

Python Functions Documentation

`def sort_edge(edge):`

Sorts the endpoints of an edge.

Parameters:

`edge (tuple)`: A tuple representing an edge, where each element is a point (e.g., (p1, p2)).

Returns:

`tuple`: A tuple representing the edge with the points sorted in ascending order.

`def eliminate_hull(triangles, edges):`

Eliminates edges that are part of the convex hull from a set of triangles.

Parameters:

`triangles (list of tuples)`: A list of tuples, where each tuple represents a triangle with three points.

`edges (list of tuples)`: A list of edges that are part of the convex hull.

Returns:

`list`: A list of edges that are not part of the convex hull.

`def check_intersection(edge, circle):`

Checks if an edge intersects with a circle.

Parameters:

`edge (tuple of np.array)`: A tuple representing an edge with two endpoints (e.g., (p1, p2)).

`circle (tuple)`: A tuple containing the radius and center of the circle (e.g., (R, center)).

Returns:

`tuple`: A tuple containing:

- An integer representing the number of intersection points (0, 1, or 2).
- A tuple containing the intersection points, if any.

```
def check_inside(edge, circle, tol=1e-6):
```

Checks if either endpoint of an edge is inside a circle.

Parameters:

edge (tuple of np.array): A tuple representing an edge with two endpoints (e.g., (p1, p2)).

circle (tuple): A tuple containing the radius and center of the circle (e.g., (R, center)).

tol (float, optional): A tolerance value for determining if a point is inside the circle. Defaults to 1e-6.

Returns:

bool: True if either endpoint is inside the circle, False otherwise.

```
def generate_data(edges, circles):
```

Generates a matrix A representing the intersection of edges with circles, and a list of new edges formed by intersections.

Parameters:

edges (list of np.array): A list of edges where each edge is represented as a NumPy array of two points.

circles (list of tuples): A list of tuples representing circles, where each tuple contains the radius and center (e.g., (R, center)).

Returns:

tuple: A tuple containing:

- A (np.array): A binary matrix where each row represents an edge and each column represents a circle.

- New_edges (list of np.array): A list of new edges formed by intersections with the circles.

```
def generate_problem(nodes, circles, rmin, rmax, method=1):
```

Generates a problem instance for the set covering problem using nodes, circles, and edges.

Parameters:

nodes (int): The number of nodes to generate.

`circles (int)`: The initial number of circles to generate.

`rmin (float)`: The minimum radius of the circles.

`rmax (float)`: The maximum radius of the circles.

`method (int, optional)`: The method to increase the number of circles or their radii. Defaults to 1.

Returns:

`tuple`: A tuple containing:

- `A (np.array)`: A binary matrix representing the coverage of edges by circles.
- `edges (list of tuples)`: The original list of edges.
- `New_edges (list of np.array)`: The list of new edges formed by intersections with circles.
- `circles_data (list of tuples)`: A list of tuples representing the circles with their radii and centers.
- `Ws (np.array)`: An array of weights for the circles.
- `circles (int)`: The updated number of circles.
- `Vs (np.array)`: The array of nodes.
- `rmin (float)`: The updated minimum radius of the circles.
- `rmax (float)`: The updated maximum radius of the circles.
- `Nedges (int)`: The number of edges.

`def Update_Cols(A, circles_data, edges_data, Ws, rmin, rmax, tol=1e-6):`

Updates the matrix `A` by adjusting circle radii to ensure all edges are covered.

Parameters:

`A (np.array)`: The binary matrix representing the coverage of edges by circles.

`circles_data (list of tuples)`: A list of tuples representing the circles with their radii and centers.

`edges_data (list of np.array)`: A list of edges where each edge is represented as a NumPy array of two points.

`Ws (np.array)`: An array of weights for the circles.

`rmin (float)`: The minimum radius of the circles.

`rmax (float)`: The maximum radius of the circles.

tol (float, optional): A tolerance value for intersection calculations. Defaults to $1e-6$.

Returns:

tuple: A tuple containing:

- A (np.array): The updated binary matrix.
- circles_data (list of tuples): The updated list of circles.
- edges_data (list of np.array): The updated list of edges.
- Ws (np.array): The updated array of weights for the circles.
- rmax (float): The updated maximum radius of the circles.