# Solving the Close Enough Traveling Salesman Problem

William Mennell, University of Maryland
Bruce Golden, University of Maryland
Edward Wasil, American University

Presented at the 9th INFORMS Telecommunications Conference
College Park, MD, March 2008

# Outline

- Problem Definition
- Literature Review
- Example
- Description of Heuristic
- Results
- Future Directions

# Problem Definition

- Classical traveling salesman problem (TSP) visits every node in tour

- Close Enough TSP (CETSP) visits within distance *r* every node in tour
  - A disc of radius *r* implicitly surrounds every node
  - Tour must touch every disc

- Applications
  - Reconnaissance aircraft route planning
  - Ship tracking
  - Aerial forest fire detection
  - Robot monitoring of wireless sensor networks

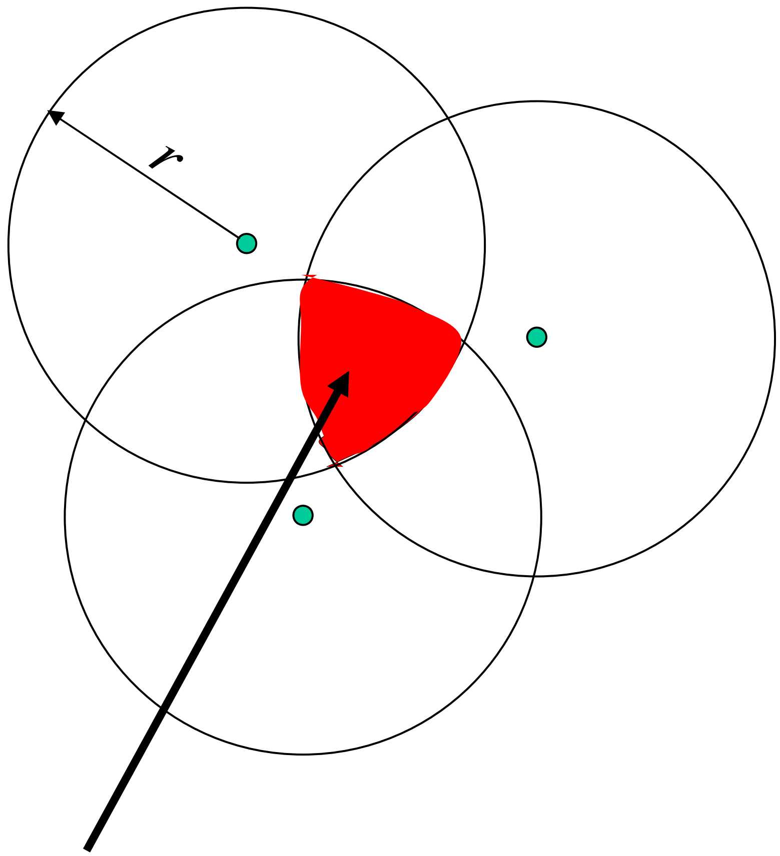
# Literature Review

- Close Enough Traveling Salesman Problem
  - Dong, Yang, Chen (2007)
  - Shuttleworth, Golden, Smith, and Wasil (2007)
  - Yuan, Orlowska, Sadiq (2007)
- Covering Tour Problem
  - Gendreau, Laporte, and Semet (1997)
  - Arkin and Hassin (1994)
- Generalized Traveling Salesman Problem
  - Fischetti, Gonzalez, and Toth (1997)
  - Silberholz and Golden (2007)
- TSP with Neighborhoods
  - Computational geometry literature seeking polynomial time approximation schemes
  - Mitchell (2007)
  - Dror, Efrat, Lubiw, and Mitchell (2003)

# Definitions

- Every node is surrounded by a disc of radius *r*
- *Steiner Zone (SZ) of degree k*

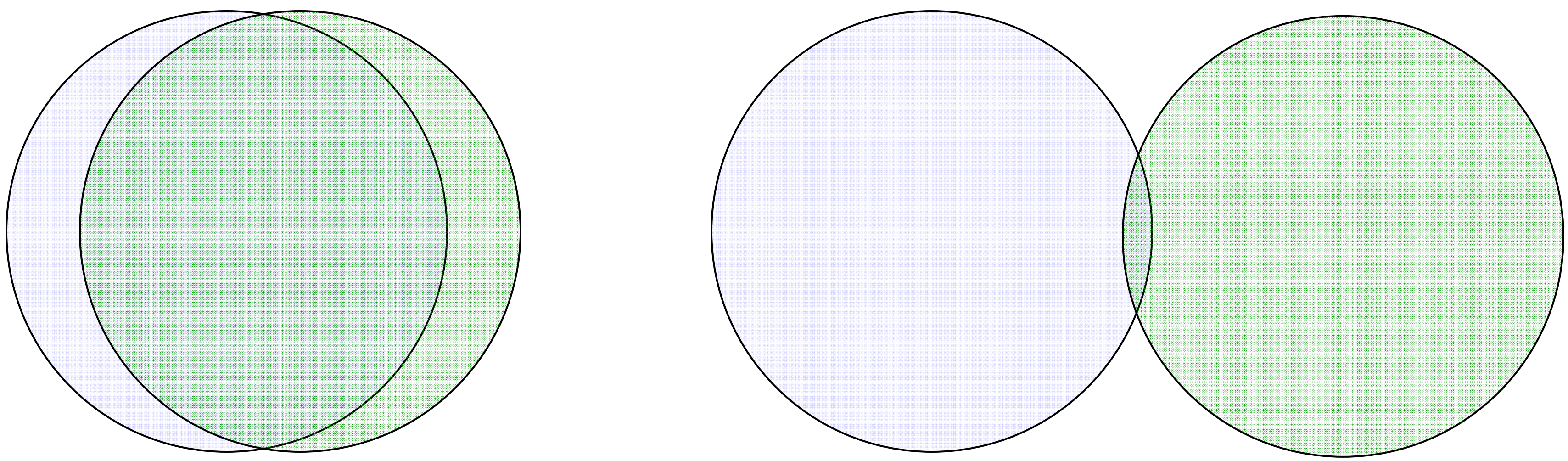
  -a region in which any point is simultaneously close enough to all k member nodes




SZ of degree 3

# Definitions

- Amount of overlap does not matter to us



- Both intersections are SZs of degree 2

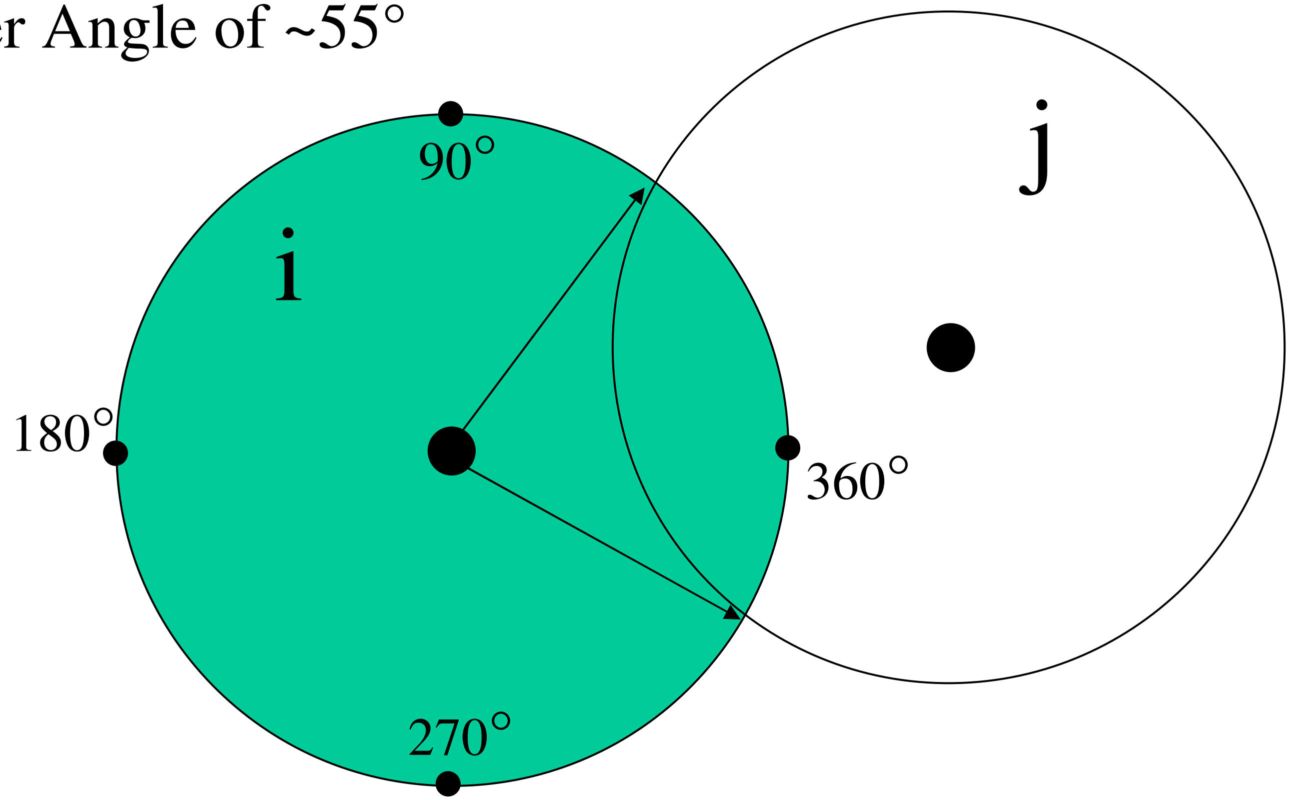
# Geometry

- The origin of a disc is its center point

- Given discs i and j, each of radius *r* and origins Or(i) and Or(j) respectively, it is straightforward to compute
  - The two points of intersection, if they exist
  - The angle of these intersections, with respect to Or(i)

# Geometry

- For example, Steiner Zone (ij) would be characterized by:
  - Origin of i
  - Lower Angle of ~330°
  - Upper Angle of ~55°

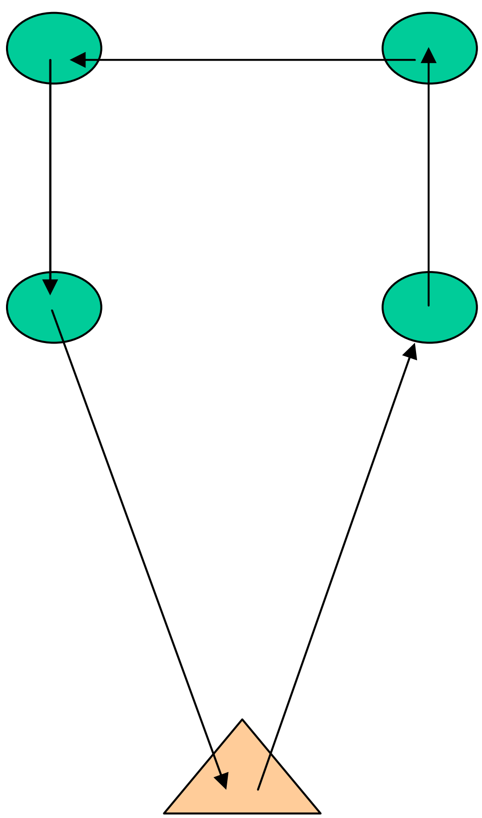


# Definition

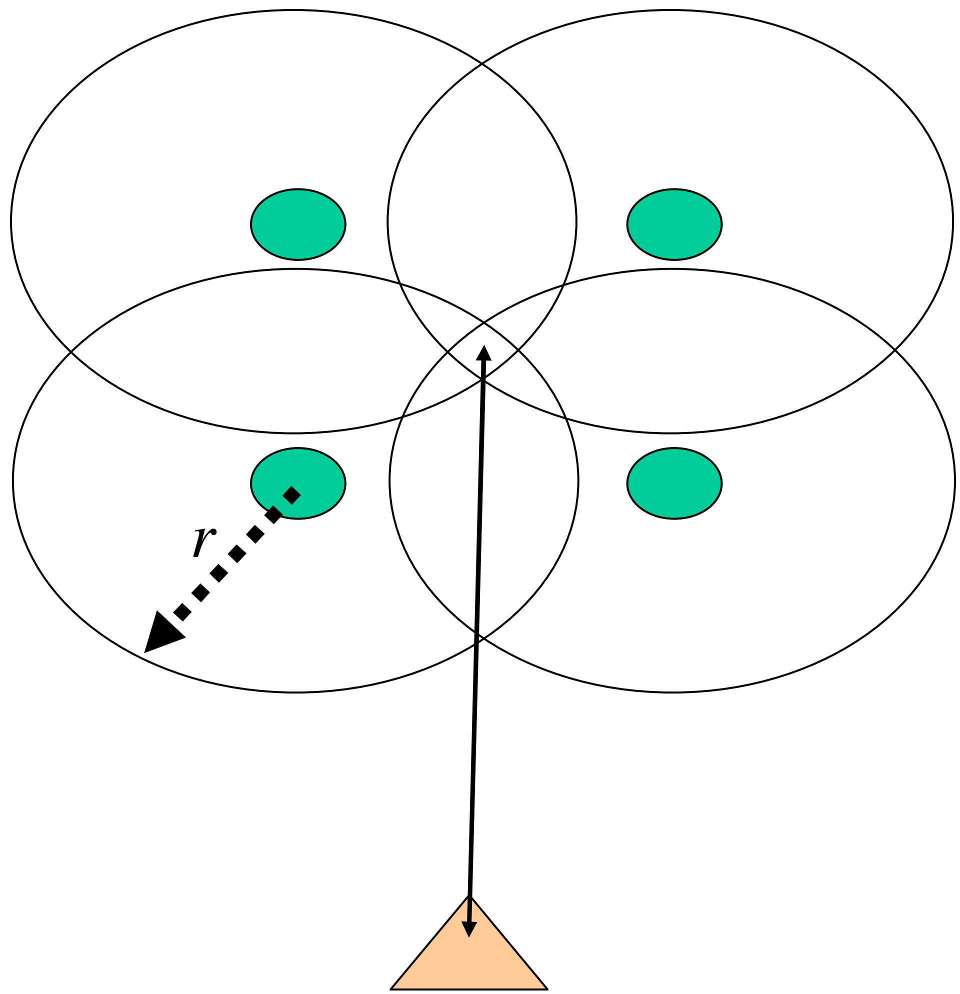
- As the radius grows, problem difficulty grows but we also expect shorter tours
  - More discs overlap → more complexity
  - More overlap → smaller number of points needed

- How then do we measure the potential for improvement?
  - Define the overlap for a problem as the ratio of *r* to the length of the smallest square surrounding all n discs

# Example



Classical TSP

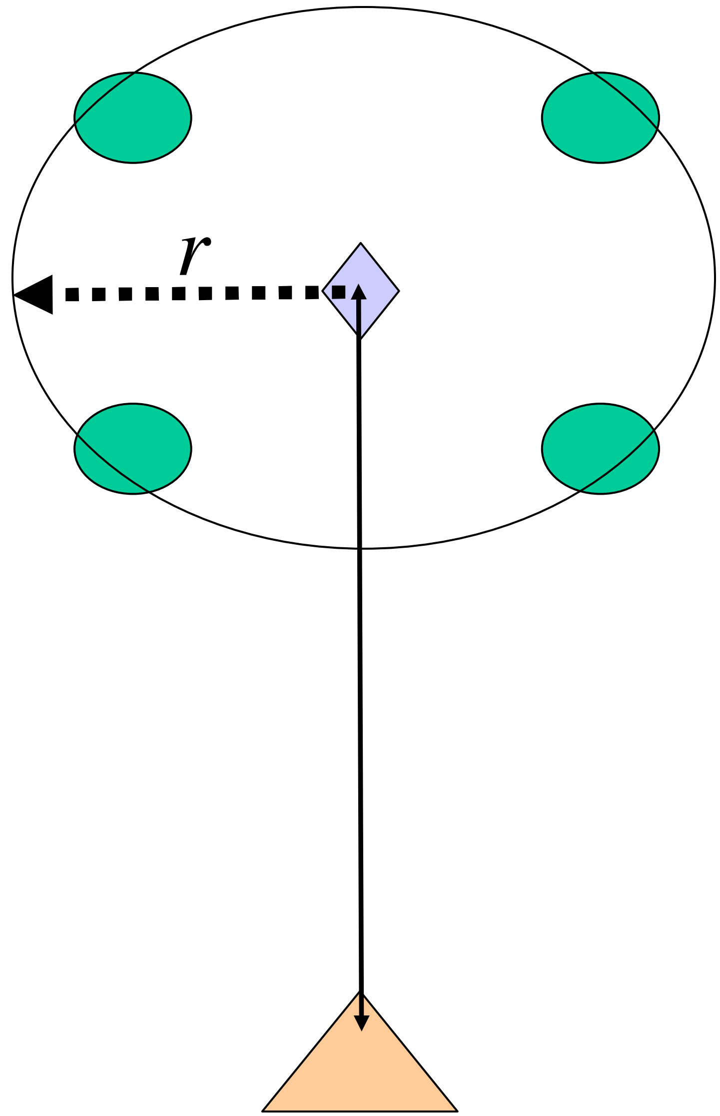



CETSP

# Example

*How is such a short tour feasible?*

-Every node is within *r* units of the visited location

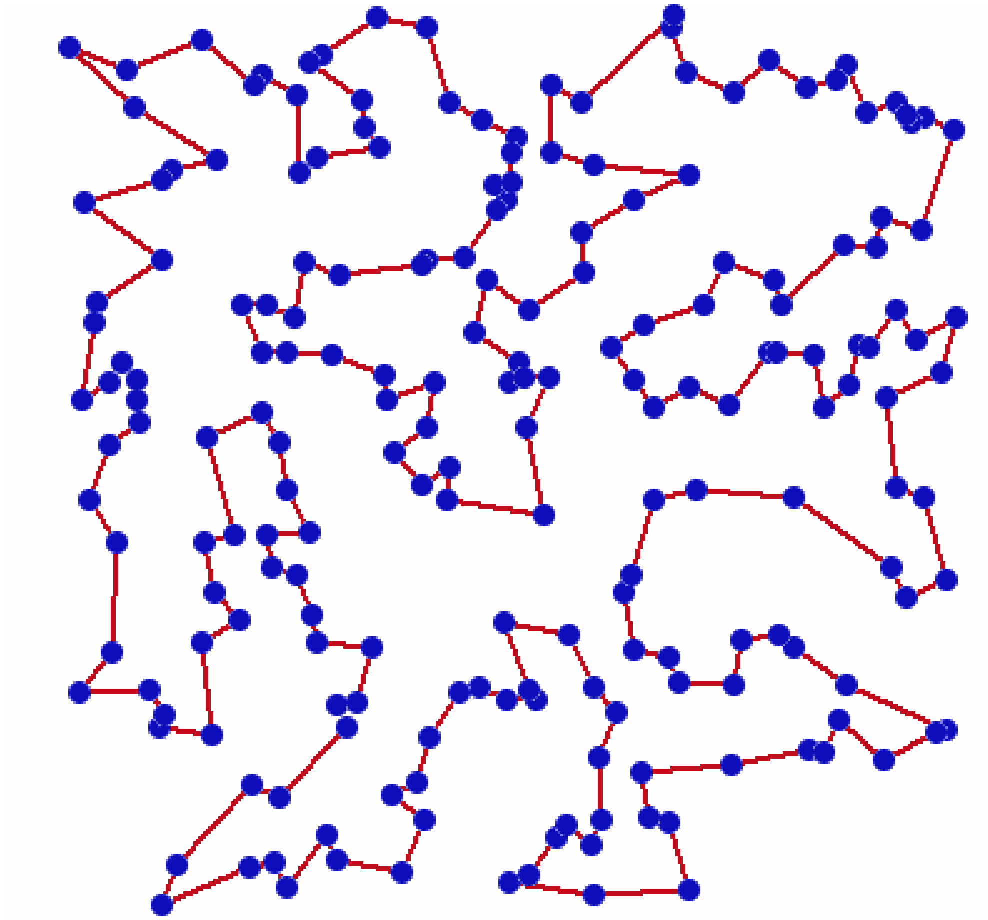


# The Steiner Zone Heuristic

- Three key steps
  - Graph reduction
  - Solve the underlying TSP
  - Optimize the TSP tour with respect to the Steiner Zones

# Before Step 1

**200-node problem**

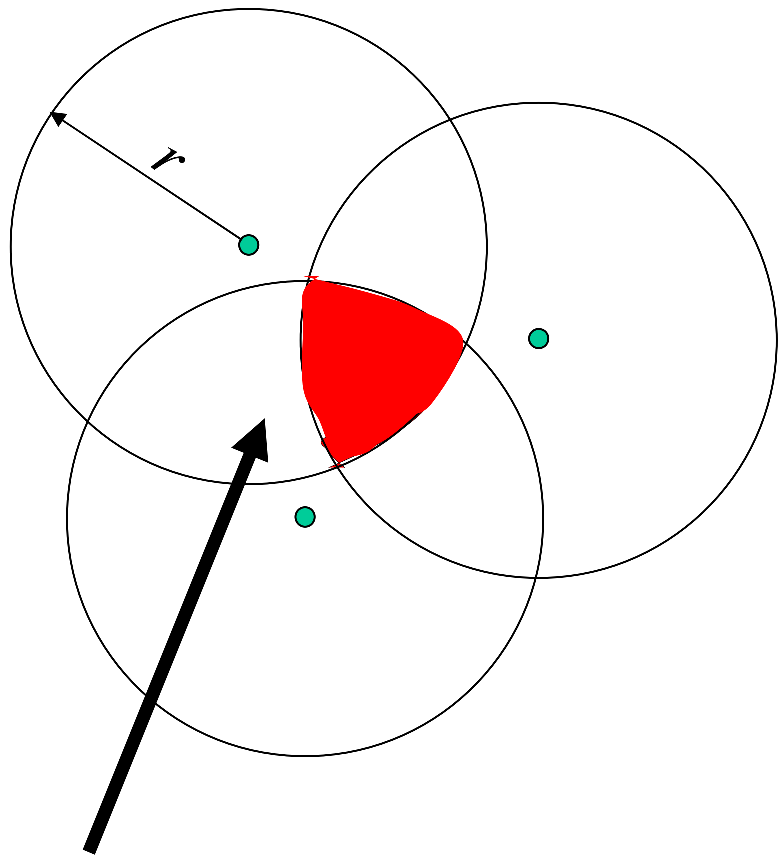


Optimal TSP solution: 1074.4

# 1. Graph Reduction

- Naïve reduction method
  - Step 1. Compute all SZs with respect to a chosen node in the graph
    - stored in descending order by degree
  - Step 2. Remove from the graph the member nodes of the highest degree SZ found
  - Step 3. If any nodes remain uncovered, go to Step 1
    - otherwise, we are done

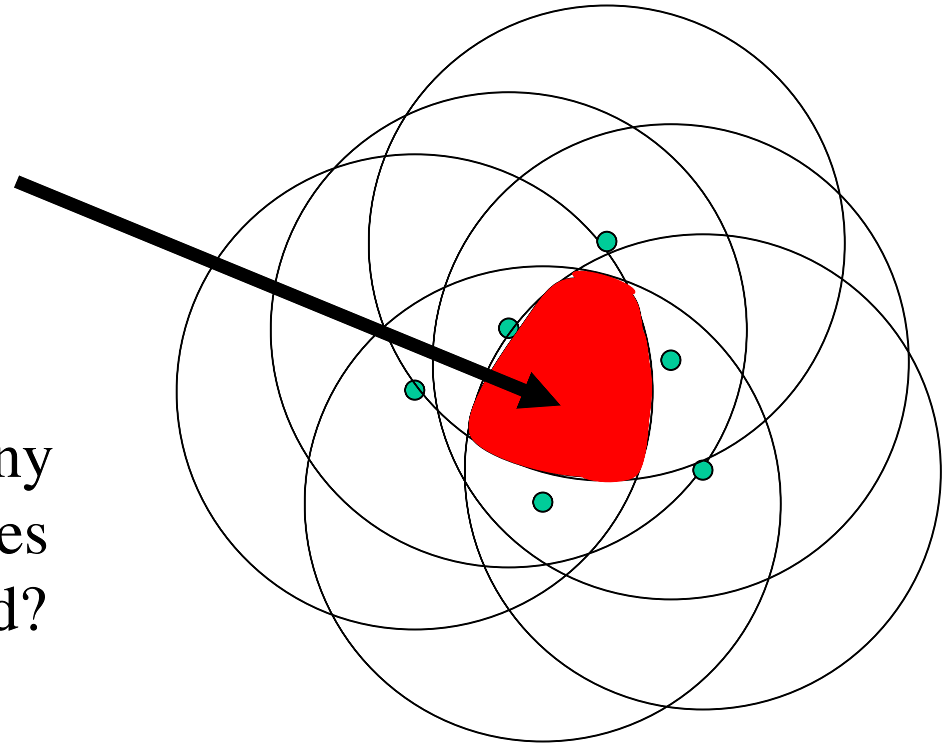
# Three Sub-Steiner Zones




Sub-Steiner
Zones for three
nodes

# Computing all Steiner Zones

Steiner Zone of
degree 6 – call it B



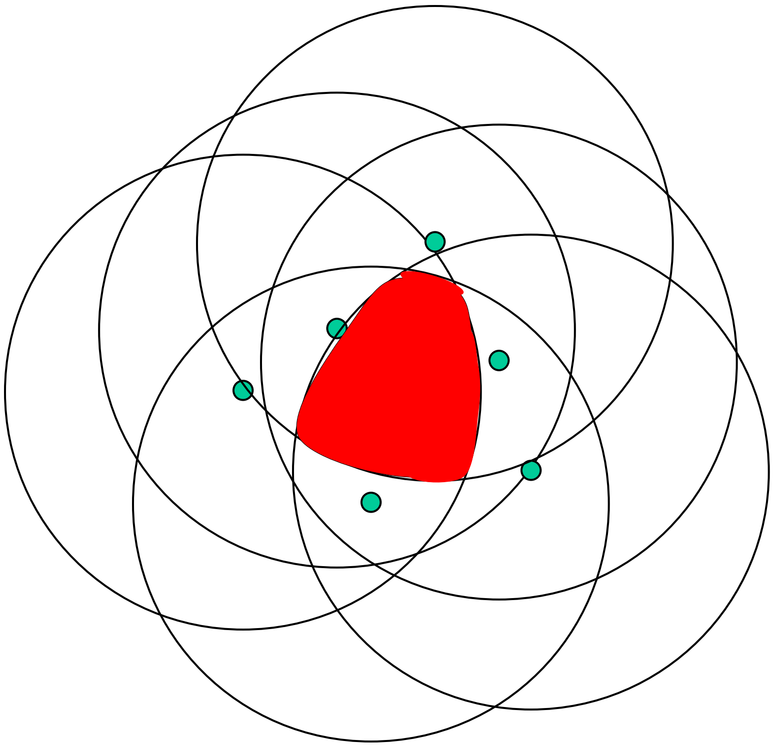
When B is created, how many potentially new Steiner Zones would the naïve method find?

| | |
|---|---|
| Degree 2: | 6 |
| Degree 3: | 10 |
| Degree 4: | 10 |
| Degree 5: | 4 |
| Degree 6: | 1 |
| Total: | 31 |

# Compute only the Necessary SZs

- Modify Step 1 of the naïve method
  - For each new SZ created, only add sub-Steiner Zones of degree 2 or 3

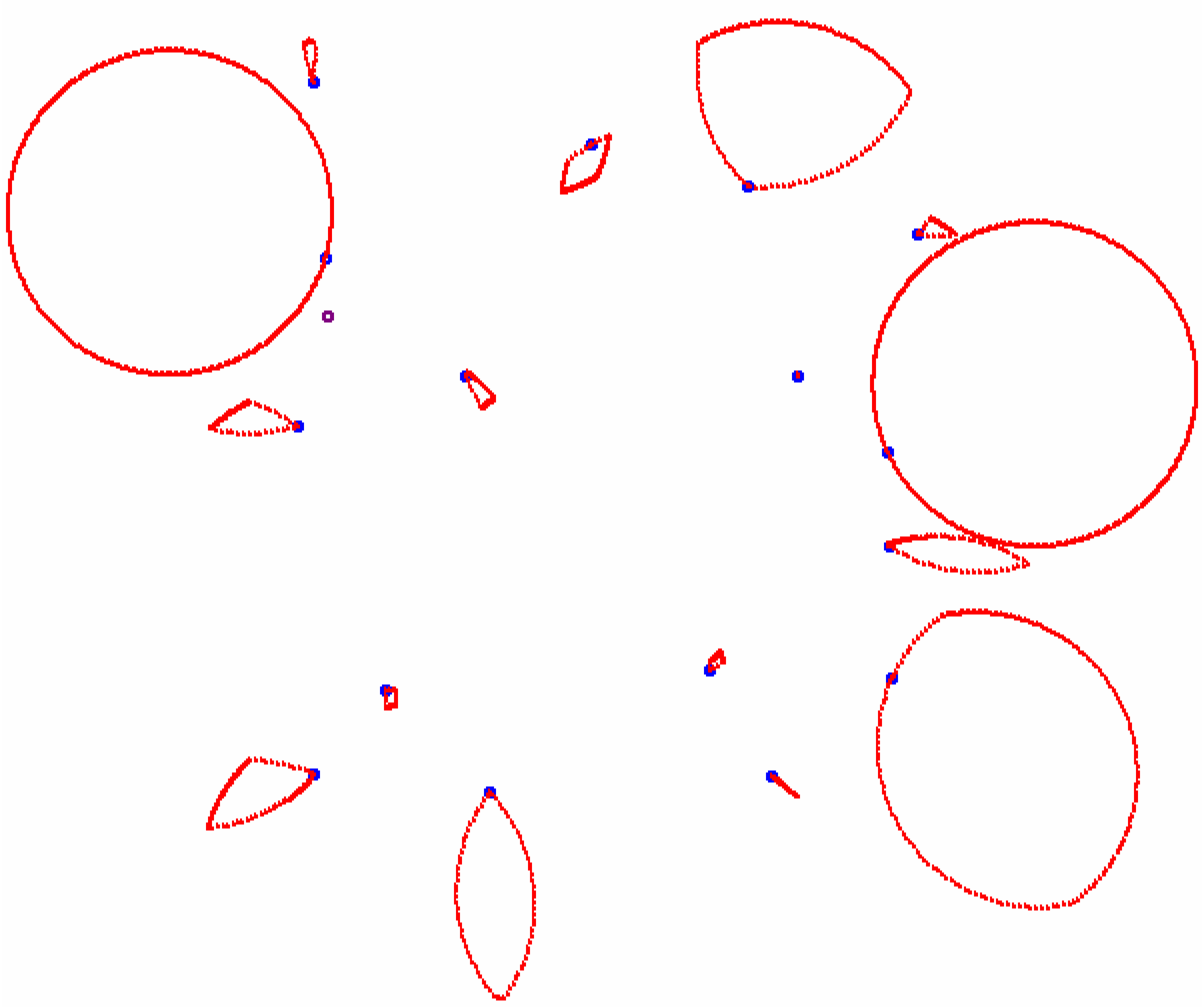
| | |
|---|---|
| Degree 2: | 6 |
| Degree 3: | 10 |
| Degree 6: | 1 |
| Total: | 17 |



The savings grows rapidly as the degree gets larger

# End of Step 1

**200-node problem reduced to 16 SZs**



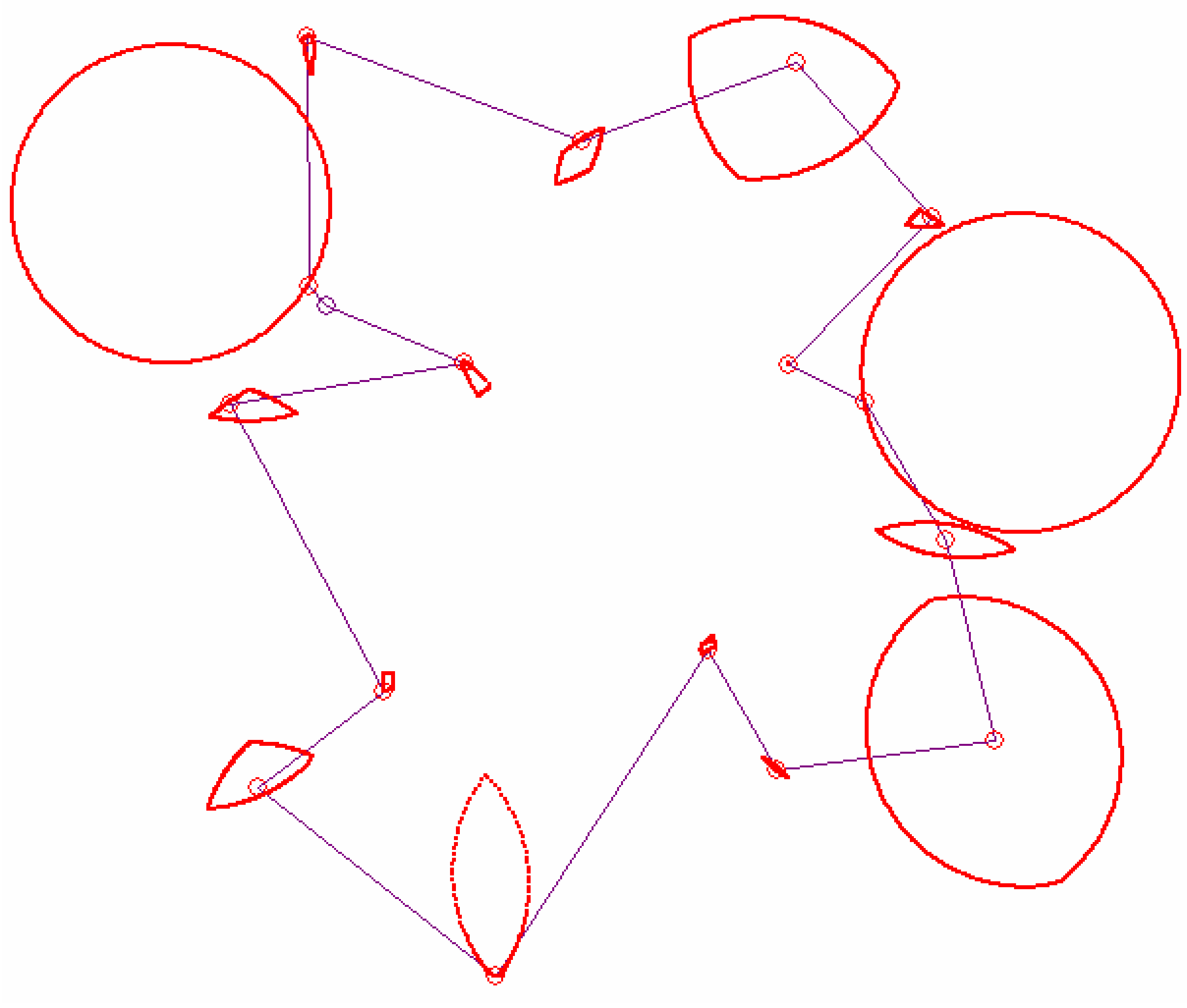
# 2. Solve the Underlying TSP

- Result of Step 1 is a set of disjoint convex regions

- How do we create a TSP on regions instead of nodes?
  - Choose a representative point for each region
    - Closest to depot, closest to centroid, randomly
  - Solve the TSP using those representative points
    - Use Concorde, Lin-Kernighan, …

# End of Step 2

**200-node problem with tour on representative points**



Distance: 404.8

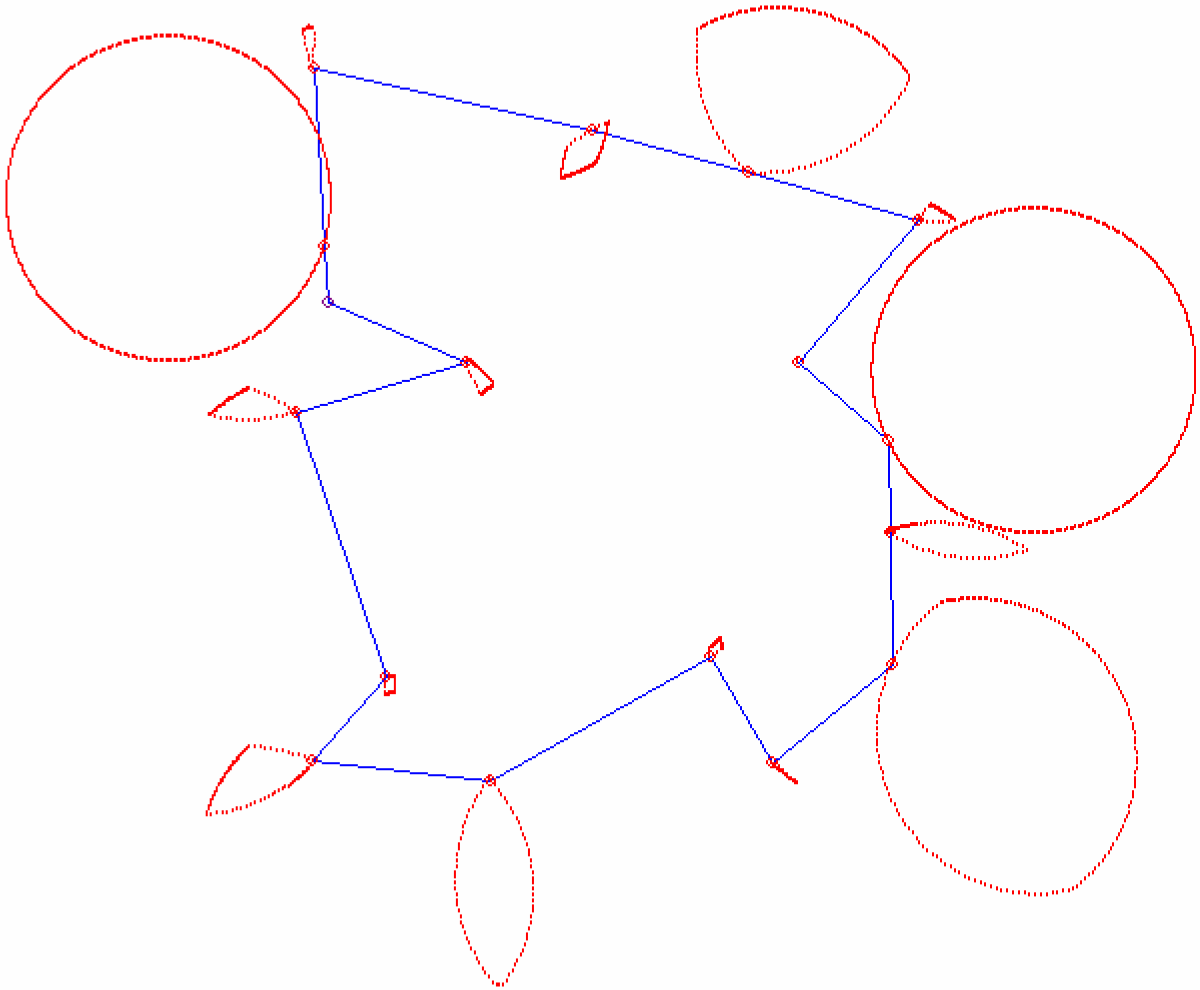
# 3. Optimize the TSP Tour

- The sequence of regions in the tour is fixed, but the location of each region's representative point is not

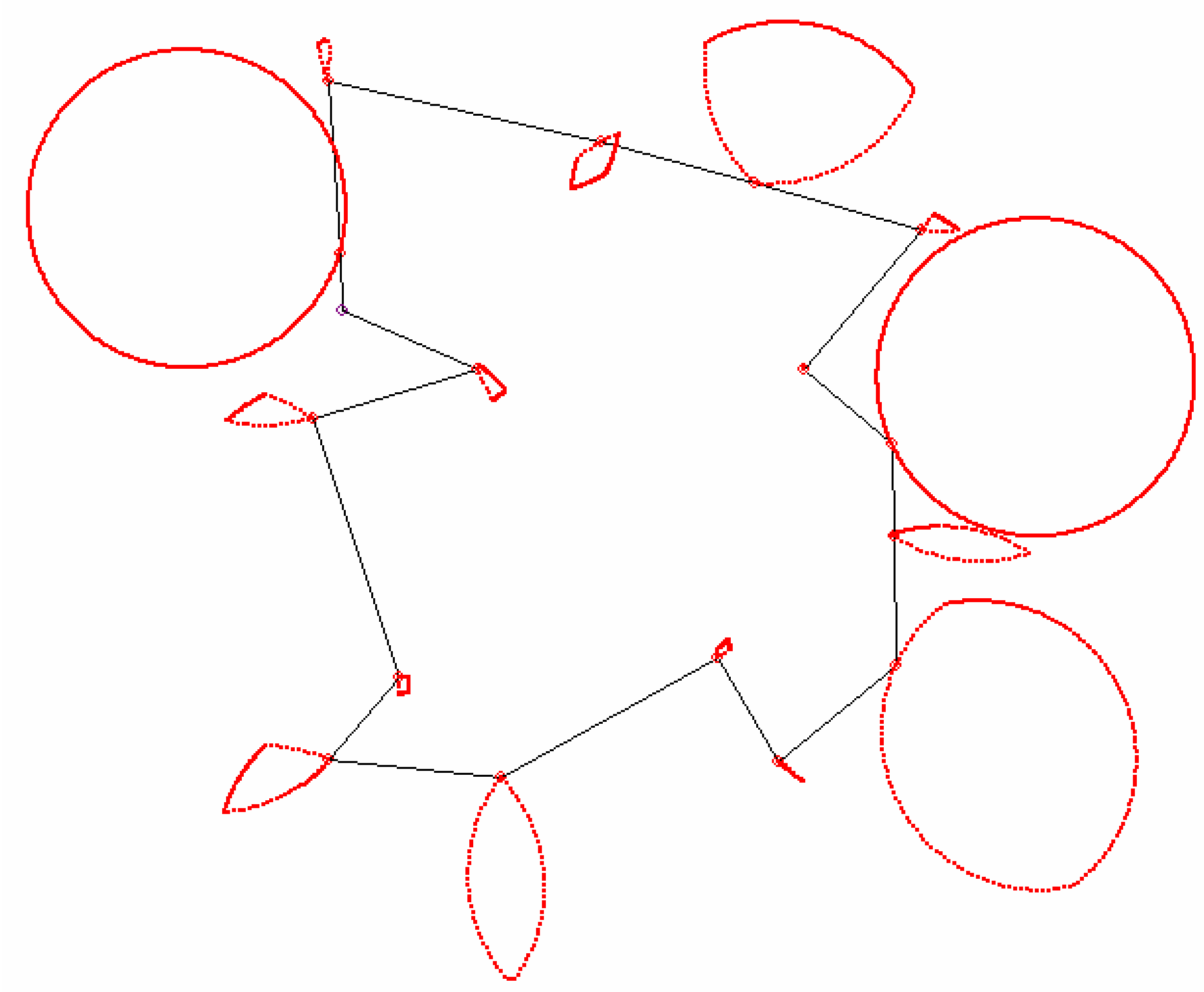
- Multiple ways to determine the locations
  1. Solve a Second Order Cone Program (SOCP)
     - Cplex can be used
  2. Approximate the SOCP – details in paper

# Iterative Approximation

**Iterative approximation of the TSP tour**



# Final Solution



Distance:  312.3

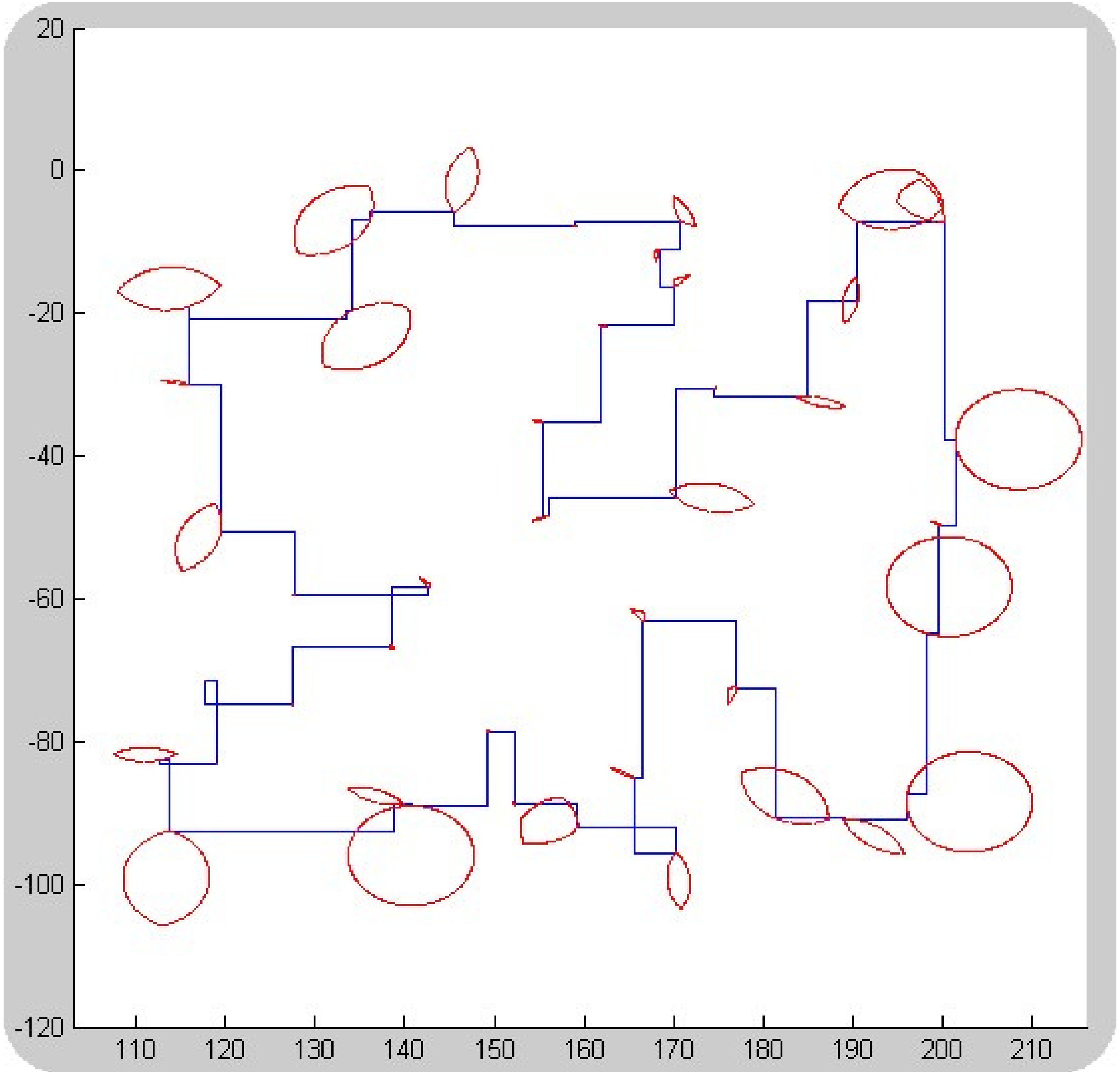
# Extensions

- Solve using Manhattan distance metric
  - Step I can use either L1 or L2 norm – results are essentially the same
  - Step II – Concorde can solve TSPs using L1 norm
  - Step III – both the heuristic and the QCP require only that edge distances are computed with L1 norm

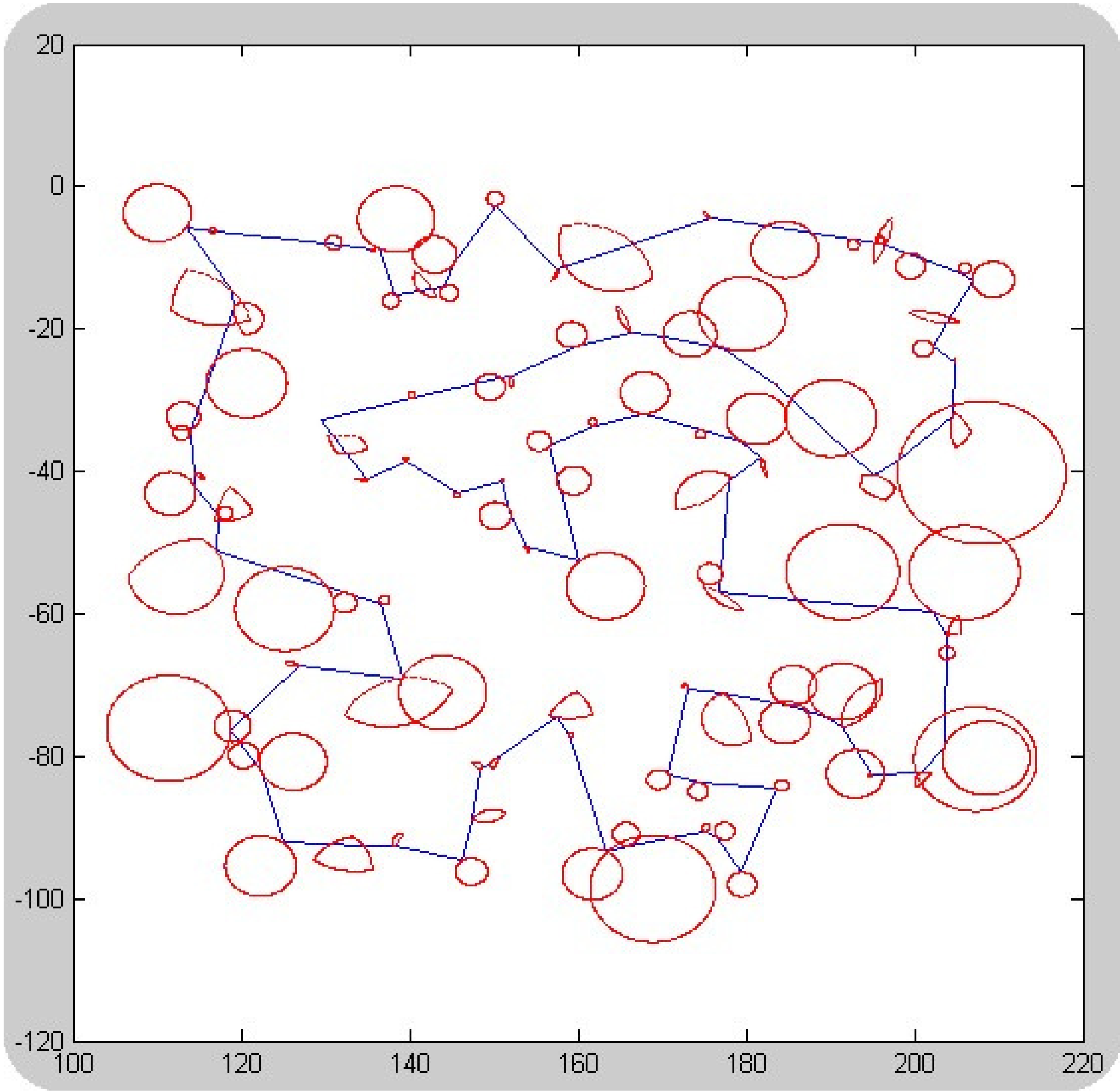
- Create test problems with an arbitrary radius for each disc
  - Radius generated uniformly from various ranges
  - Only change necessary is in Step I
    - Must handle the case when one disc lies completely inside another

- Solve in 3D
  - Two different Step I methods
  - Different ways to visualize the problem

# Manhattan Distance

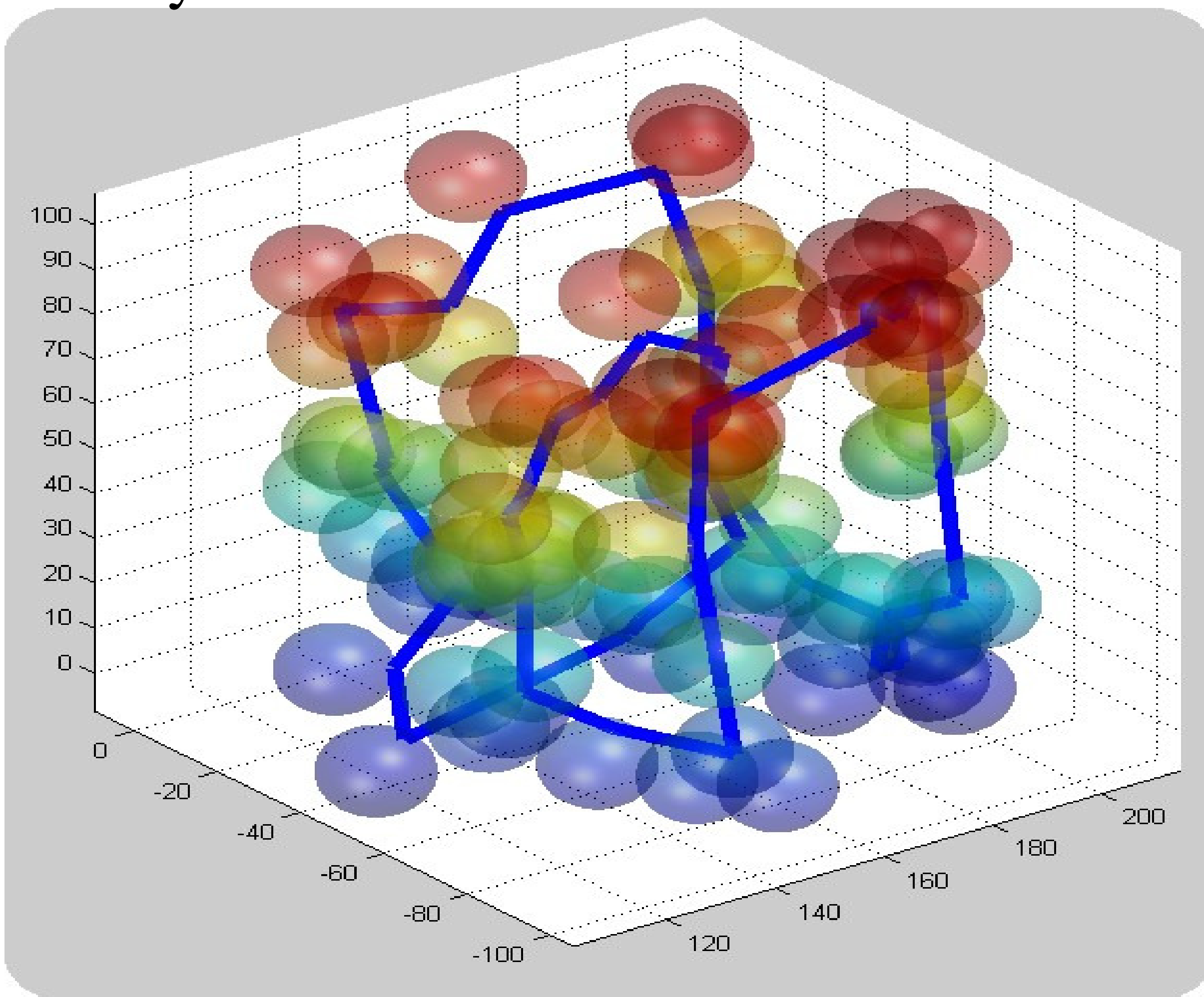


# Arbitrary Radius



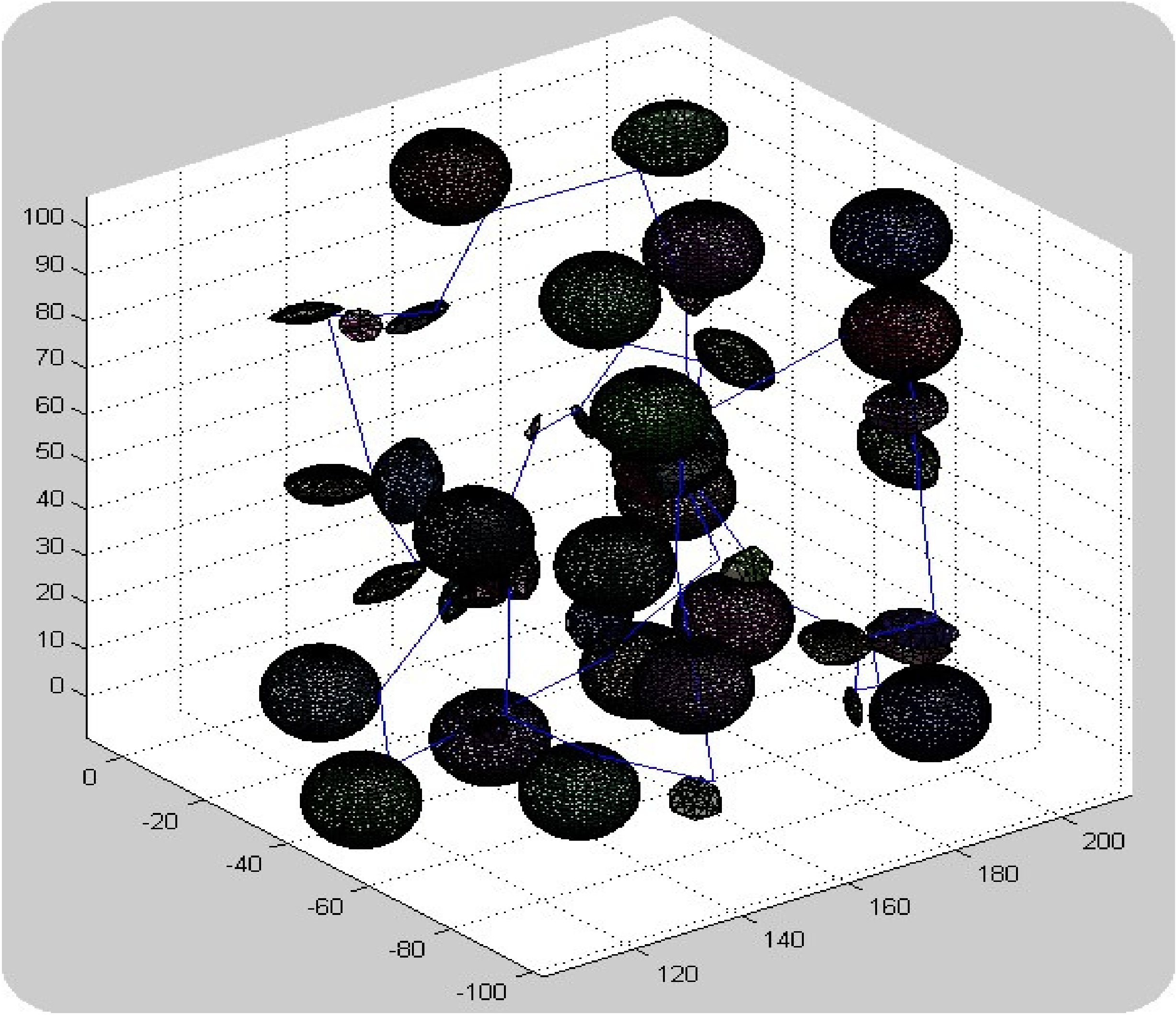
# 3D Visualizations

- Show the spheres surrounding all nodes
- Colors are only for differentiation



# 3D Visualizations

- Show polyhedral approximations of the convex hulls of  Steiner spheres – very slow



# Observations

- Greedy algorithm generates high-quality solutions

- Problems of 1000 nodes solved in less than 7 CPU seconds
  - Over half of the time spent solving the underlying TSP

- What benchmarks exist for computational comparison?
  - Genetic algorithm code of Silberholz and Golden (2007) produced excellent results on generic GTSP instances
  - No Step I → Lin-Kernighan + Step III
  - Greedy
    - Must know all Steiner Zones

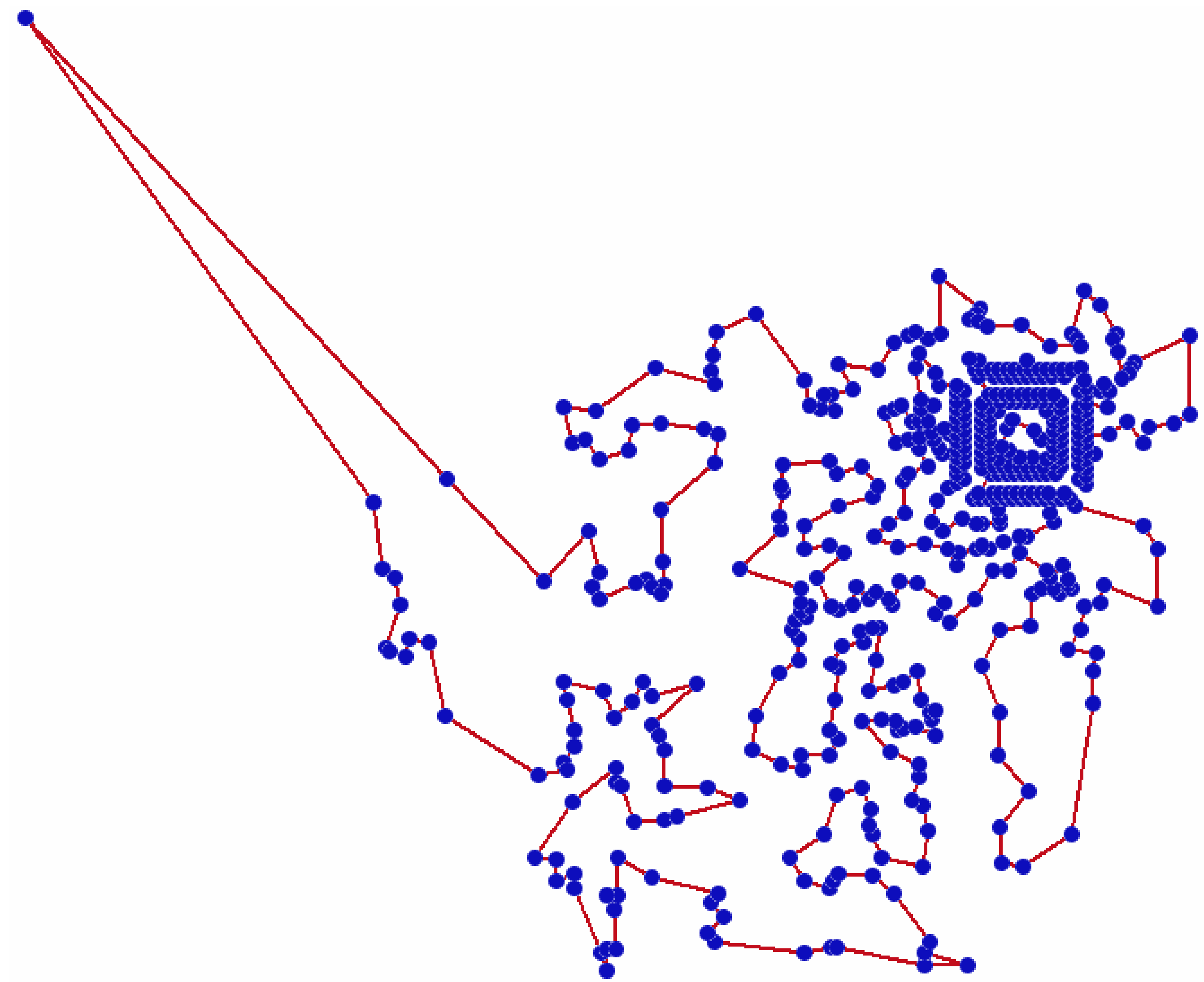
# Results

- Moderate and high-overlap problems
  - SZ heuristic outperforms GA in solution quality
    - Up to 150% better

- Low overlap problems
  - GA usually outperforms SZ by 5% - 7%
  - Steiner Zones were often half the highest degree found by our heuristic

- GA takes much more time and lots of memory
  - Must store a full or partial distance matrix
  - Few hundred nodes → days
  - 1000 nodes → weeks or months

# Results



d493.tsp from TSPLIB optimal solution: 350.19

# Results

| | | | | |
|---|---|---|---|---|
| | d493 | | | |
| | radius | 0.74904 | 3.7452 | 11.2356 |
| | overlap | 0.02 | 0.1 | 0.3 |
| GTSP-GA-24 | | 206.77 | 133.81 | 148.97 |
| continuous solution | | 204.49 | 112.55 | 82.83 |
| | CPU sec. | 309067 | 44862 | 59105 |
| SZ heuristic | | 215.46 | 106.24 | 71.16 |
| | CPU sec. | 4 | 1 | 1 |

- GTSP-GA-24 → each disc is approximated by 24 equally spaced points.
  - Solved using the genetic algorithm
  - Apply Step III to discrete GTSP solution to make it continuous

# Conclusions

- The Steiner Zone heuristic:
  - Produces consistently high-quality solutions quickly
  - Simple in concept
  - Readily adaptable to problem variants
- We have prepared the first benchmark instances for this problem
- Cannot conclusively judge any CETSP heuristic until optimal solutions are known

# Future Research Directions

- Generate SZ solutions that are better than the GA solutions for low-overlap problems without sacrificing speed
  - Local search using a pool of SZs
  - Use a GA to pick the best set of Steiner Zones
- Devise good lower bounding procedures