An Algorithmic Review of Largest Empty Circle Problem Using Voronoi Diagrams

Samriddha Pathak

1 Introduction

The Largest Empty Circle (LEC) problem represents one of the fundamental challenges in computational geometry, with applications spanning facility location theory, wireless network design, and urban planning. The solution to this problem relies heavily on the geometric properties of Voronoi diagrams and convex hulls, making it essential to understand these foundational concepts before delving into algorithmic approaches.

1.1 Voronoi Diagrams: Fundamental Concepts

Voronoi diagrams, named after the mathematician Georges Voronoi, constitute one of the most important geometric structures in computational geometry. These diagrams partition the plane into regions based on proximity to a given set of points, called sites, or generators.

Definition 1.1 (Voronoi Diagram)**.** Given a set of distinct points $P = \{p_1, p_2, \ldots, p_n\}$ in the Euclidean plane, the Voronoi diagram $\mathrm{Vor}(P)$ is a partition of the plane into n regions such that each region $V(p_i)$ contains all points closer to p_i than to any other point in P. Formally:

$$V(p_i) = \{x \in \mathbb{R}^2 : \|x - p_i\| \leq \|x - p_j\| \text{ for all } j \neq i\} \tag{1}$$

Each Voronoi cell $V(p_i)$ is a convex polygon (possibly unbounded) whose boundaries consist of perpendicular bisectors of line segments connecting p_i to its neighboring sites. The intersection points of these bisectors form Voronoi vertices, while the bisector segments themselves constitute Voronoi edges.

The significance of Voronoi diagrams in solving the LEC problem stems from their proximity properties: At any Voronoi vertex, the distances to the surrounding sites are equal, making these vertices natural candidates for optimal circle centers.

1.2 Types of Voronoi Diagrams

1.2.1 2D Euclidean Voronoi Diagrams

The classical 2D Euclidean Voronoi diagram operates in the standard Cartesian plane using the Euclidean distance metric. This represents the variant most commonly studied and forms the basis for LEC problem solutions.

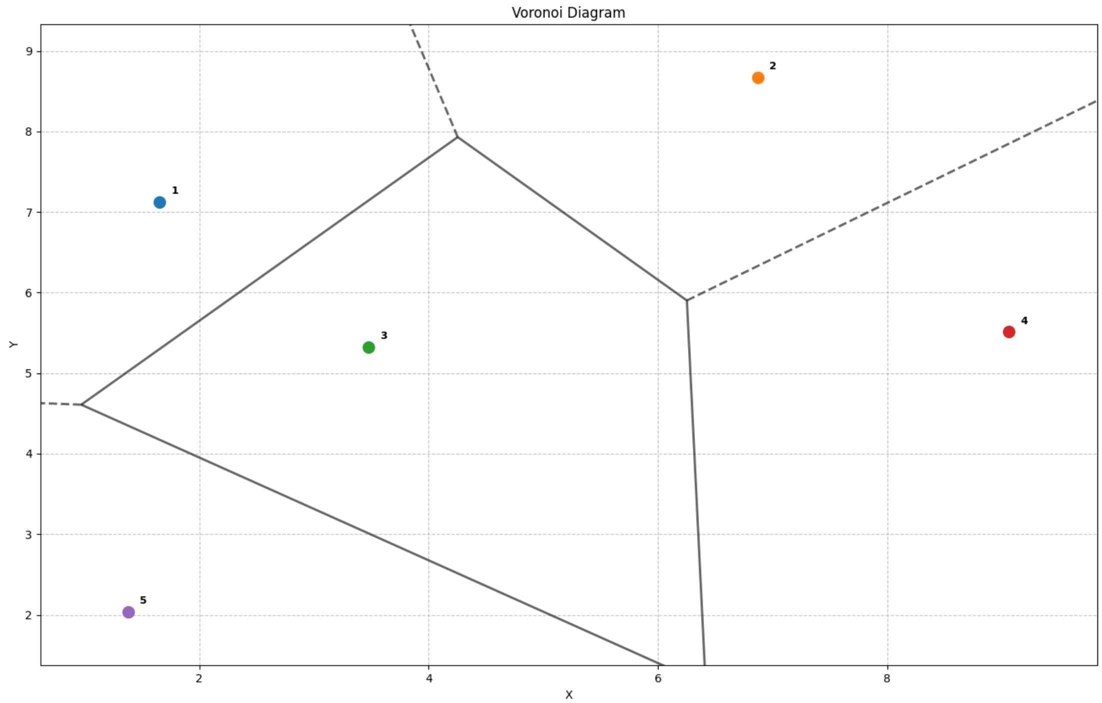

Figure 1: 2D Euclidean Voronoi diagram with few sites (5-7 points). Each colored point represents a site, with black lines showing cell boundaries.

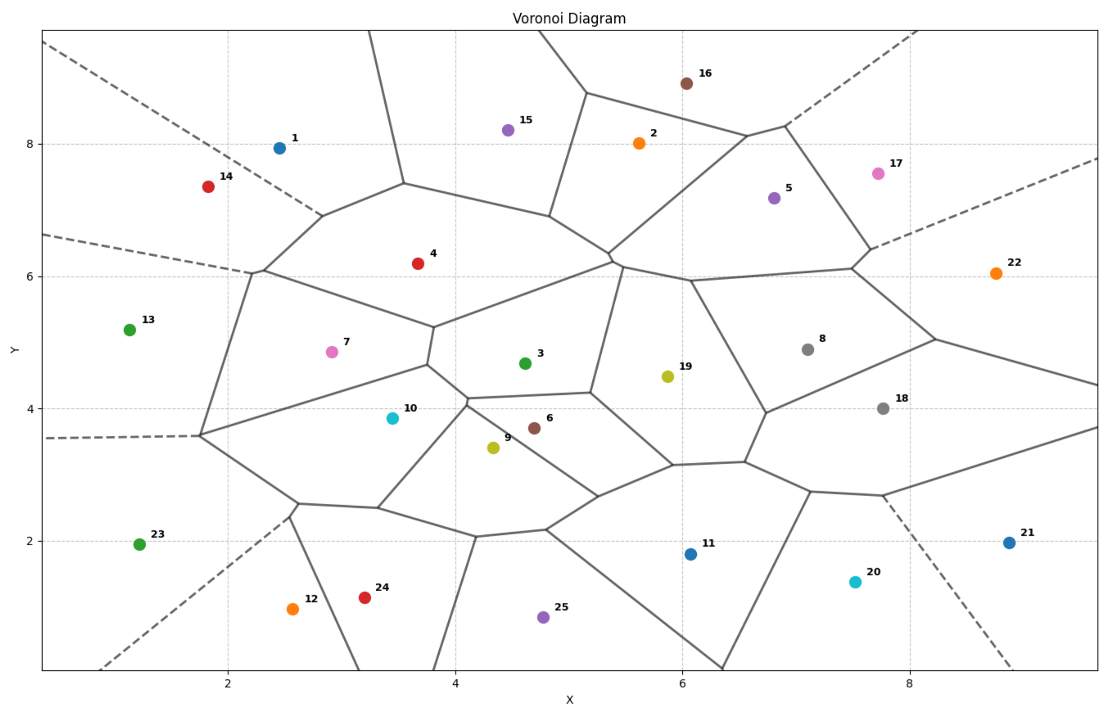

Figure 2: 2D Euclidean Voronoi diagram with numerous sites (15+ points). The increased density creates a more complex cellular structure, demonstrating the scalability of Voronoi partitioning.

The mathematical properties of 2D Euclidean Voronoi diagrams include linear complexity in terms of vertices and edges, with exactly $2n - 5$ vertices and $3n - 6$ edges for n sites in general position.

1.2.2 Spherical Voronoi Diagrams

Spherical Voronoi diagrams extend the concept to the surface of a sphere, using geodesic distances instead of Euclidean distances. Each Voronoi cell on the sphere is bounded by great circle arcs rather than straight lines.

Definition 1.2 (Spherical Voronoi Diagram)**.** For sites $P = \{p_1, p_2, \ldots, p_n\}$ on the unit sphere S^2, the spherical Voronoi diagram partitions the sphere such that:

$$V_s(p_i) = \{x \in S^2 : d_s(x, p_i) \leq d_s(x, p_j) \text{ for all } j \neq i\} \tag{2}$$

where $d_s(x, y)$ denotes the geodesic distance on the sphere.

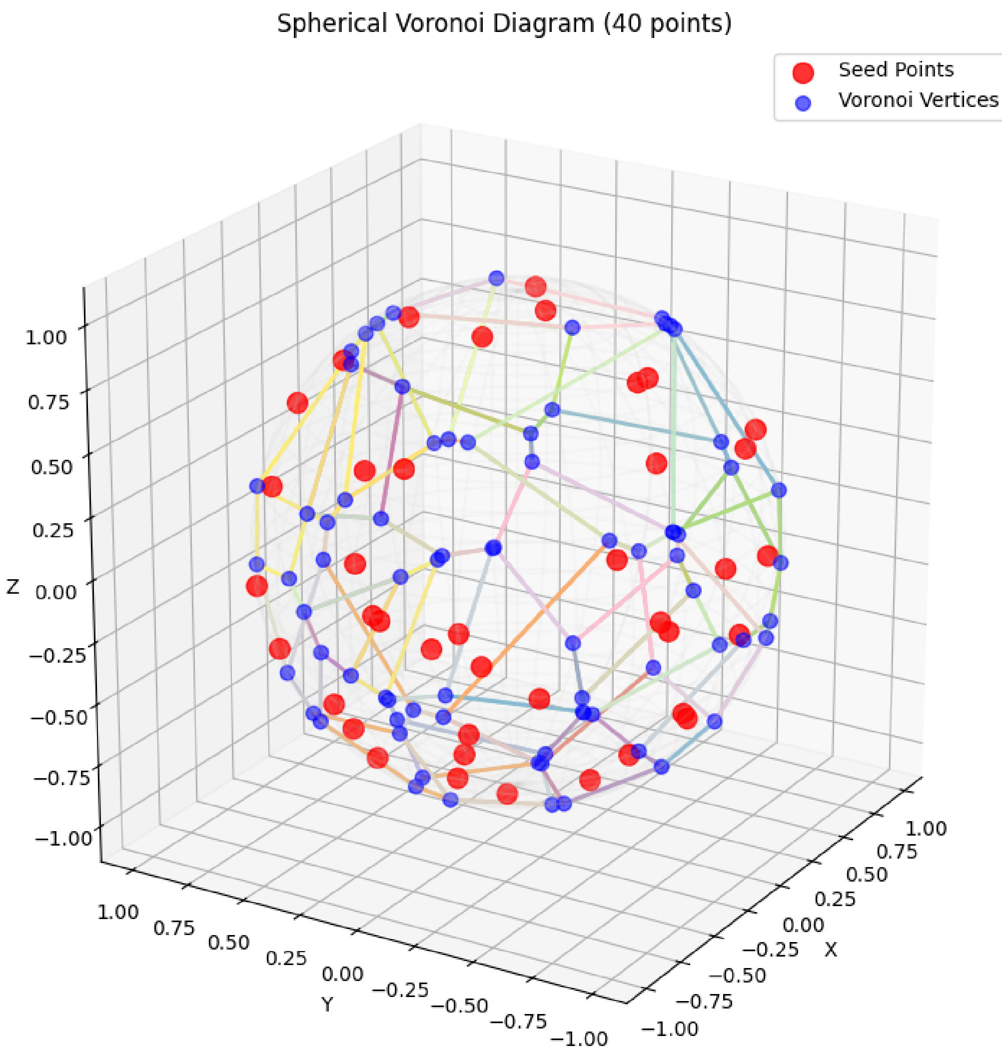

Figure 3: Spherical Voronoi diagram showing the partition of a sphere surface into geodesic-bounded regions.

Spherical Voronoi diagrams satisfy Euler's formula for polyhedra: $V - E + F = 2$, where V, E, and F represent vertices, edges, and faces, respectively. This constraint creates exactly $2n - 4$ vertices and $3n - 6$ edges for n sites.

1.3 Terminology and Attributes

Understanding Voronoi diagrams requires familiarity with key terminology:

- **Sites (Generators):** The input points p_i that define the diagram
- **Voronoi Cell:** The region $V(p_i)$ associated with site p_i
- **Voronoi Edge:** A line segment forming the boundary between two adjacent cells
- **Voronoi Vertex:** The intersection point of three or more Voronoi edges
- **Dual Graph:** The Delaunay triangulation, which connects sites whose Voronoi cells share an edge
- **Circumcenter:** The center of the circumcircle passing through the vertices of a Delaunay triangle, corresponding to a Voronoi vertex

The fundamental duality between Voronoi diagrams and Delaunay triangulations provides computational advantages, as algorithms can operate on either representation depending on the specific requirements.

1.4 Natural Voronoi Patterns

Voronoi-like structures appear frequently in nature, demonstrating the fundamental role of proximity-based partitioning in physical and biological systems.

Soap Bubble Patterns: When soap bubbles cluster together, they naturally form boundaries that minimize surface energy. These boundaries approximate Voronoi edges, and each bubble occupies the region closest to its center. The mathematical principle underlying this phenomenon is Plateau's laws, which govern minimal surface configurations.

Giraffe Skin Patterns: The distinctive polygonal patches on the skin of the giraffe follow Voronoi-like tessellations. Each patch develops around a growth center during embryonic development, with boundaries forming where different growth regions meet. This biological process creates natural proximity-based partitioning.

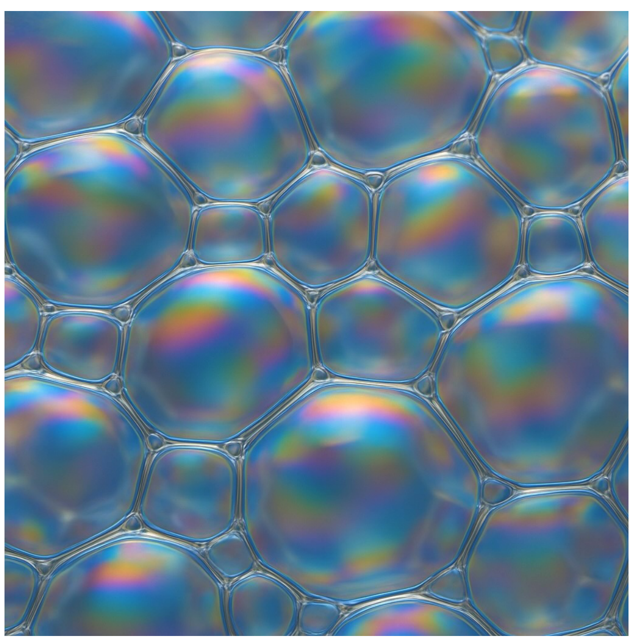

Figure 4: Soap bubble foam structure showing natural Voronoi-like cell boundaries. The bubbles arrange themselves to minimize surface energy, creating polygonal patterns similar to mathematical Voronoi diagrams.

Figure 5: Giraffe skin pattern exhibiting natural Voronoi tessellation. Each polygonal patch corresponds to a growth center, with boundaries forming the natural Voronoi edges between adjacent regions.

Other examples include crystal growth patterns, territorial divisions in animal behavior, and cellular structures in plant tissues, all demonstrating the universality of Voronoi-type organization in natural systems.

1.5 Algorithms for Voronoi Diagram Construction

1.5.1 2D Euclidean Algorithms

Several algorithms exist for constructing 2D Voronoi diagrams, each with distinct computational characteristics:

Fortune's Sweep Line Algorithm: This optimal $O(n \log n)$ algorithm uses a sweep line that moves across the plane, maintaining a beach line of parabolic arcs. As the sweep line progresses, it detects circle events that correspond to Voronoi vertices.

Algorithm 1 Fortune's Sweep Line Algorithm

```
Sort sites by y-coordinate
Initialize event queue with site events
Initialize empty beach line
while event queue not empty do
    Process next event (site or circle event)
    if site event then
        Insert new parabola into beach line
        Check for new circle events
    else
        Remove parabola from beach line
        Create Voronoi vertex
        Check for new circle events
    end if
end while
return Voronoi diagram
```

Incremental Algorithm: This approach adds sites one by one, updating the diagram incrementally. Although conceptually simpler, it typically requires $O(n^2)$ time in the worst case.

Divide and Conquer: This method recursively divides the set of points, constructs Voronoi diagrams for sub-problems, and merges the results. It achieves $O(n \log n)$ complexity, but involves complex merging procedures.

1.5.2 Spherical Voronoi Algorithms

Spherical Voronoi diagram construction requires specialized algorithms that handle the curved geometry of the sphere:

Stereographic Projection Method: Projects the sphere onto the plane, constructs the planar Voronoi diagram, and projects back to the sphere. This method preserves many geometric properties, but requires careful handling of the projection point.

Direct Spherical Construction: Works directly on the sphere surface using spherical geometry. This approach avoids projection distortions, but requires more complex geometric computations involving great circles and spherical trigonometry.

Algorithm 2 Spherical Voronoi Construction

1: Project sites from sphere to plane via stereographic projection
2: Construct planar Voronoi diagram using Fortune's algorithm
3: Project Voronoi vertices and edges back to sphere
4: Convert straight edges to great circle arcs
5: Handle special cases near projection point
6: **return** Spherical Voronoi diagram

The choice between these methods depends on the specific application requirements, with direct methods preferred for high-precision applications and projection methods suitable for general-purpose computing.

1.6 Convex Hull: Definition and Properties

The convex hull concept is fundamental to the LEC problem, as it defines the feasible region within which the largest empty circle must be centered.

Definition 1.3 (Convex Hull)**.** For a set of points $P = \{p_1, p_2, \dots, p_n\}$ in the Euclidean plane, the convex hull $\mathrm{CH}(P)$ is the smallest convex set containing all points in P. Equivalently, it is the intersection of all half-planes containing P:

$$\mathrm{CH}(P) = \bigcap_{\text{all half-planes } H \text{ containing } P} H \tag{3}$$

The convex hull can be visualized as the shape formed by stretching a rubber band around the outermost points of the set. Mathematically, it represents all convex combinations of the input points:

$$\mathrm{CH}(P) = \left\{ \sum_{i=1}^{n} \lambda_i p_i : \lambda_i \geq 0, \sum_{i=1}^{n} \lambda_i = 1 \right\} \tag{4}$$

Proposition 1.4 (Convex Hull Properties)**.** *The convex hull of n points in the plane has the following properties:*

(1) It is a convex polygon with at most n vertices

(2) Its vertices are a subset of the original point set

(3) It can be computed in $O(n \log n)$ time optimally

(4) It provides the minimal convex container for the point set

The boundary of the convex hull, denoted $\partial\mathrm{CH}(P)$, consists of line segments connecting consecutive hull vertices. These segments are called convex hull edges and play a crucial role in the solution of the LEC problem, since optimal circle centers may lie at intersections between Voronoi edges and hull edges.

1.7 Convex Hull Illustrations

Understanding the visual characteristics of convex hulls is essential for grasping their role in the LEC problem.

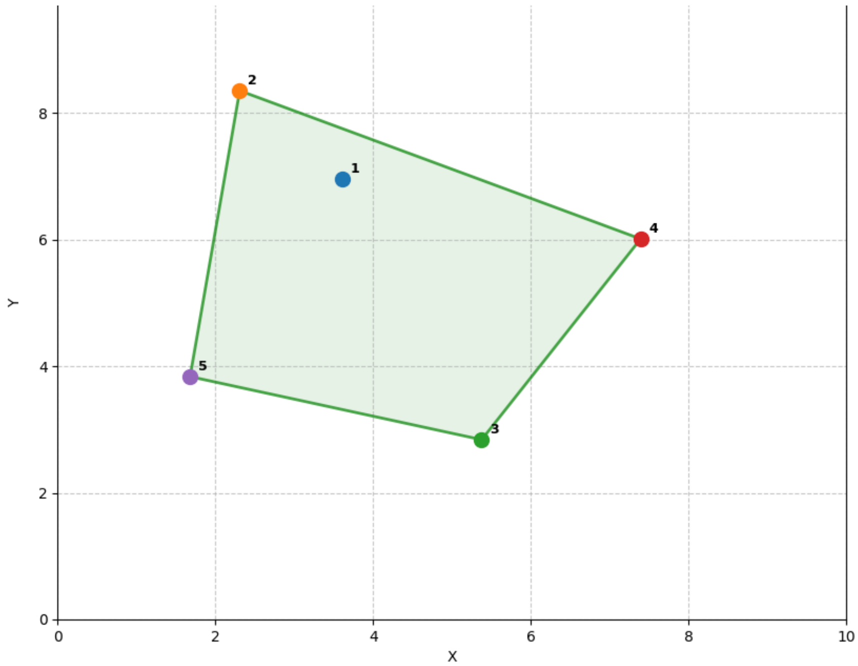

Figure 6: Simple convex hull example with 5 points. The green lines show the convex hull edges.

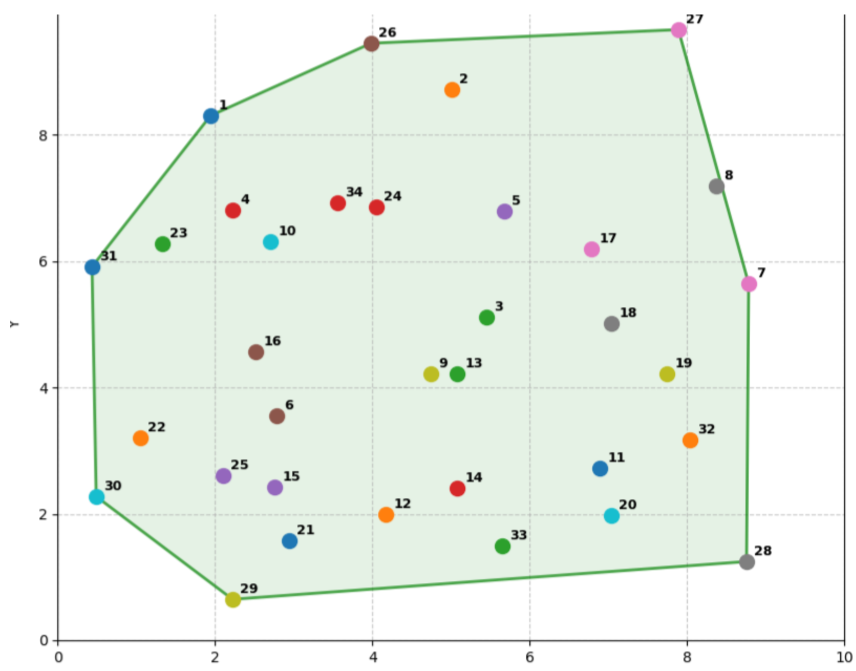

Figure 7: Complex convex hull with many points showing how the hull captures only the outermost boundary. Despite having numerous interior points, the hull remains a simple polygon defined by its extreme points.

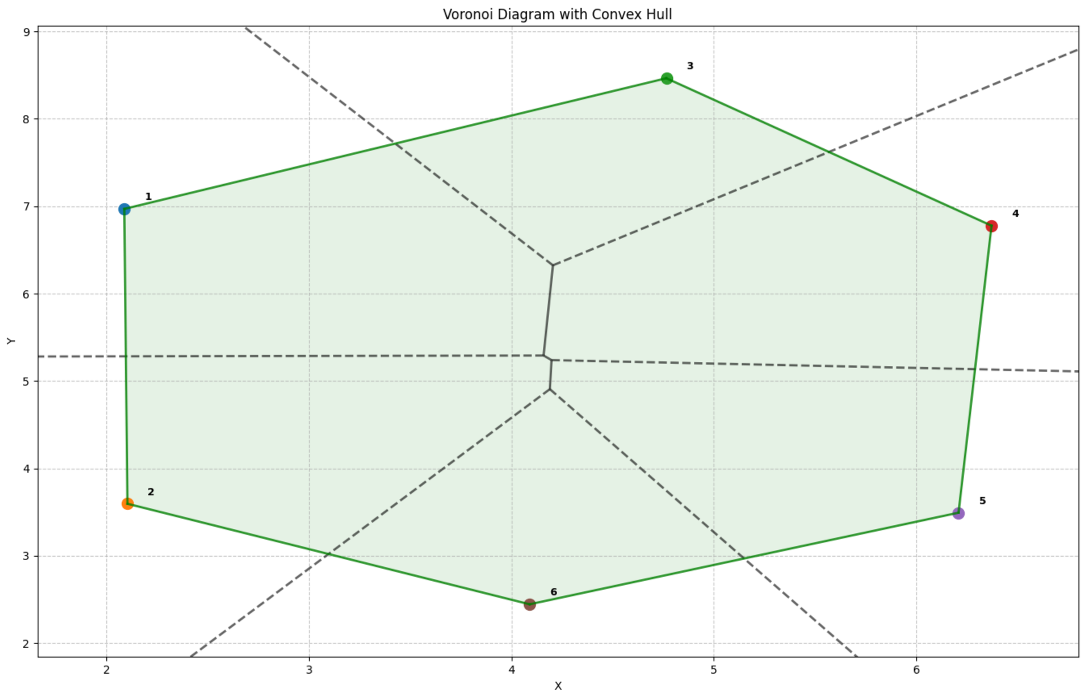

Figure 8: Overlay of Voronoi diagram and convex hull, illustrating the relationship between these structures in the context of the LEC problem. The intersection of Voronoi edges with hull boundaries creates candidate locations for optimal circle centers.

The interaction between Voronoi diagrams and convex hulls forms the theoretical foundation for LEC algorithms. Voronoi vertices that lie within the convex hull represent points equidistant from multiple sites, while intersections between Voronoi edges and hull boundaries create additional candidate locations where the optimal empty circle might be centered.

1.8 Conclusion

This introduction establishes the mathematical and geometric foundations necessary for understanding LEC problem solutions. Voronoi diagrams provide the proximity structure that identifies candidate locations for optimal circle centers, while convex hulls define the feasible region constraining the solution space. The duality between these geometric structures, combined with their efficient algorithmic construction, enables the development of optimal LEC algorithms.

The natural occurrence of Voronoi patterns in physical and biological systems demonstrates the fundamental importance of proximity-based spatial organization. From soap bubble arrangements to animal skin patterns, these structures reflect universal principles of space partitioning that have been mathematically formalized in computational geometry.

Understanding both 2D Euclidean and spherical variants of Voronoi diagrams, along with their construction algorithms, provides the computational tools necessary for practical LEC problem implementation. The integration of these concepts with convex hull theory creates a comprehensive framework for analyzing and solving largest empty circle problems across diverse application domains.

The subsequent sections will build upon these foundational concepts to develop specific algorithms for the LEC problem, demonstrating how the theoretical properties of Voronoi diagrams and convex hulls translate into efficient computational solutions for this classical problem in computational geometry.