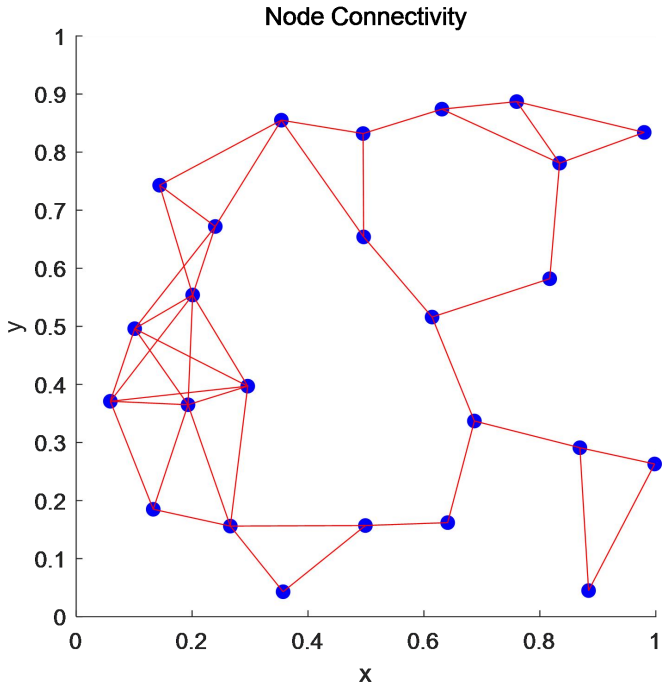
Lab3 Report

Shihao luo

Statement

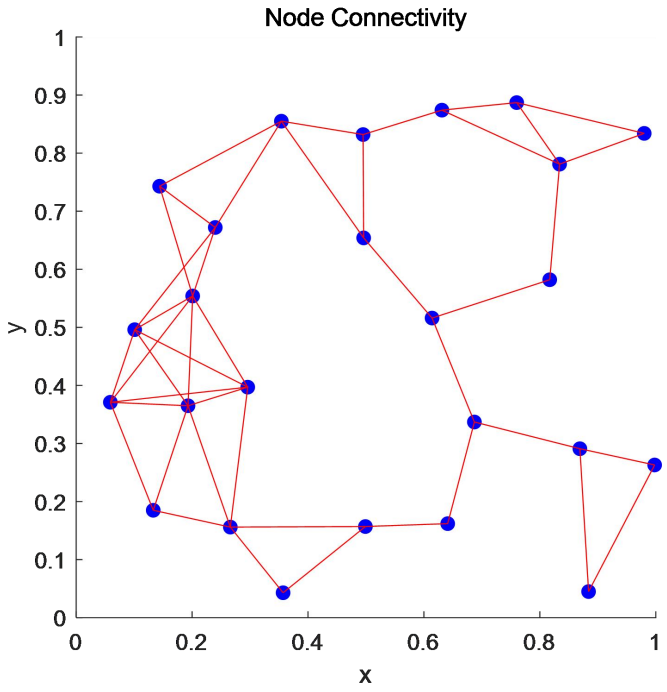
Complete the pseudo-randomly generated graph to ensure full connectivity. Quantitatively discuss the connectivity and coverage of this graph. Accurately complete a Voronoi tessellation diagram and a Delauney triangulation diagram for the graph. Assess the maximal breach distance and the maximal support distance of the graph.

Connectivity

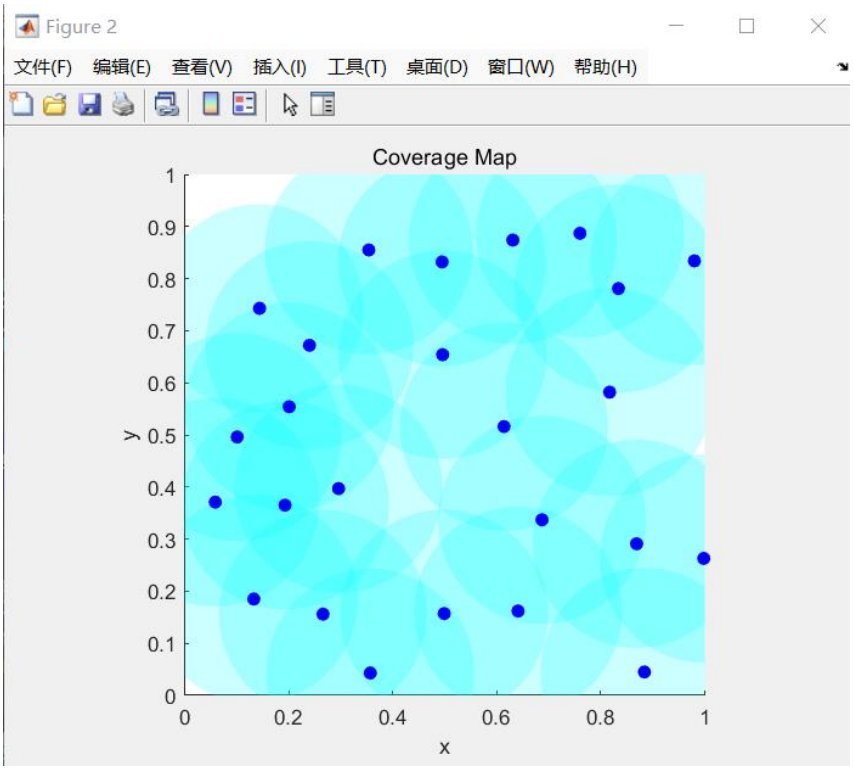


As we can see the above picture, the k-connectivity is 1, and the network will go dark for the area is on lower-right corner and upper right corner, which is very sparse and for the critical nodes like (0.6,0.3) seems a bridge distinct sub-network.If one of them fails, the network will split into disjoint area.

Accurate Diagram Of Node Connectivity

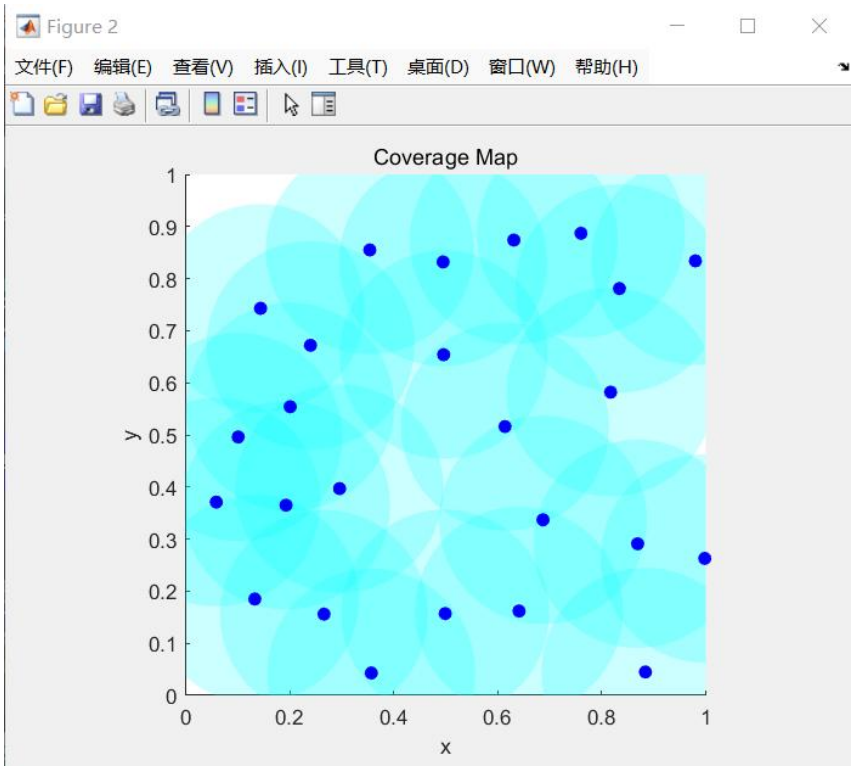


Coverage Of The Network



From the above picture,the central left corner has best coverage and the upper left and the bottom left corner have worst coverage. When it comes to the k-coverage is 0 because there is still no coverage in the upper left corner.

Accurate Diagram Of Coverage Map



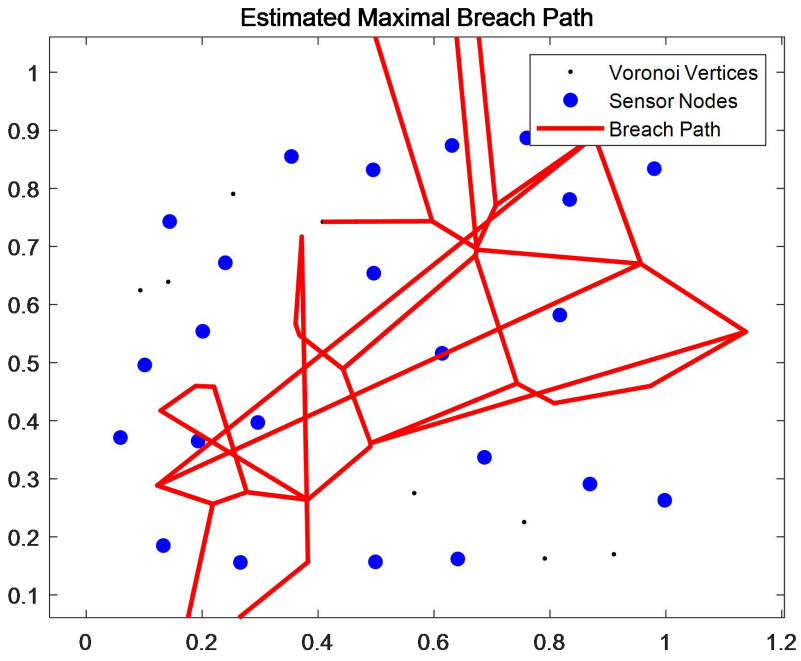
Calculate the maximal breach distance

Code:

```
DT = delaunayTriangulation(points);
[V, R] = voronoiDiagram(DT);

breachPoints = [];
for i = 1:length(R)
region = R{i};
if all(region > 0)
poly = V(region, :);
if any(abs(poly(:,1) - poly(:,2)) < 0.1)
breachPoints = [breachPoints; poly];
end
end
end
figure;
```

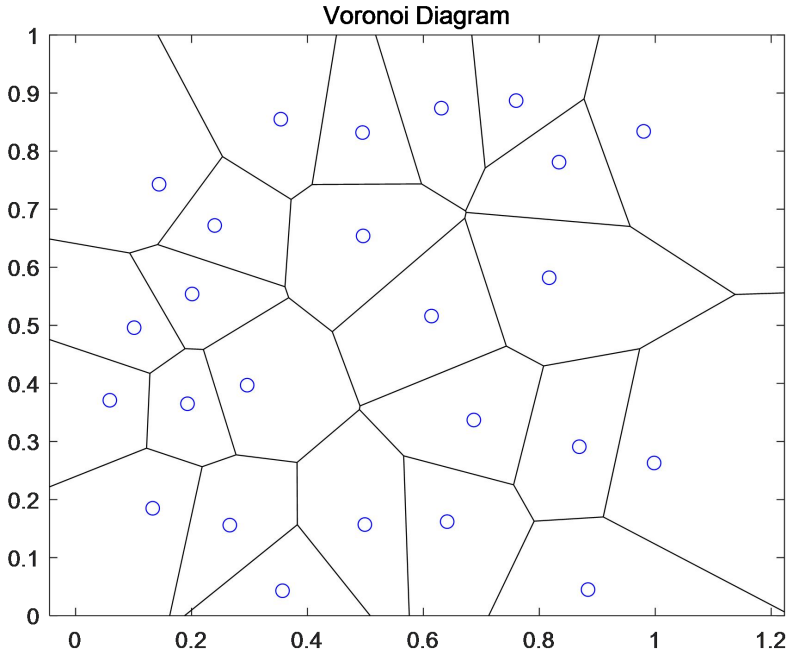
```
plot(V(:,1), V(:,2), 'k.'); hold on;
plot(points(:,1), points(:,2), 'bo', 'MarkerFaceColor', 'b');
plot(breachPoints(:,1), breachPoints(:,2), 'r-', 'LineWidth', 2);
xlim([0 1]); ylim([0 1]); axis equal;
title('Estimated Maximal Breach Path');
legend('Voronoi Vertices', 'Sensor Nodes', 'Breach Path');
minDists = zeros(size(breachPoints,1),1);
for i = 1:length(minDists)
dists = vecnorm(points - breachPoints(i,:), 2, 2);
minDists(i) = min(dists);
end
maximalBreach = min(minDists);
fprintf(' Estimated Maximum Breach Distance: %.4f\n', maximalBreach);
```

Picture:



Estimated Maximum Breach Distance: 0.0833

Accurate Diagram Of Voronoi tessellation



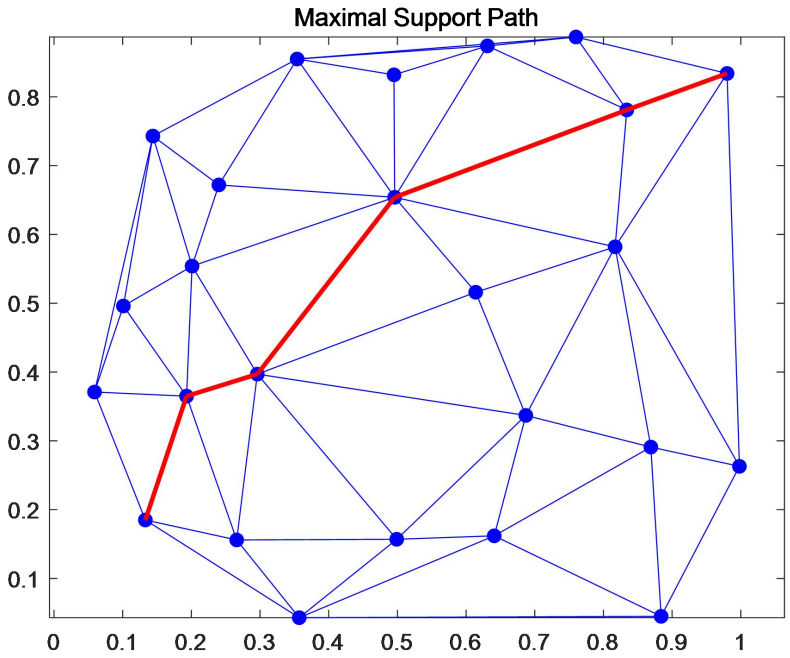
Calculate the maximal support distance

Code:

```
startNode = find(vecnorm(points - [0 0], 2, 2) == min(vecnorm(points - [0 0], 2, 2)), 1);
endNode = find(vecnorm(points - [1 1], 2, 2) == min(vecnorm(points - [1 1], 2, 2)), 1);

edges = dt.edges;
G = graph();

for i = 1:size(edges,1)
p1 = points(edges(i,1), :);
p2 = points(edges(i,2), :);
len = norm(p1 - p2);
G = addedge(G, edges(i,1), edges(i,2), len/2);
end

[spath, ~] = shortestpath(G, startNode, endNode);

figure;
```

```
triplot(dt); hold on;
plot(points(:,1), points(:,2), 'bo', 'MarkerFaceColor', 'b');
for k = 1:(length(spath)-1)
i = spath(k);
j = spath(k+1);
plot([x(i), x(j)], [y(i), y(j)], 'r-', 'LineWidth', 2);
end
title('Maximal Support Path');
axis equal;

pathLens = vecnorm(points(spath(2:end),:) - points(spath(1:end-1),:), 2, 2);
supportCost = max(pathLens)/2;
fprintf('  Maximal Support Distance: %.4f\n', supportCost);
```

Picture:



Maximal Support Distance: 0.1805

Accurate Diagram Of Delaunay Triangulation