1995).

To the best of our knowledge, there are only two algorithms that obtain the optimal solution of the WPS. Drezner (1985) proposed an iterative algorithm based on the transformation of the problem into a series of minimax problems on the sphere. The algorithm adds extra points on the sphere one by one, and solves the minimax problem iteratively. The convergence of the algorithm to the optimal point is proved. However, numerical examples are not presented, and the convergence is expected to be slow.

Hansen et al. (1995) proposed another algorithm called the Big Region Small Region (BRSR) method based on the BSSS method. The algorithm of the BRSR divides the feasible region into spherical rectangles. It obtains a lower bound of the objective function in each spherical region. By the branch-and-bound process, they narrow the region where the solution exists. They implemented the algorithm and present computational results.

We apply the BTST type method described in the former section for solving the WPS. For the triangulation of the sphere, we utilize the algorithm of "spherical" Voronoi diagram developed by Sugihara (2002). Figure 4.2 is an example of the spherical Voronoi diagram drawn by the program available by Sugihara for randomly distributed 50 generators on the surface of the sphere.

The BTST method performs better than the BRSR method for the same reasons that the ordinary BTST method is superior to BSSS method. Furthermore, when the BRSR method divides the sphere into spherical rectangles, it uses latitude and longitude. As a result, it treats the regions near the north and south poles differently, while the BTST method does not require such special treatment.

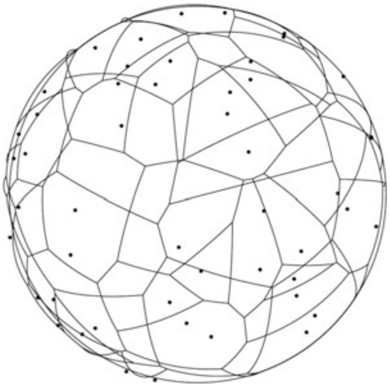


Fig. 4.2 Perspective view of a Voronoi diagram on the sphere for randomly distributed 50 generators by the program in Sugihara (2002)

4.4 Formulation of the Weber Problem on the Sphere

There are n demand points located on the sphere. Demand point i has an associated weight $w_i > 0$. The points on the sphere are represented by polar coordinates:

$$a_i = (\phi_i, \theta_i), -\frac{\pi}{2} \leq \phi_i \leq \frac{\pi}{2}, -\pi \leq \theta_i \leq \pi, i = 1, \ldots, n.$$

ϕ is the longitude and θ is the latitude if we consider the sphere as the globe.

The distance between two points on the sphere is measured by the great circle distance. The distance between $s(\phi, \theta)$ and $a_i(\phi_i, \theta_i)$ is

$$d(s, a_i) = \arccos\{\cos\phi \cos\phi_i \cos(\theta - \theta_i) + \sin\phi \sin\phi_i\}.$$

The great circle distance is the shortest distance from s to a_i when traveling on the surface of the globe. Note that $0 \leq d(s, a_i) \leq \pi$.

The objective function of the WPS is

$$F(s) = \sum_{i=1}^{n} w_i d(s, a_i). \tag{4.1}$$

We assume that $w_i > 0$. If $w_i < 0$ for some $1 \leq i \leq n$, we replace a_i by its antipode with weight $-w_i$. By Property 4.2 described below, we can transform the case where $w_i < 0$ for some values of i into problem (4.1). It means that if the WPS can be solved, the Weber problem with attraction and repulsion on the sphere can be solved as well.

As the WPS attracts significant interest, various properties have been studied. They are listed by Hansen et al. (1995). We show several definitions and properties which are utilized in our algorithm.

Definition 4.1 Given a center point and radius, a *spherical circle* is the set of the points whose shortest distance from the center is equal to the radius.

Definition 4.2 A set on the surface of the sphere is convex if all the points on the shortest arc connecting two points in the set are included in the set.

Definition 4.3 We define a function f on a convex set S on the surface of a sphere. f is convex if and only if for any two points a and b in S, and all $0 \leq \lambda \leq 1$,

$$f(\lambda a + (1-\lambda)b) \leq \lambda f(a) + (1-\lambda) f(b).$$

Property 4.1 The distance from a given point s to a point on the sphere is a convex function if the point is in the spherical circle whose center is s and radius is less than or equal to $\frac{\pi}{2}$.

Property 4.1 is proved in many papers such as Katz and Cooper (1980) and Drezner and Wesolowsky (1978). We utilize this property to obtain the lower bound in the BTST method.

Definition 4.4 The antipode of a point $a(\phi, \theta)$ on the sphere is $\bar{a}(-\phi, \theta - \pi)$ if $\theta \geq 0$ or $\bar{a}(-\phi, \theta + \pi)$ if $\theta \leq 0$.

The following property is easy to prove by the definition of the great circle distance and Definition 4.4.

Property 4.2 The sum of the distances from s to a and from s to $\bar{a}$ is equal to π.

Before describing the details of the algorithm, we re-write the objective function to facilitate the derivation of the lower bound for the BTST method. We define index sets of demand points. We consider a spherical triangle T whose vertices are $T^k, k = 1, 2, 3$. The polar coordinates of T^k are $(\phi^k, \theta^k), k = 1, 2, 3$. I is the index set of all demand points. We define the following:

$$I^k = \{i \in I | d(T^k, a_i) \leq \frac{\pi}{2}\}, k = 1, 2, 3$$
$$\bar{I^k} = I \setminus I^k.$$

I^k is a set of demand points whose distance from a vertex T^k of the triangle T is less than or equal to $\frac{\pi}{2}$.

We divide the objective function into three terms. The first term is the sum of the weighted distances from the demand points whose distance from all three vertices of the triangle is less than or equal to $\frac{\pi}{2}$. The second term is the sum of the weighted distances from the demand points whose distance from all three vertices of the triangle is greater than or equal to $\frac{\pi}{2}$. The third term is the sum for the rest of the demand points

$$\begin{aligned} F(s) &= \sum_{i \in I} w_i d(s, a_i) \\ &= \sum_{i \in \cap_{k=1}^{3} I^k} w_i d(s, a_i) + \sum_{i \in \cap_{k=1}^{3} \bar{I}^k} w_i d(s, a_i) + \sum_{i \in I \setminus (\cap_{k=1}^{3} I^k \cup \cap_{k=1}^{3} \bar{I}^k)} w_i d(s, a_i) \quad (4.2) \\ &\equiv F_1(s) + F_2(s) + F_3(s). \end{aligned}$$

The first term is convex for any s in the triangle, and the second term is concave. The third term is neither convex nor concave. We prove the convexity of the first term and the concavity of the second term.

First, we show that the first term of (4.2) is convex. Consider a spherical circle C_i whose center is a_i and radius $\frac{\pi}{2}$. By Property 4.1, $d(t, a_i)$ is a convex function in C_i. Note that $T^k \in C_i$. Let T' be a point on the edge T^1T^2. As $d(t, a_i)$ is a convex function in C_i, and C_i is convex, by Definition 4.3,

$$d(T', a_i) \leq \lambda d(T^1, a_i) + (1 - \lambda)d(T^2, a_i)) \leq \frac{\pi}{2}, 0 \leq \lambda \leq 1.$$

Let T'' be a point on the edge $T'T^3$. By the same reason above,

$$d(T'', a_i) \leq \lambda' d(T', a_i) + (1 - \lambda')d(T^3, a_i)) \leq \frac{\pi}{2}, 0 \leq \lambda' \leq 1.$$

As T'' moves in T when λ and λ' vary in their range,

$$d(s, a_i) \leq \frac{\pi}{2}, \forall s \in T$$

Then, by Property 4.1, $d(s, a_i)$ is convex function of $\forall s \in T$. As $F_1(s)$ is the weighted sum of the convex functions and the weights are positive, $F_1(s)$ is a convex function.

Next, we show that the second term of (4.2) is concave. By Property 4.2, the term is rewritten as follows:

$$\begin{aligned} F_2(s) = \sum_{i \in \cap_{k=1}^{3} \bar{I}^k} w_i d(s, a_i) &= \sum_{i \in \cap_{k=1}^{3} \bar{I}^k} w_i \{\pi - d(s, \bar{a}_i)\} \\ &= \pi \sum_{i \in \cap_{k=1}^{3} \bar{I}^k} w_i - \sum_{i \in \cap_{k=1}^{3} \bar{I}^k} d(s, \bar{a}_i). \quad (4.3) \end{aligned}$$

As the distance from s to a_i is larger than $\frac{\pi}{2}$ for $a_i, i \in \cap_{k=1}^{3} \bar{I}^k$, the distance from s to the $\bar{a}_i$ is less than or equal to $\frac{\pi}{2}$. It means that the distance from s to the $\bar{a}_i$ is a convex function of s. As the second term of (4.3) is the sum of convex functions, it is convex. The first term of (4.3) is constant. As F_2 is a difference between a constant and a convex function, it is a concave function. This can also be proved by observing that all the distances from the antipode are less than or equal to $\frac{\pi}{2}$ and in general if $F(X)$ is convex, then $-F(X)$ is concave.

4.5 Outline of the BTST Method for the Weber Problem on the Sphere

The outline of the algorithm is similar to the planar BTST method. In the algorithm, we use spherical triangles rather than planar triangles. The calculation of the lower bound is more complicated than the calculation for the planar BTST method. We show the details in the next section.

1. Triangulation: The sphere is divided into spherical triangles. We utilize the Delaunay triangulation of the sphere which is a dual graph of the Voronoi diagram on the sphere.
2. Initial upper bound *UB*: We evaluate the objective function at the centroid of the spherical triangles, and set the minimum of them as the initial upper bound.
3. Initial lower bound *LB* for each spherical triangle: We evaluate a lower bound of the objective function for each spherical triangle.
4. Branch-and-bound
   (a) All spherical triangles whose *LB* is greater than $UB(1-\epsilon)$ are discarded.
   (b) The spherical triangle with the minimum *LB* is divided into four similar spherical triangles by connecting the middle points of the three edges of the triangle. The *UB* and *LB* in these triangles are evaluated. *UB* may be updated.
   (c) If the minimum of *LB* of all spherical triangles is greater than $UB(1-\epsilon)$, then *UB* is the solution.

As the initial upper bound, we calculate the weighted sum of the distances from the centroid of each spherical triangle to the demand points. The initial upper bound *UB* is the minimum of these upper bounds. Note that if there are n demand points, the number of the triangles in the initial triangulation is $2n-4$. The calculation of the initial upper bound is calculated in linear time.

By our experience, the effectiveness of the algorithm strongly depends on the quality of the lower bound like in any branch-and-bound method. In the next section we describe the details of the calculation of the lower bound used in the computational experiments.

In the branch-and-bound process, we use a heap with the key of the lower bound of the spherical triangles. We scan the heap to discard spherical triangles with *LB* greater than the $UB(1-\epsilon)$. When we pick the spherical triangle with the smallest *LB*, the spherical triangle is at the top of the heap, and we update the heap. Then we add four spherical triangles to the heap. As the spherical triangle with minimum *LB* is always on the top of the heap, comparing the minimum *LB* to $UB(1-\epsilon)$ is quite easy.

4.6 Calculation of the Lower Bound

To obtain a lower bound, we need to consider the first term, which is convex, and the third term of (4.2), which is neither convex nor concave. The second term is concave. Therefore, its minimum should be attained at one of the vertices of the spherical triangle T.

The first term of (4.2) is transformed as follows. A point s in T is represented as a linear combination of the three vertices of the spherical triangle T^1, T^2, and T^3

$$G_1(t_1, t_2, t_3) \equiv F_1(t_1 T^1 + t_2 T^2 + t_3 T^3),$$

where

$$T^k = (\phi^k, \theta^k); \quad s = t_1 T^1 + t_2 T^2 + t_3 T^3, t_1 + t_2 + t_3 = 1. \tag{4.4}$$

As F_1 is convex in T, G_1 is convex in the triangle defined by (4.4). For a lower bound of a convex function, we propose the tangent plane method in Drezner and Suzuki (2004). We calculate the tangent plane of the convex function and set the minimum of the tangent plane values on the three vertices of the triangle as the lower bound. In this case, the lower bound is obtained as follows:

$$\begin{aligned} G_1(t_1, t_2, t_3) = \sum_{i \in \cap_{k=1}^3 I^k} & w_i \arccos\{\cos(t_1\phi^1 + t_2\phi^2 + t_3\phi^3)\cos\phi_i \\ & \times \cos(t_1\theta^1 + t_2\theta^2 + t_3\theta^3 - \theta_i) \\ & + \sin(t_1\phi^1 + t_2\phi^2 + t_3\phi^3)\sin\phi_i\}. \end{aligned}$$

The tangent plane of $G_1(t_1, t_2, t_3)$ at $(\bar{t}_1, \bar{t}_2, \bar{t}_3)$ is

$$\begin{aligned} \bar{G}_1(t_1, t_2, t_3) = {} & G_1(\bar{t}_1, \bar{t}_2, \bar{t}_3) \\ & + \left.\frac{\partial G_1}{\partial t_1}\right|_{t_1=\bar{t}_1, t_2=\bar{t}_2, t_3=\bar{t}_3} (t_1 - \bar{t}_1) \\ & + \left.\frac{\partial G_1}{\partial t_2}\right|_{t_1=\bar{t}_1, t_2=\bar{t}_2, t_3=\bar{t}_3} (t_2 - \bar{t}_2) \\ & + \left.\frac{\partial G_1}{\partial t_3}\right|_{t_1=\bar{t}_1, t_2=\bar{t}_2, t_3=\bar{t}_3} (t_3 - \bar{t}_3). \end{aligned} \tag{4.5}$$

As the value of $F_1(s)$ is equal to that of $G_1(t_1, t_2, t_3)$, $s = t_1 T^1 + t_2 T^2 + t_3 T^3$, and $G_1(t_1, t_2, t_3) \le \bar{G}_1(t_1, t_2, t_3)$,

$$F_1(s) \ge \bar{G}_1(t_1, t_2, t_3), s = t_1 T^1 + t_2 T^2 + t_3 T^3.$$

The derivation of $\partial G_1/\partial t_k$, $k = 1, 2, 3$ is given in the Appendix.

As mentioned above, $F_2(s)$ is concave. Therefore, $F_4(s) = F_2(s) + \bar{G}_1(t_1, t_2, t_3)$ is concave as a sum of a concave function and a linear function by Eq. (4.5). The minimum of $F_4(s)$ is attained at a vertex of T

$$\begin{aligned} F_4(s) &= \sum_{i \in \cap_{k=1}^3 \bar{I}^k} w_i d(s, a_i) + F_1(s) \\ \geq LB_1 &= \min_{k=1,2,3} \left\{ \sum_{i \in \cap_{k=1}^3 \bar{I}^k} w_i d(T^k, a_i) + \bar{G}^{(k)} \right\}, \end{aligned}$$

where $G^{(1)} = G_1(1, 0, 0)$; $G^{(2)} = G_1(0, 1, 0)$; $G^{(3)} = G_1(0.0.1)$.

$F_3(s)$ is neither convex nor concave. The distance from a demand point to any point in T is higher than the minimum distance from the demand point to the triangle T. Therefore,

$$\begin{aligned} F_3(s) &= \sum_{i \in I \setminus (\cap_{k=1}^3 I^k \cup \cap_{k=1}^3 \bar{I}^k)} w_i d(s, a_i) \\ \geq LB_2 &= \sum_{i \in I \setminus (\cap_{k=1}^3 I^k \cup \cap_{k=1}^3 \bar{I}^k)} w_i \min_{s \in T} d(s, a_i). \end{aligned}$$

The details of the derivation of $\min_{s \in T} d(s, a_i)$ is given in the Appendix.

As the lower bound LB_2 is a naïve one, its quality may not be very good. However, the size of the triangles is relatively small as the algorithm progresses. The number of demand points in $I \setminus (\cap_{k=1}^3 I^k \cup \cap_{k=1}^3 \bar{I}^k)$ becomes small. Therefore, the quality of the lower bound improves as the algorithm progresses.

As the result,

$$F(s) \geq LB_1 + LB_2.$$

4.7 Computational Experiments

We implemented the procedure in FORTRAN and tested randomly generated demand points on the sphere. The computer used for the experiments has Intel Core i5-4200U CPU (1.6 GHz clock), the operating system is Windows 10 Professional, and the RAM is 4.00 GB. The FORTRAN Compiler we used is FORTRAN Power Station version 4.

We tested problems with 100, 200, 300, 400, and 500 demand points which are randomly distributed on the sphere. All the weights are equal to 1. We compare the effectiveness of our algorithm with the one in Hansen et al. (1995). As they used $\epsilon = 10^{-3}$ as the stopping criterion, we used the same ϵ in our tests. We also used $\epsilon = 10^{-6}$ to obtain more precise optimal function values. For every combination of the number of demand points and ϵ, we solved 10 sample problems. We measure the number of iterations and CPU time to obtain the optimum objective function values. In Table 4.1 we report the average and standard deviation of the number of iterations and the run times in seconds of the 10 runs. We also report the average value of the objective function and for $\epsilon = 10^{-3}$ and report its percentage above the average obtained using $\epsilon = 10^{-6}$. In Table 4.2 we give the details of the 10 results for the $n = 500$ instances.

The computer (Hansen et al. 1995) used for the computational experiments is SPARC 4/75-64 (28.5 MIPS, 64M RAM, 4.2 MFLOPS). The computer we used has 4.84 GFLOPS (2.42 GFLOPS per one core and the CPU has two cores). The ratio of the computer speeds is 1152 times. In Table 4.1, the average CPU times are 0.16, 0.41, 0.77, 1.09, and 1.26 s for 100, 200, 300, 400, and 500 demand points sample problems, respectively, while in Hansen et al. (1995), the CPU times for the same size of the sample problems are 517, 958, 2586, 3018, and 5233 s. The ratios between these are 3230, 2320, 3360, 2740, and 4000. As the speed of the computers increases 1152 times, the CPU time by our algorithm is at least twice as fast.

For more precise solutions, we use $\epsilon = 10^{-6}$ as the stopping rule of the algorithm. It appears that the CPU times to obtain the solutions increase moderately.

Table 4.1 Summary results for $\epsilon = 10^{-3}$ and $\epsilon = 10^{-6}$

	Iterations		CPU time		Objective value	
n	Mean	S.D.	Mean	S.D.	Mean	[a]
$\epsilon = 10^{-3}$						
100	212.6	76.7	0.17	0.05	145.4025	0.0019%
200	241.8	91.4	0.41	0.11	299.2156	0.0004%
300	292.5	158.4	0.78	0.28	454.1721	0.0005%
400	265.6	198.3	1.09	0.45	604.7062	0.0002%
500	190.5	75.4	1.26	0.22	757.1503	0.0003%
$\epsilon = 10^{-6}$						
100	14555.3	28154.0	22.41	54.42	145.3997	–
200	20435.0	23103.1	41.21	51.96	299.2144	–
300	12837.4	24230.9	34.90	75.29	454.1697	–
400	8235.3	12164.0	23.30	36.79	604.7047	–
500	11040.1	26156.0	46.32	115.86	757.1481	–

[a] % above optimum

Table 4.2 Results for $n = 500$

Case	$\epsilon = 10^{-3}$			$\epsilon = 10^{-6}$		
	Iteration	Time	Value	Iteration	Time	Value
1	245	1.44	770.8228	1524	5.05	770.8228
2	118	1.05	752.8868	2385	8.06	752.8784
3	157	1.14	757.7859	703	2.72	757.7853
4	254	1.44	765.9260	2631	8.30	765.9211
5	79	0.94	741.5615	310	1.61	741.5582
6	313	1.59	776.6203	1676	5.39	776.6203
7	259	1.47	752.1417	89327	393.50	752.1416
8	181	1.25	756.8979	1413	4.80	756.8967
9	212	1.33	761.8746	3460	10.98	761.8717
10	87	0.97	734.9853	6972	22.83	734.9851
Mean	190.5	1.26	757.1503	11040.1	46.32	757.1481
S.D.	75.4	0.22	12.01	26156.0	115.86	12.01

4.8 Conclusions

We proposed the spherical BTST algorithm for the Weber problem on a sphere (WPS). It obtained the exact solution in a short CPU time. For the calculation of the lower bound of the branch and bound process of the BTST, we divide the objective function into three terms. The first one is the sum of the weighted distances from the point in the triangle to the demand points whose distance is less than $\frac{\pi}{2}$ from all the points in the spherical triangle. We show the term is convex and use a variation of the tangent plane method to obtain the lower bound. The second one is the sum of the weighted distances from the point in the triangle to the demand points whose distance is larger than or equal to $\frac{\pi}{2}$. We show that the second term is concave. Therefore, the sum of the second term and the tangent plane constructed for the first term is also concave. The lower bound for the sum is attained at one of the vertices of the triangle. For the third one, we calculate the shortest distance from the demand points which is neither less than $\frac{\pi}{2}$ nor larger than or equal to $\frac{\pi}{2}$. We implemented the algorithm and found that it obtained the solution in a short CPU time.

The methodology proposed in this paper can be used to optimally solve other location problems on a sphere. Competitive facility location, forbidden regions (such as bodies of water or enemy countries) using the Weber or the minimax objective, finding the location that attracts the maximum weight within a given distance, equity location models such as minimizing the range, the variance, and other location models that were investigated in the plane.

Appendix

Partial Derivatives of G_1

The calculation of the partial derivatives of $G_1(t_1, t_2, t_3)$ is as follows:

$$\begin{aligned} g_{1i}(t_1, t_2, t_3) \equiv{} & \cos(t_1\phi^1 + t_2\phi^2 + t_3\phi^3)\cos\phi_i \times \cos(t_1\theta^1 + t_2\theta^2 + t_3\theta^3 - \theta_i) \\ & + \sin(t_1\phi^1 + t_2\phi^2 + t_3\phi^3)\sin\phi_i. \end{aligned}$$

Then

$$G_1(t_1, t_2, t_3) = \sum_{i \in \cap_{k=1}^{3} I^k} w_i g_{1i}(t_1, t_2, t_3).$$

The partial derivatives of G_1 are

$$\frac{\partial G_1}{\partial t_k} = - \sum_{i \in \cap_{k=1}^{3} I^k} w_i (1 - g_{1i}(t_1, t_2, t_3)^2)^{\frac{1}{2}} \frac{\partial}{\partial t_k} g_{1i}(t_1, t_2, t_3), k = 1, 2, 3.$$

The partial derivatives of g_{1k} are calculated as follows:

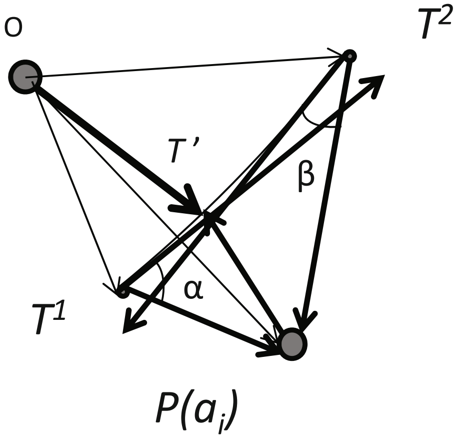
$$\begin{aligned} \frac{\partial}{\partial t_k} g_{1i}(t_1, t_2, t_3) ={} & \cos\phi_i\{-\phi^k \sin(t_1\phi^1 + t_2\phi^2 + t_3\phi^3) \cdot \cos(t_1\theta^1 + t_2\theta^2 + t_3\theta^3 - \theta_i) \\ & - \theta^k \cos(t_1\phi^1 + t_2\phi^2 + t_3\phi^3) \cdot \sin(t_1\theta^1 + t_2\theta^2 + t_3\theta^3 - \theta_i)\} \\ & + \sin\phi_i\{\phi^k \cos(t_1\phi^1 + t_2\phi^2 + t_3\phi^3)\}. \end{aligned}$$

Shortest Distance to the Triangle T

For the lower bound of $F_3(s)$, we need to calculate the shortest distance from a demand point to the triangle T. The shortest distance from a demand point is attained on the edges or the vertices of T. Therefore, we need to obtain the shortest distance from the demand point $P(a_i)$ to an edge T^1T^2 of the triangle T. We calculate the distance from the demand point to the three edges and set the minimum of them as the shortest distance from the demand point to the triangle T.

For the calculation of the distance from the demand point $P(a_i)$ to an edge T^1T^2, we need to consider two cases. Consider the spherical angles $\alpha = \angle PT^1T^2$ and $\beta = \angle PT^2T^1$ (see Fig. 4.3).

Fig. 4.3 The shortest distance from a demand point to the edge T^1T^2



The first case is that both α and β are less than $\pi/2$. In this case, the shortest distance from P to T^1T^2 is attained at a point T' on T^1T^2 see Fig. 4.3. The distance is calculated as

$$d(T', a_i) = \arccos(\overrightarrow{OT'} \cdot \overrightarrow{OP}),$$

where

$$\overrightarrow{OT'} = \frac{\overrightarrow{OP} - k(\overrightarrow{OT^1} \times \overrightarrow{OT^2})}{||\overrightarrow{OP} - k(\overrightarrow{OT^1} \times \overrightarrow{OT^2})||},$$

$$k = \frac{(\overrightarrow{OT^1} \times \overrightarrow{OT^2}) \cdot \overrightarrow{OP}}{||\overrightarrow{OT^1} \times \overrightarrow{OT^2}||}.$$

The second case is that either α or β is greater than or equal to $\frac{\pi}{2}$. If $\alpha \geq \frac{\pi}{2}$, the shortest distance is attained at T^1. If $\beta \geq \frac{\pi}{2}$, the shortest distance is attained at T^2.

References

Drezner, T., & Drezner, Z. (2004). Finding the optimal solution to the Huff competitive location model. *Computational Management Science, 1*, 193–208.

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

Drezner, Z. (1985). A solution to the Weber location problem on the sphere. *Journal of the Operational Research Society, 36*, 333–334.

Drezner, Z. (1996). A note on accelerating the Weiszfeld procedure. *Location Science, 3*, 275–279.

Drezner, Z. (2007). A general global optimization approach for solving location problems in the plane. *Journal of Global Optimization, 37*, 305–319.

Drezner, Z. (2015). The fortified Weiszfeld algorithm for solving the Weber problem. *IMA Journal of Management Mathematics, 26*, 1–9.

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

Drezner, Z., & Suzuki, A. (2010). Covering continuous demand in the plane. *Journal of the Operational Research Society, 61*, 878–881.

Drezner, Z., & Wesolowsky, G. O. (1978). Facility location on a sphere. *Journal of the Operational Research Society, 29*, 997–1004.

Hansen, P., Jaumard, B., & Krau, S. (1995). An algorithm for Weber's problem on the sphere. *Location Science, 3*, 217–237.

Hansen, P., Peeters, D., Richard, D., & Thisse, J.-F. (1985). The minisum and minimax location problems revisited. *Operations Research, 33*, 1251–1265.

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

Katz, I. N., & Cooper, L. (1980). Optimal location on a sphere. *Computers & Mathematics with Applications, 6*, 175–196.

Lee, D. T., & Schachter, B. J. (1980). Two algorithms for constructing a Delaunay triangulation. *International Journal of Parallel Programming, 9*, 219–242.

Ohya, T., Iri, M., & 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.

Okabe, A., Boots, B., Sugihara, K., & Chiu, S. N. (2000). *Spatial tessellations: Concepts and applications of Voronoi diagrams*. *Wiley series in probability and statistics*. New York: Wiley.

Sugihara, K. (2002). Laguerre Voronoi diagram on the sphere. *Journal for Geometry and Graphics, 6*, 69–81.

Sugihara, K., & Iri, M. (1992). Construction of the Voronoi diagram for "one million" generators in single-precision arithmetic. *Proceedings of the IEEE, 80*, 1471–1484.

Suzuki, A., & Drezner, Z. (1996). The *p*-center location problem in an area. *Location Science, 4*, 69–82.

Suzuki, A., & Drezner, Z. (1997). On the airline hub problem: The continuous model. *The Journal of the Operations Research Society of Japan, 40*, 62–74.

Suzuki, A., & Drezner, Z. (2009). The minimum equitable radius location problem with continuous demand. *European Journal of Operational Research, 195*, 17–30.

Suzuki, A., & Drezner, Z. (2013). Solving constrained two-facility location problems. *The Journal of the Operations Research Society of Japan, 56*, 157–165.

Suzuki, A., & Okabe, A. (1995). Using Voronoi diagrams. In Z. Drezner (Ed.), *Facility location: A survey of applications and methods* (pp. 103–118). New York: Springer.

Weiszfeld, E. (1936). Sur le point pour lequel la somme des distances de n points donnés est minimum. *Tohoku Mathematical Journal, 43*, 355–386.