```
Method: Ant Colony Optimization Method
Shortest Distance: 1,79529
Shortest Path: [1, 4, 9, 8, 11, 10, 7, 6, 5, 3, 2, 1]
Time it takes to find the shortest path: 0,06 seconds.
```

Ant Colony Optimization Output-1

```
Method: Ant Colony Optimization Method
Shortest Distance: 2,93588
Shortest Path: [1, 2, 3, 12, 4, 5, 6, 9, 10, 11, 7, 8, 1]
Time it takes to find the shortest path: 0,07 seconds.
```

Ant Colony Optimization Output-2

```
Method: Ant Colony Optimization Method
Shortest Distance: 3,80292
Shortest Path: [1, 2, 4, 3, 5, 9, 8, 13, 12, 11, 10, 7, 6, 1]
Time it takes to find the shortest path: 0,07 seconds.
```

Ant Colony Optimization Output-3

```
Method: Ant Colony Optimization Method
Shortest Distance: 3,71091
Shortest Path: [1, 2, 3, 13, 10, 14, 11, 9, 5, 6, 4, 8, 7, 12, 1]
Time it takes to find the shortest path: 0,09 seconds.
```

Ant Colony Optimization Output-4

```
Method: Ant Colony Optimization Method
Shortest Distance: 4,77101
Shortest Path: [1, 27, 25, 26, 10, 9, 17, 11, 6, 7, 28, 29, 3, 2, 4, 30, 5, 13, 8, 12, 14, 20, 19, 18, 21, 24, 15, 16, 23, 22, 1]
Time it takes to find the shortest path: 0,24 seconds.
```

Ant Colony Optimization Output-5

Best Ant Colony Hyperparameters

Iteration Count = 100

Ant Count Per Iteration = 50

Degradation Factor = 0.92 (0.90 for input 5)

Alpha = 1.8

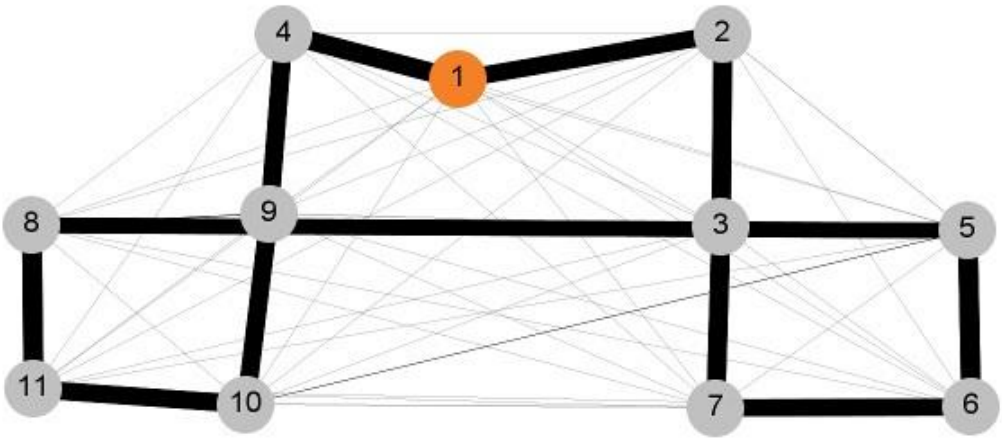
Beta = 2.5

Q = 0.0001

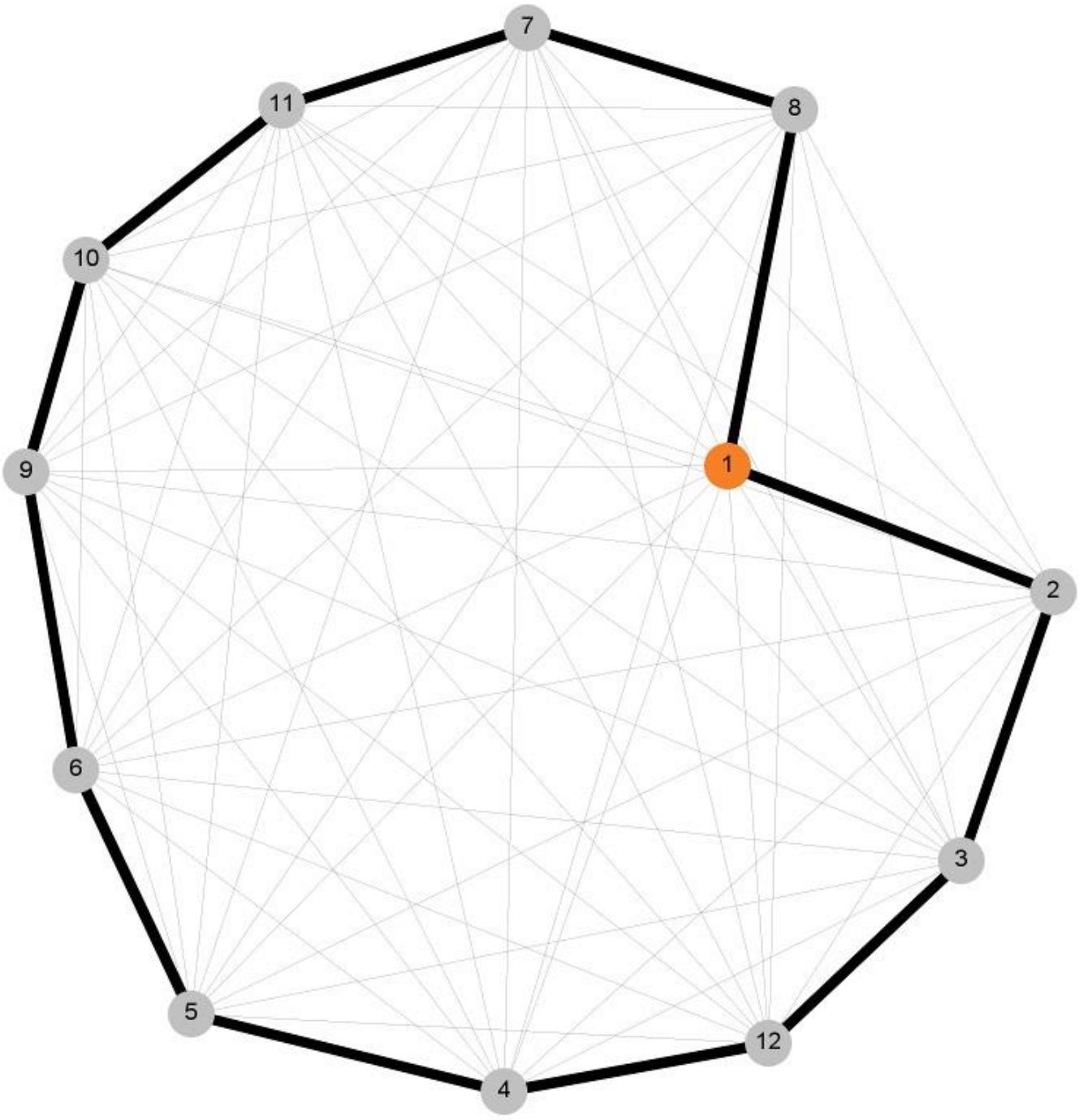
Initial Pheromone Density = 0.1

Brute-Force vs Ant Colony Table

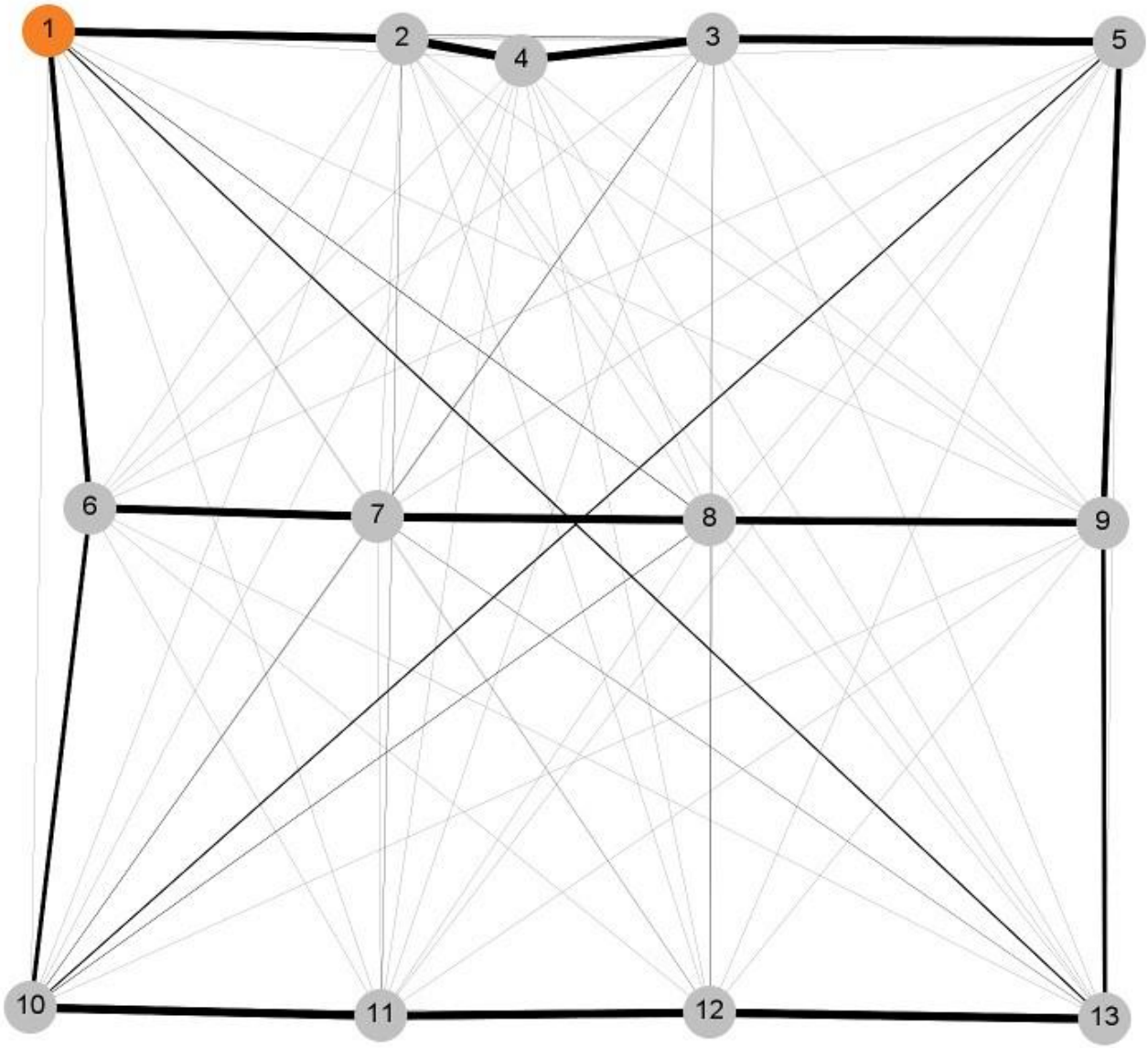
Input File	Number of Houses + Migros	Brute-Force Time (seconds)	Ant Colony Time (seconds)	Speed Up Factor
Input 1	11	0.10 (Distance: 1.79529)	0.06 (Distance: 1.79529)	1.67
Input 2	12	0.40 (Distance: 2.93588)	0.07 (Distance: 2.93588)	5.71
Input 3	13	3.04 (Distance: 3.80292)	0.07 (Distance: 3.80292)	43.43
Input 4	14	32.04 (Distance: 3.71091)	0.09 (Distance: 3.71091)	356
Input 5	30	Too Long to Compute	0.24 (Distance: 4.77101)	x



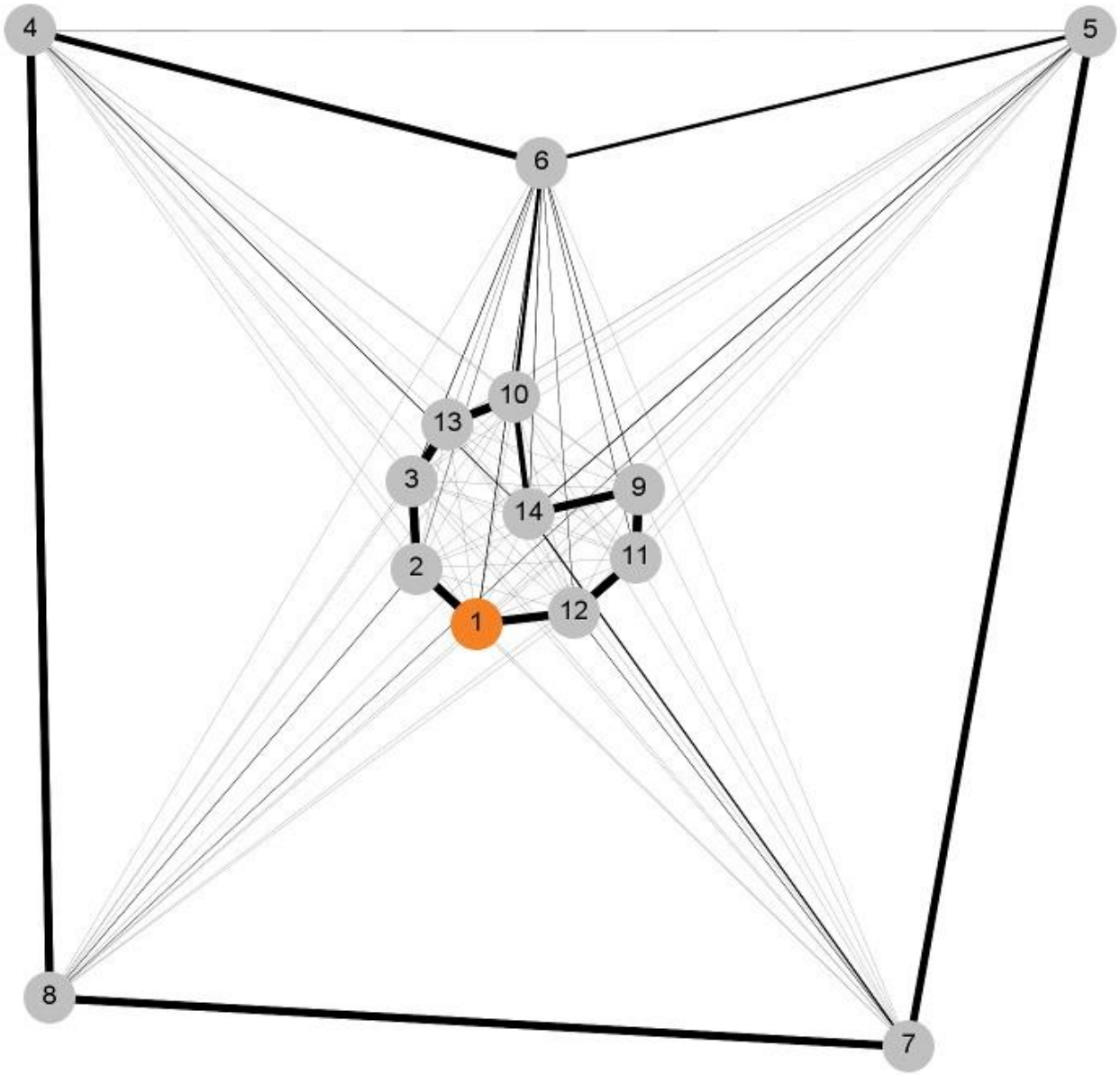
Ant Colony Optimization Input 1 Pheromone Map



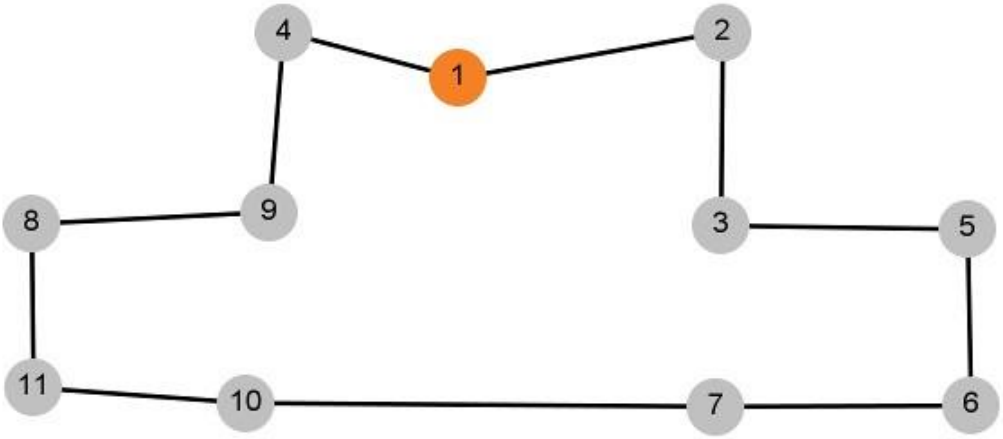
Ant Colony Optimization Input 2 Pheromone Map



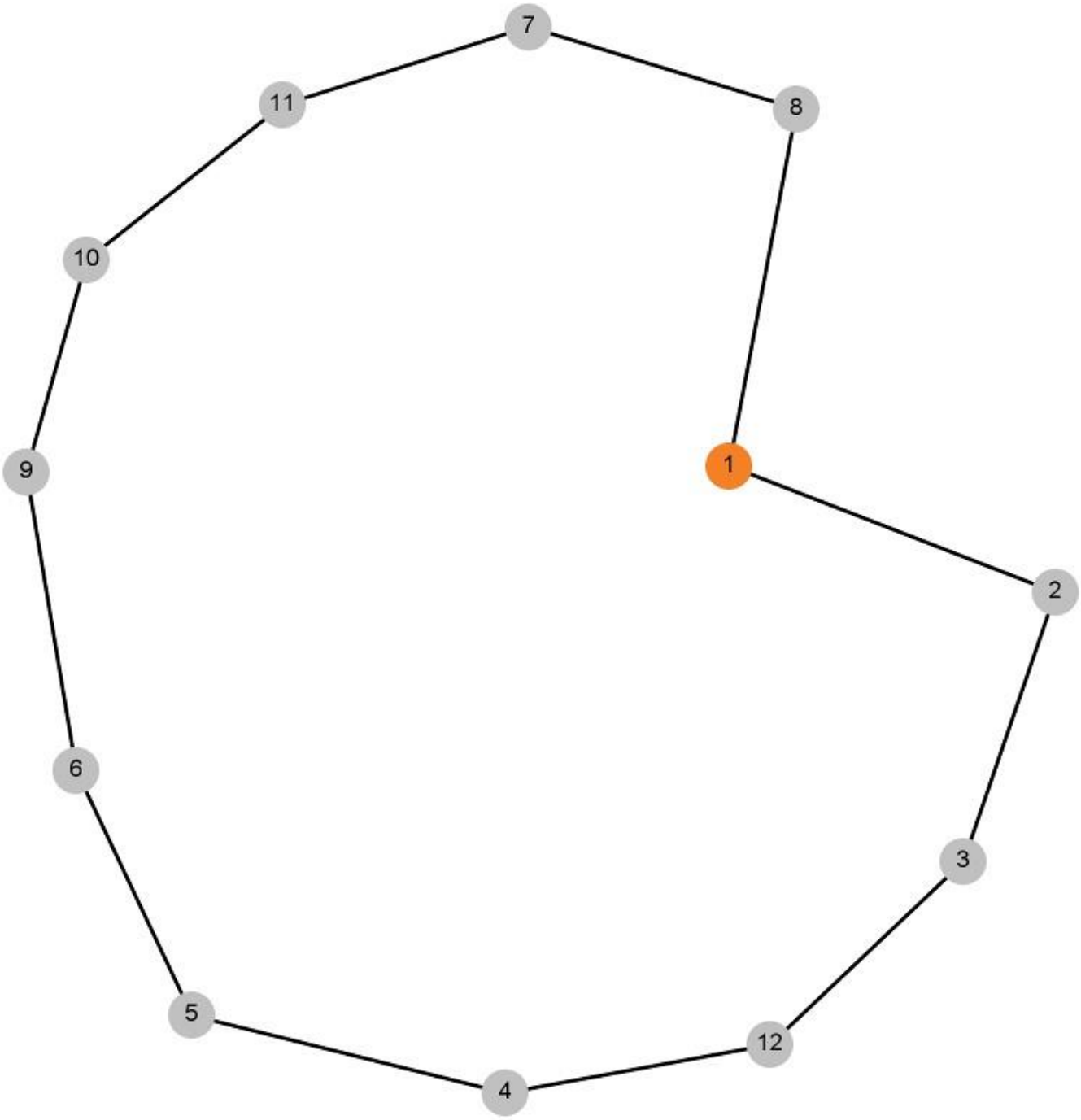
Ant Colony Optimization Input 3 Pheromone Map



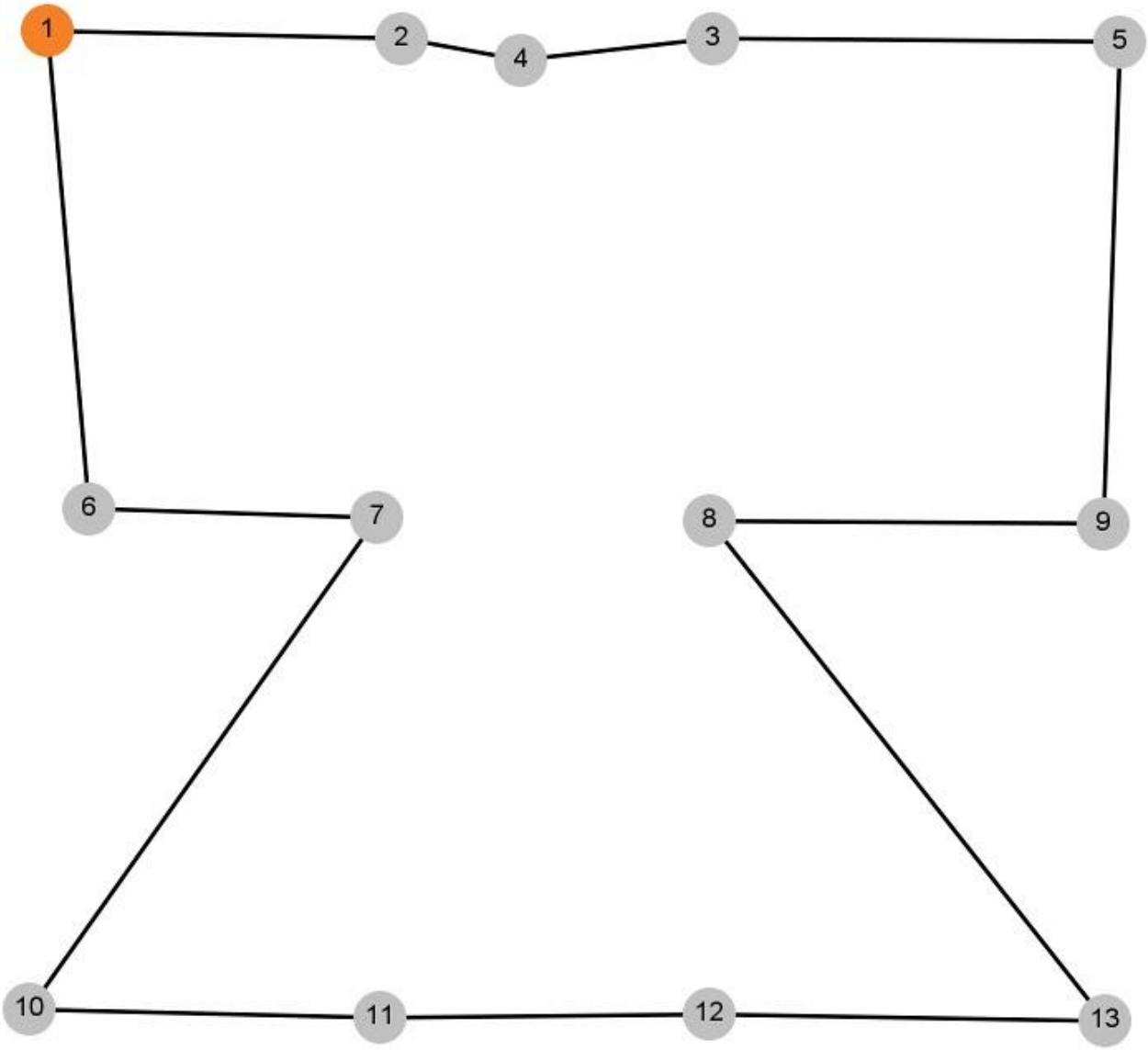
Ant Colony Optimization Input 4 Pheromone Map



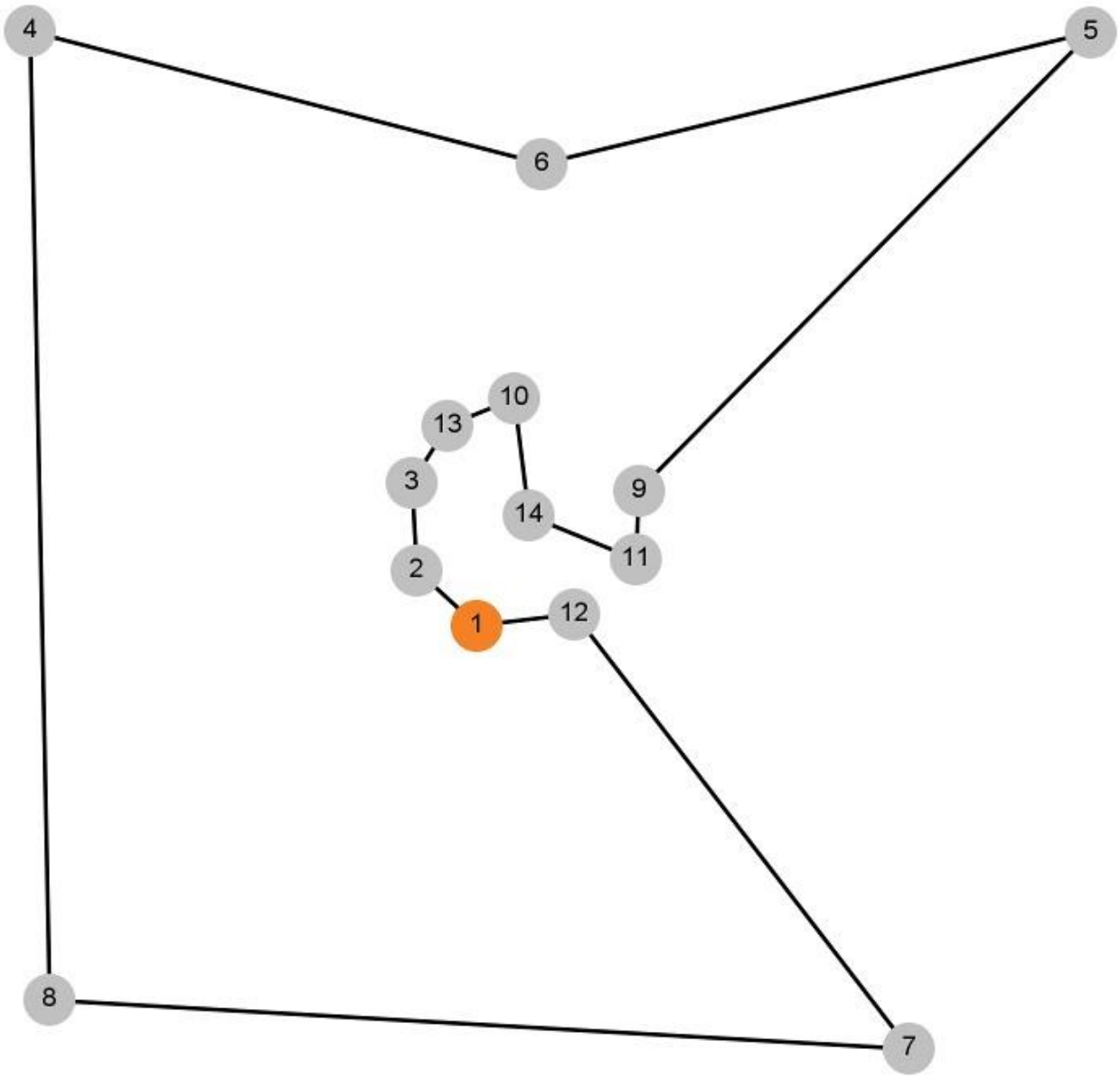
Ant Colony Optimization Input 1 Shortest Route Map



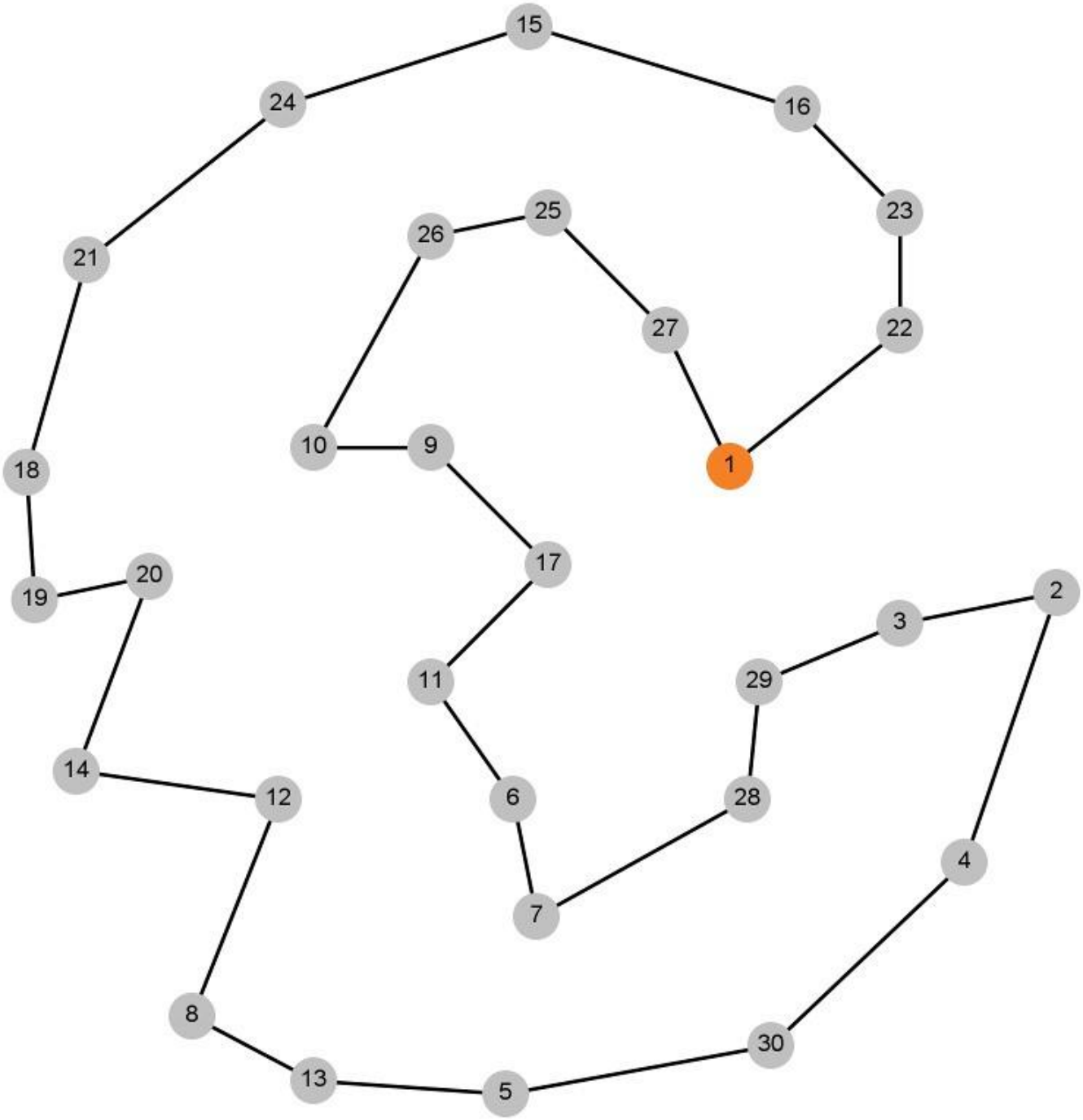
Ant Colony Optimization Input 2 Shortest Route Map



Ant Colony Optimization Input 3 Shortest Route Map



Ant Colony Optimization Input 4 Shortest Route Map



Ant Colony Optimization Input 5 Shortest Route Map

```
Method: Brute-Force Method
Shortest Distance: 1,79529
Shortest Path: [1, 2, 3, 5, 6, 7, 10, 11, 8, 9, 4, 1]
Time it takes to find the shortest path: 0,10 seconds.
```

Brute Force Output-1

```
Method: Brute-Force Method
Shortest Distance: 2,93588
Shortest Path: [1, 2, 3, 12, 4, 5, 6, 9, 10, 11, 7, 8, 1]
Time it takes to find the shortest path: 0,40 seconds.
```

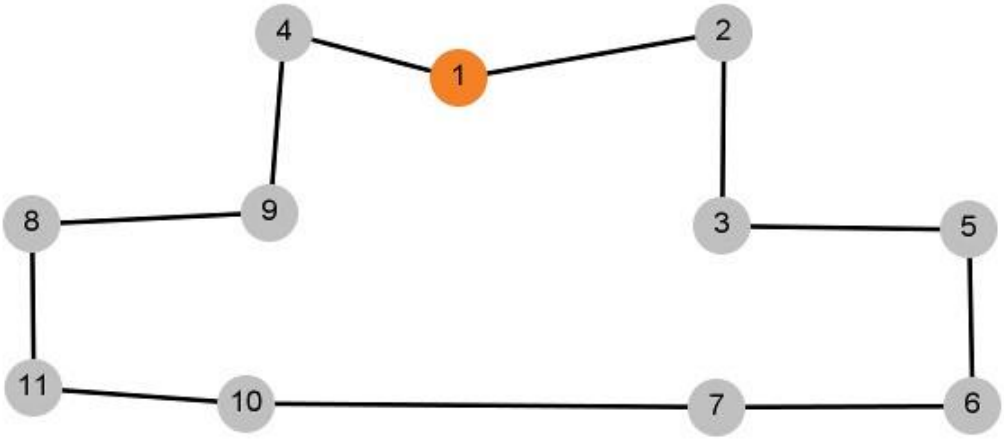
Brute Force Output-2

```
Method: Brute-Force Method
Shortest Distance: 3,80292
Shortest Path: [1, 2, 4, 3, 5, 9, 8, 13, 12, 11, 10, 7, 6, 1]
Time it takes to find the shortest path: 3,04 seconds.
```

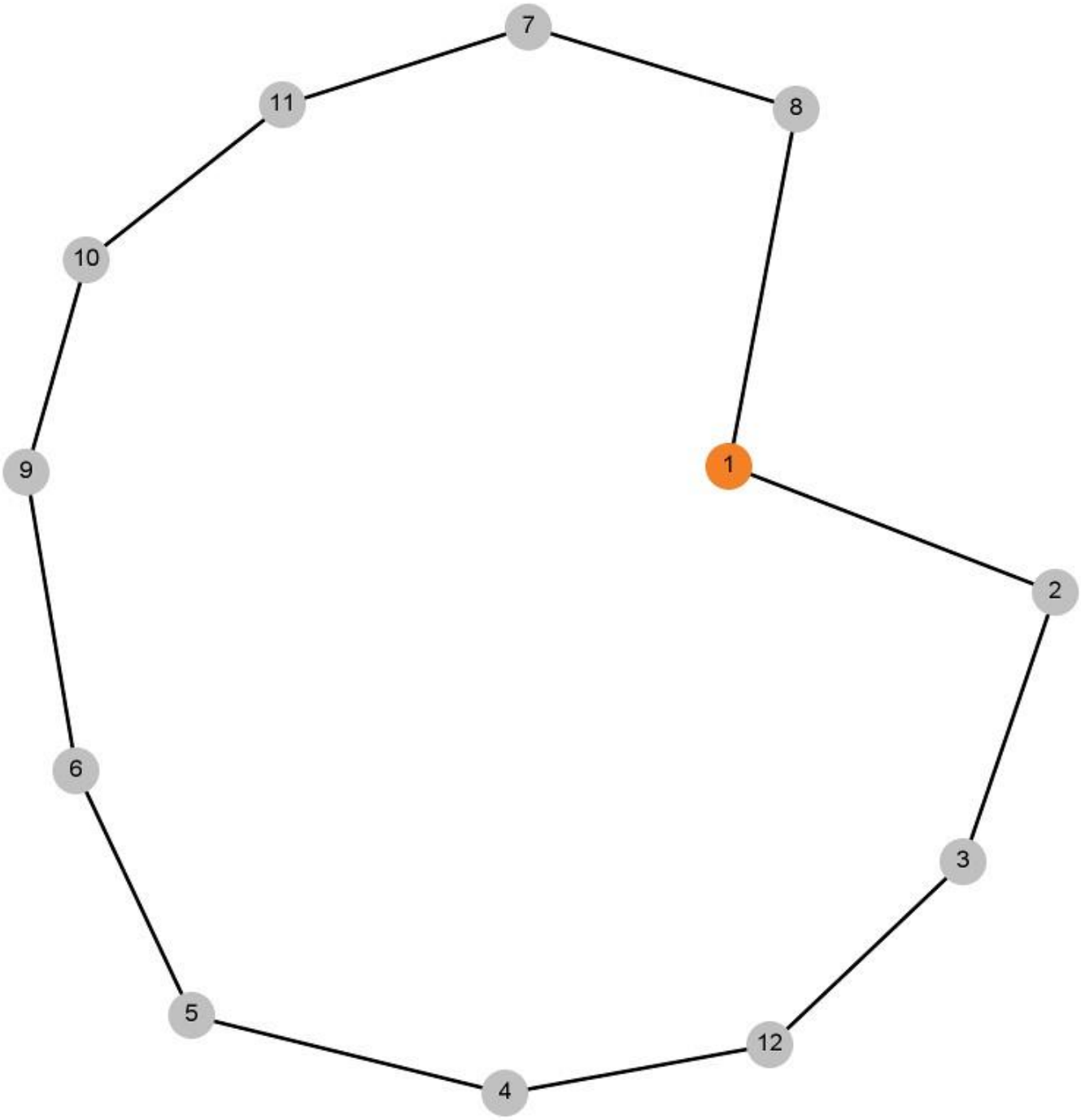
Brute Force Output-3

```
Method: Brute-Force Method
Shortest Distance: 3,71091
Shortest Path: [1, 2, 3, 13, 10, 14, 11, 9, 5, 6, 4, 8, 7, 12, 1]
Time it takes to find the shortest path: 32,04 seconds.
```

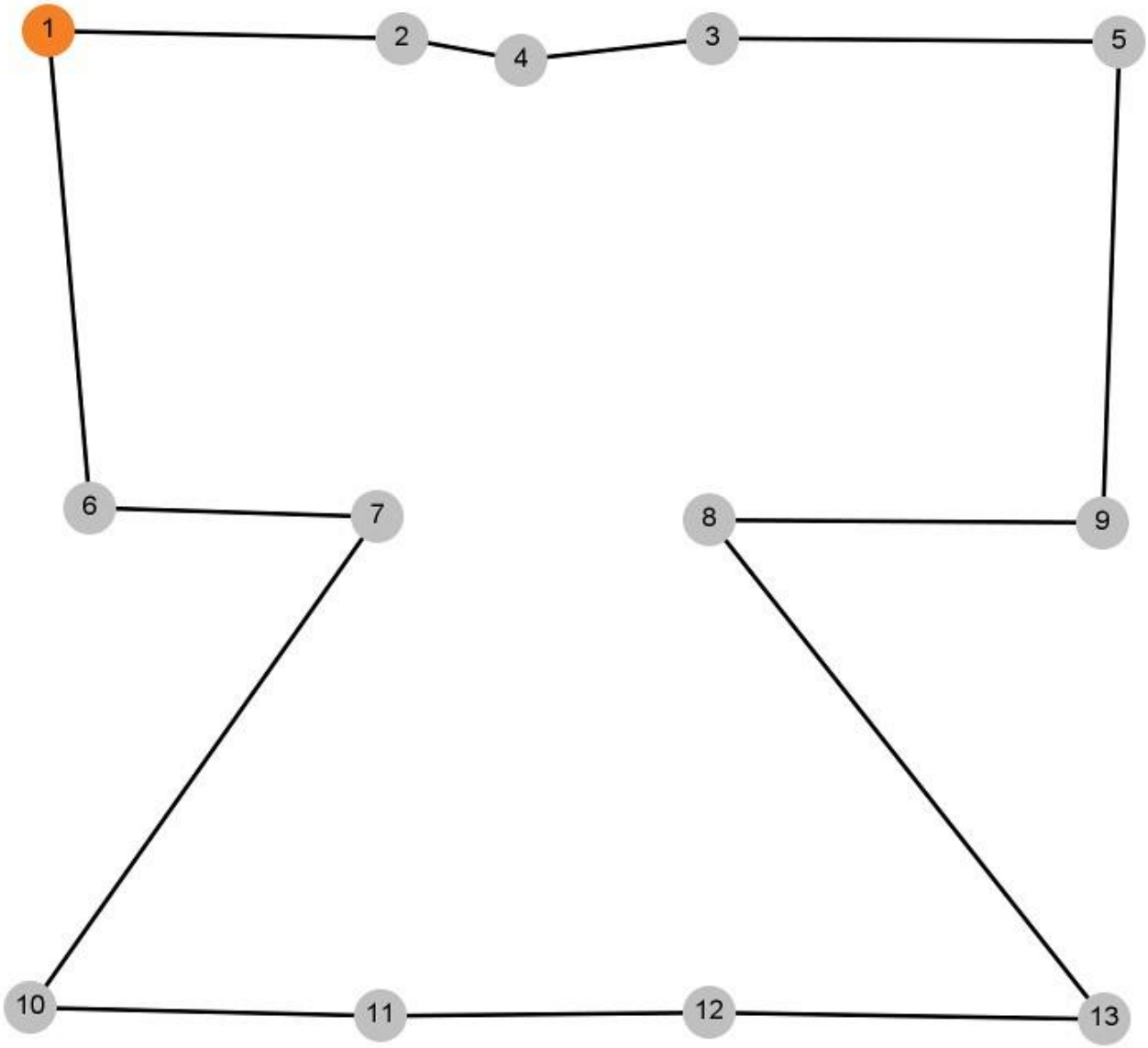
Brute Force Output-4



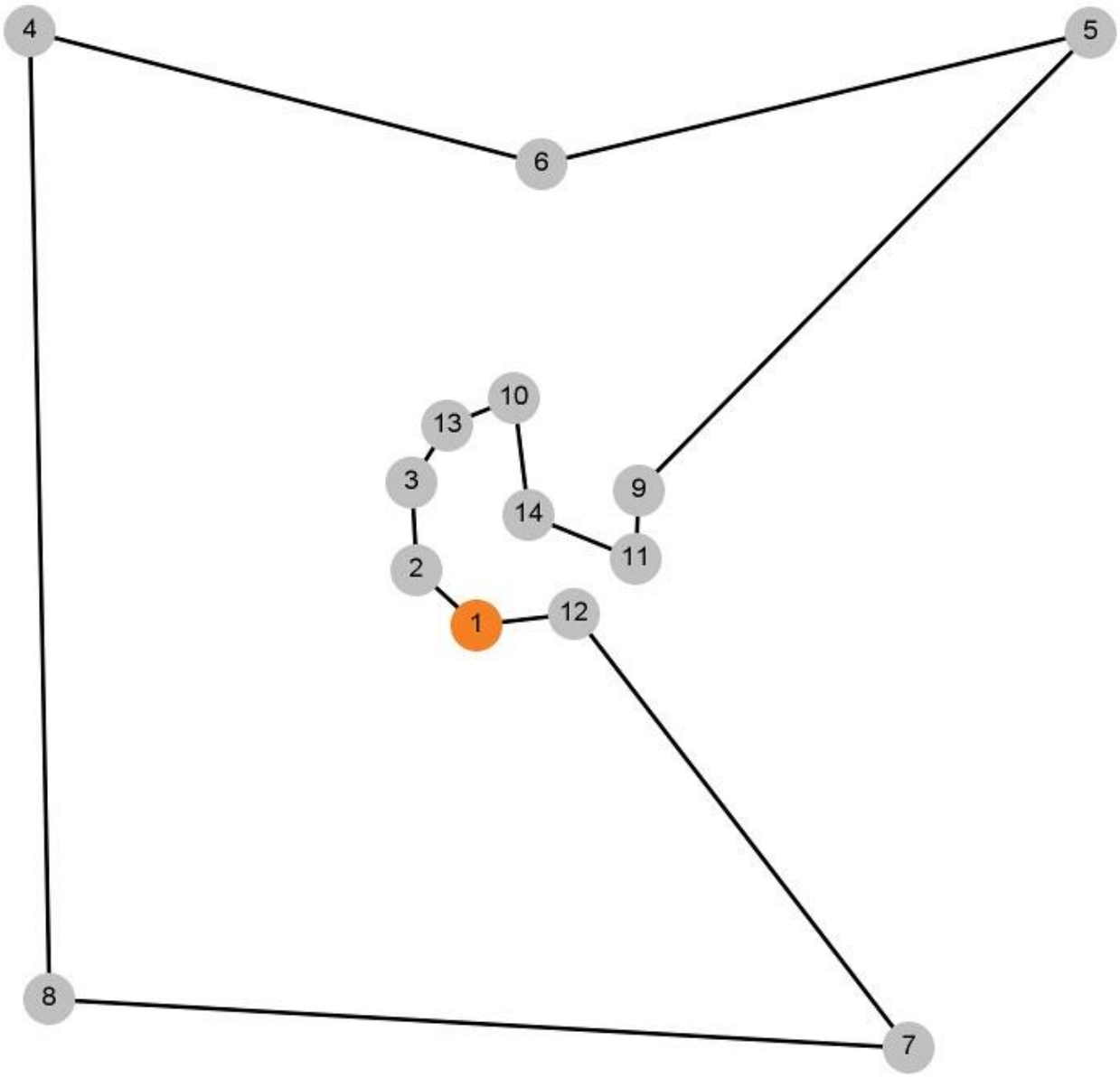
Brute Force Input-1 Shortest Route Map



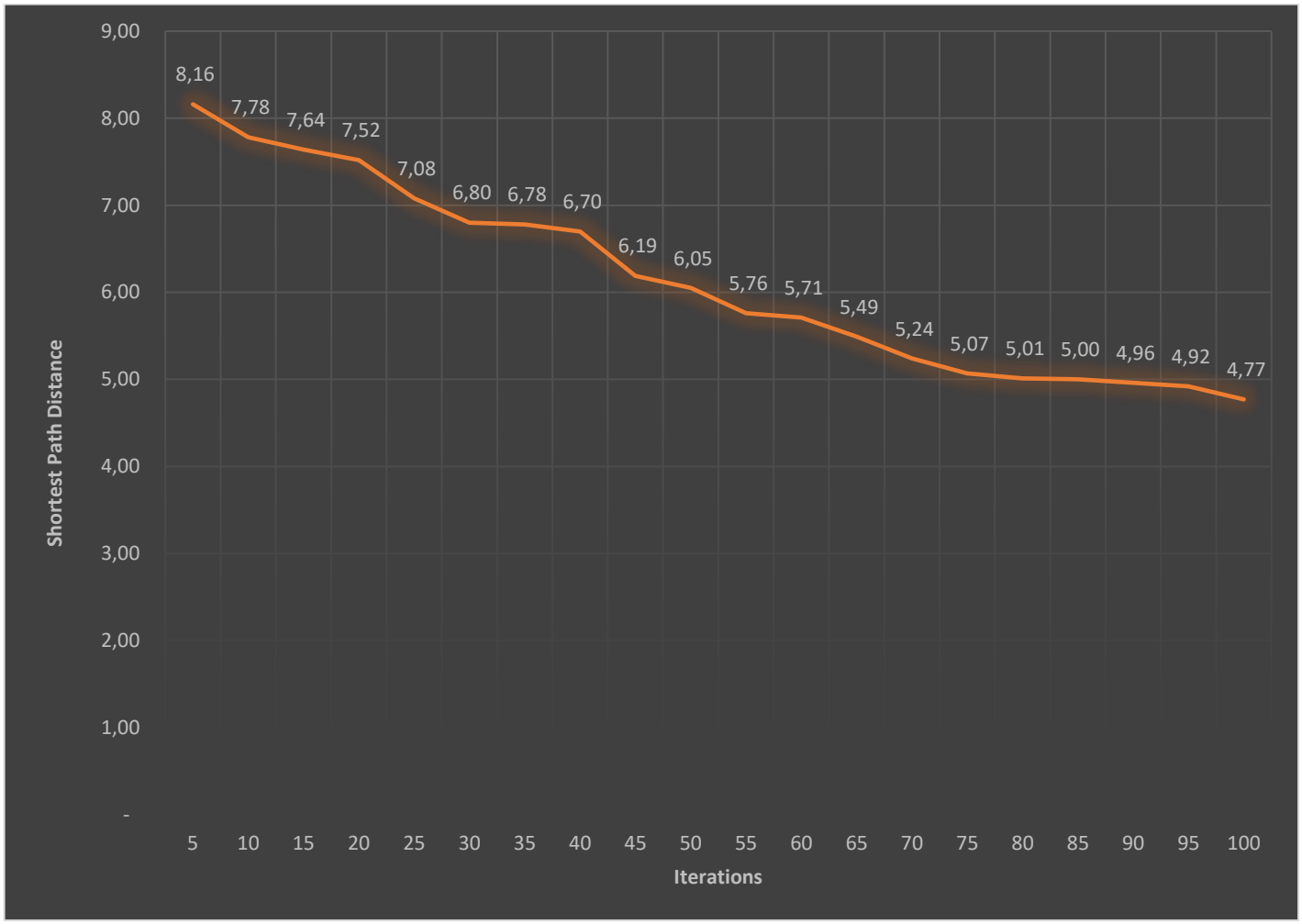
Brute Force Input-2 Shortest Route Map



Brute Force Input-3 Shortest Route Map



Brute Force Input-4 Shortest Route Map



External Resource : https://www.youtube.com/watch?v=783ZtAF4j5g

Advantages of Ant Colony Optimization:

1. Efficiency: ACO typically converges to good solutions quickly, especially in large search spaces, compared to brute force.

2. Scalability: ACO is well-suited for solving complex optimization problems with large solution spaces, as it doesn't require examining all possible solutions.

3. Adaptability: It can adapt to changing environments or requirements due to its decentralized nature and ability to update pheromone trails dynamically.

4. Exploration of Solution Space: ACO explores the solution space intelligently by focusing on promising regions, which helps in finding near-optimal solutions efficiently.

5. Global Optima Search: ACO can find global optima by utilizing pheromone trails to guide the search process towards better solutions.

6. Parallelism: ACO can be parallelized effectively, enabling efficient utilization of computational resources for solving large-scale optimization problems.

Disadvantages of Ant Colony Optimization:

1. Solution Quality: Although ACO usually finds good solutions, it might not always guarantee the best possible solution, as it relies on heuristic methods and probabilistic decisions.

2. Parameter Tuning: ACO algorithms typically involve several parameters (e.g., pheromone evaporation rate, exploration vs. exploitation balance) that require careful tuning to achieve optimal performance.

3. Sensitivity to Parameters: ACO's performance can be sensitive to parameter settings, and finding the right combination of parameters can be challenging.

4. Convergence Speed: While ACO is more efficient than brute force, it might still take longer to converge to a satisfactory solution compared to some other optimization methods, especially for certain types of problems.

5. Complexity: ACO algorithms can be complex to implement and understand, particularly for beginners, due to the intricate interplay of pheromone updates, heuristic information, and stochastic decision-making.

6. Local Optima Traps: Like many optimization algorithms, ACO can get trapped in local optima, especially in multimodal or deceptive fitness landscapes, although the use of pheromone trails helps mitigate this issue to some extent.

Advantages of Brute Force Method:

1. Guaranteed Optimality: Brute force guarantees finding the optimal solution if feasible within the search space, making it suitable for smaller problem instances where exhaustively searching all possibilities is feasible.

2. Simplicity: Brute force methods are conceptually simple and straightforward to implement, making them accessible even to those without advanced optimization knowledge.

3. No Parameter Tuning: Brute force methods typically do not require parameter tuning, as they systematically evaluate all possible solutions.

4. Deterministic: Brute force methods produce deterministic results, which can be advantageous for certain applications where reproducibility and determinism are critical.

Disadvantages of Brute Force Method:

1. Inefficiency: Brute force becomes computationally infeasible as the search space grows exponentially with problem size, making it impractical for large-scale optimization problems.

2. Resource Intensive: Brute force methods require significant computational resources, such as memory and processing power, especially for problems with large solution spaces.

3. Impracticality for Large Search Spaces: For problems with many variables or constraints, the time and resources required to exhaustively search all possibilities make brute force impractical.

4. No Adaptability: Brute force methods lack adaptability and cannot dynamically adjust the search strategy based on the evolving problem characteristics or constraints.

5. Not Suitable for Continuous Spaces: Brute force methods are often limited to discrete optimization problems and may not be directly applicable to continuous optimization domains without discretization.