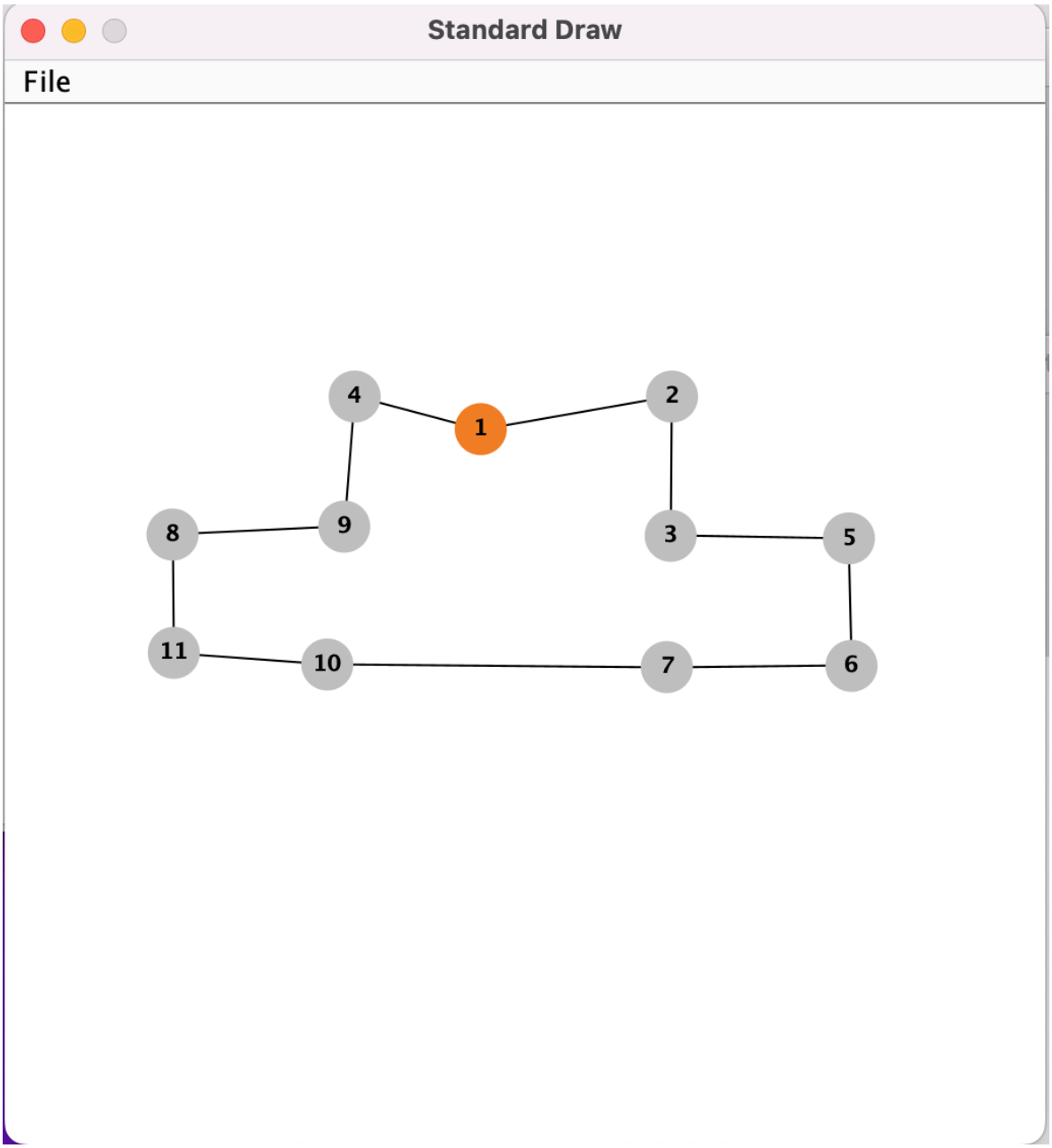
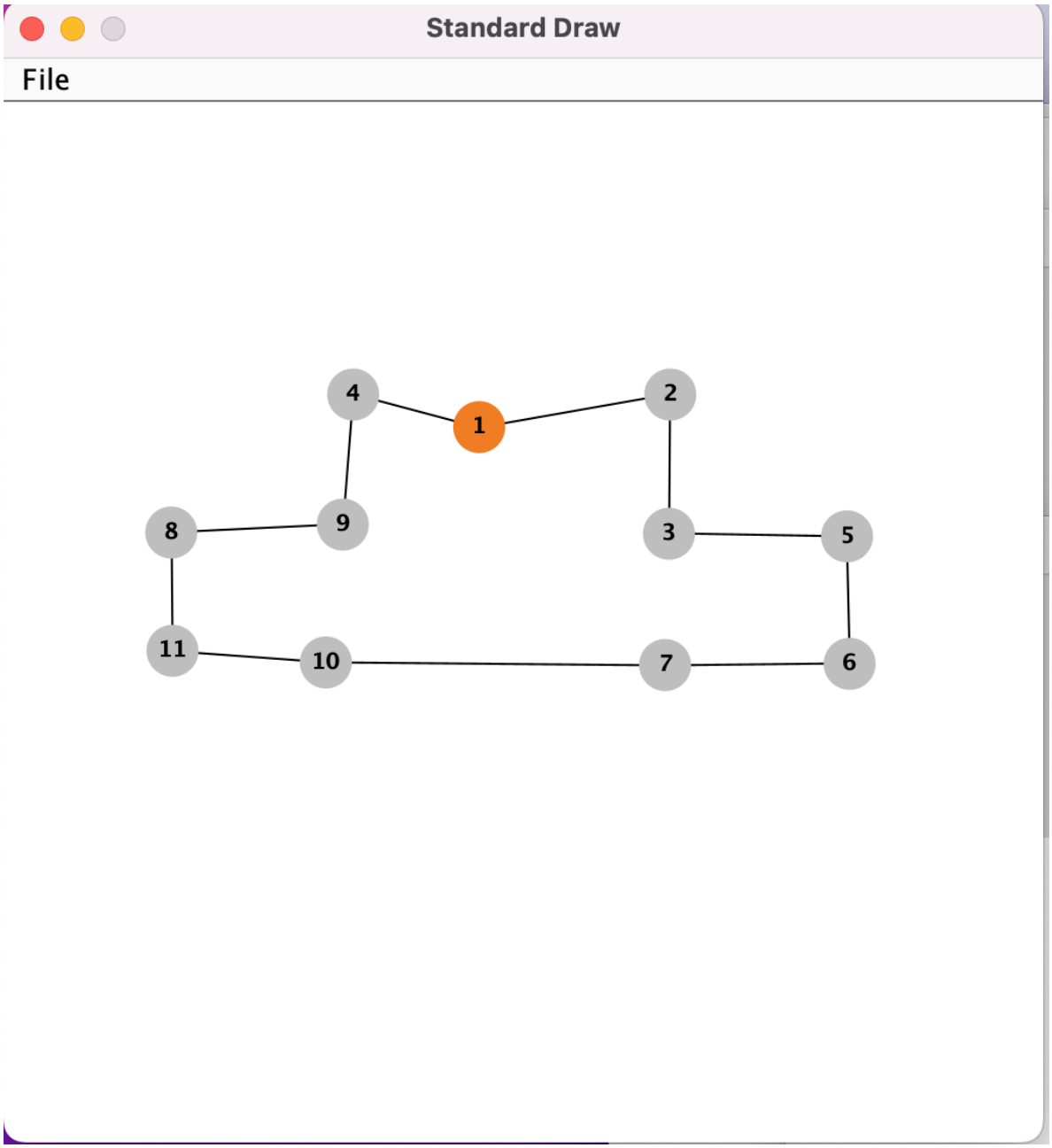
Brute- force Method with Input1

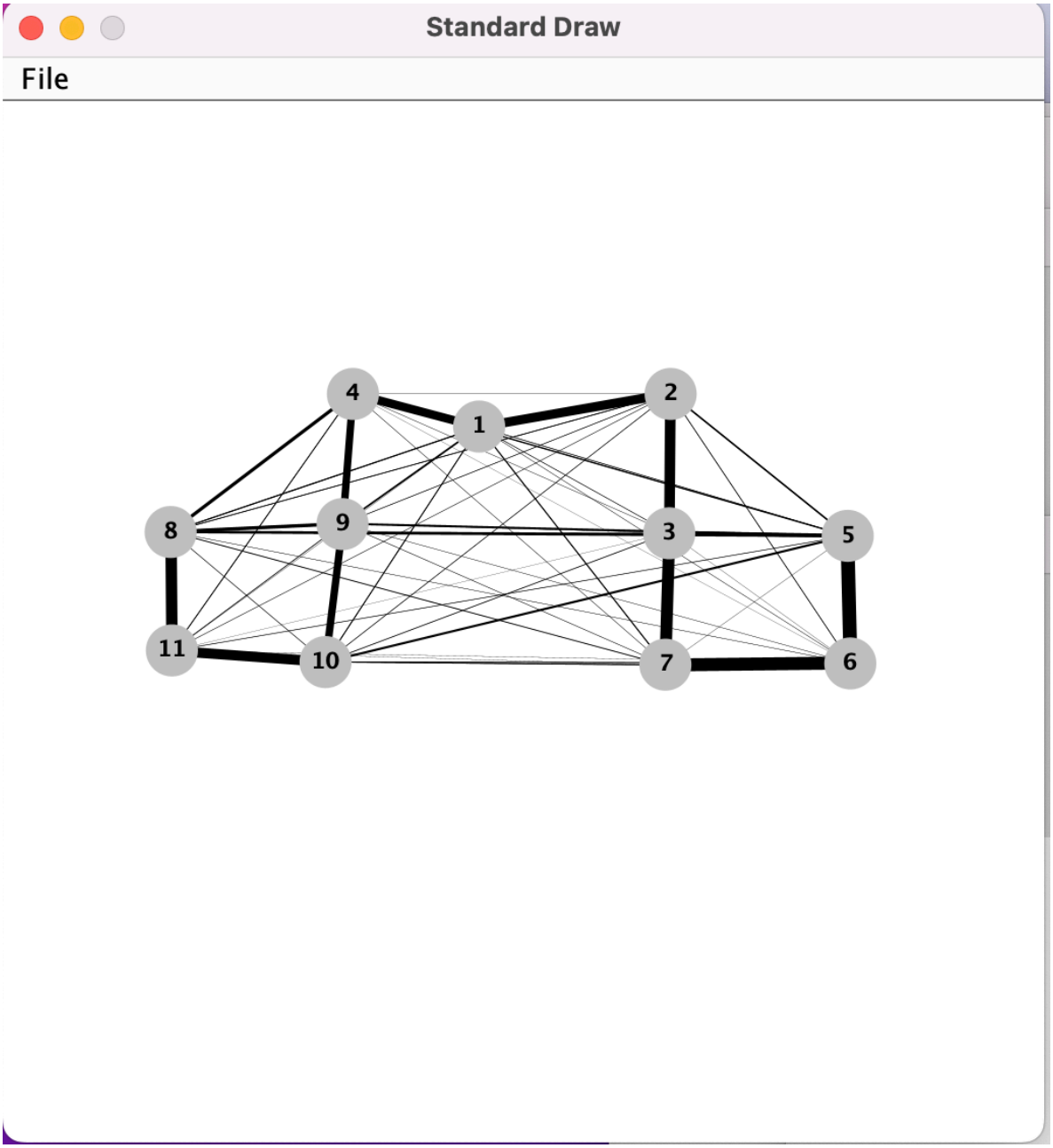



```
Method: Brute-Force Method
Shortest Distance: 1.79529
Shortest Path: [1, 4, 9, 8, 11, 10, 7, 6, 5, 3, 2, 1]
Time it takes to find the shortest path: 1.35
```

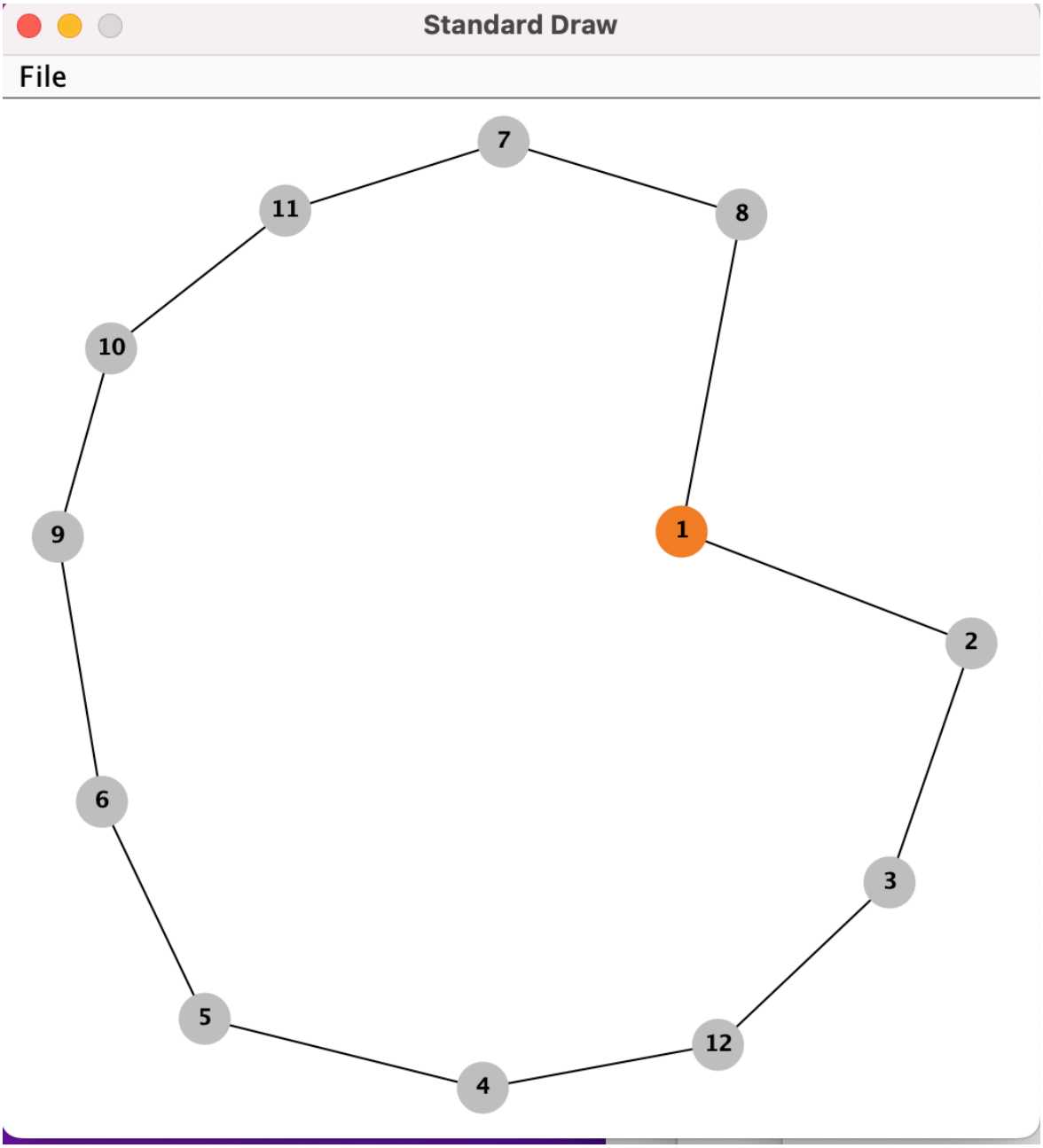
Ant Colony Method with Input1




```
Method: Ant Colony Optimization
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: 1.20
```

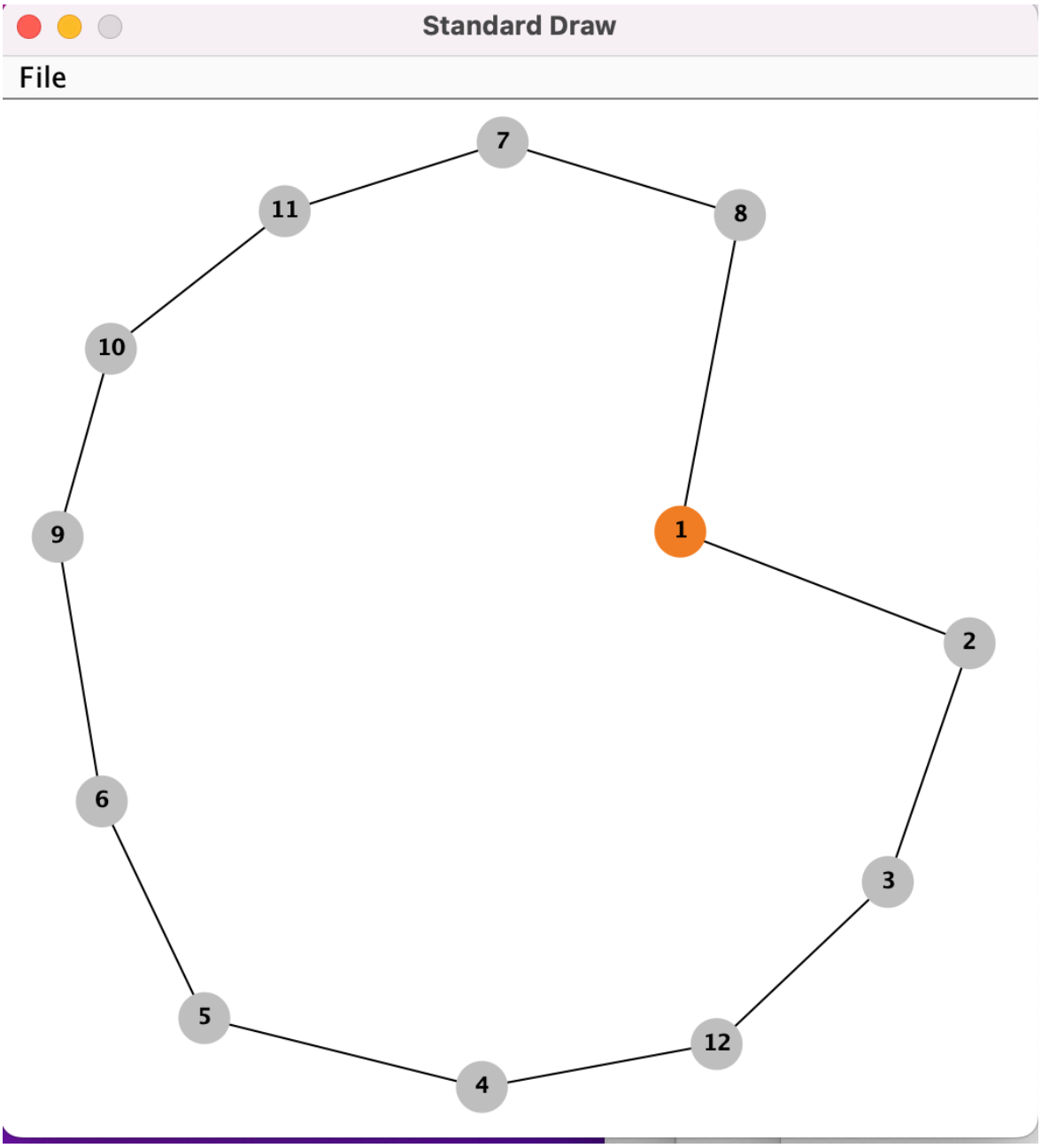

Standard Draw
File
4
2
1
8
9
3
5
11
10
7
6

Brute-force Method with Input2

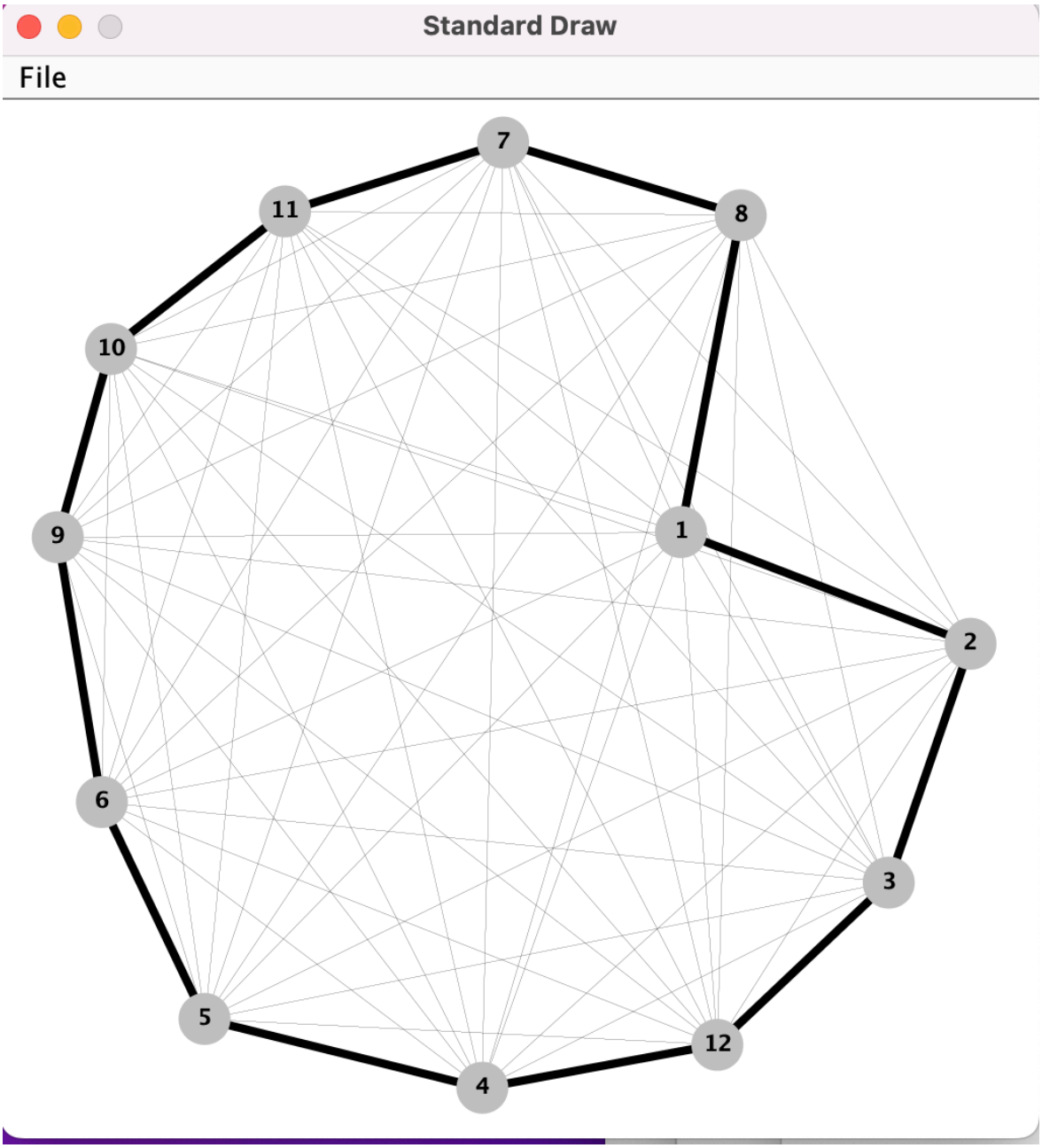



```
Method: Brute-Force Method
Shortest Distance: 2.93588
Shortest Path: [1, 8, 7, 11, 10, 9, 6, 5, 4, 12, 3, 2, 1]
Time it takes to find the shortest path: 7.40
```

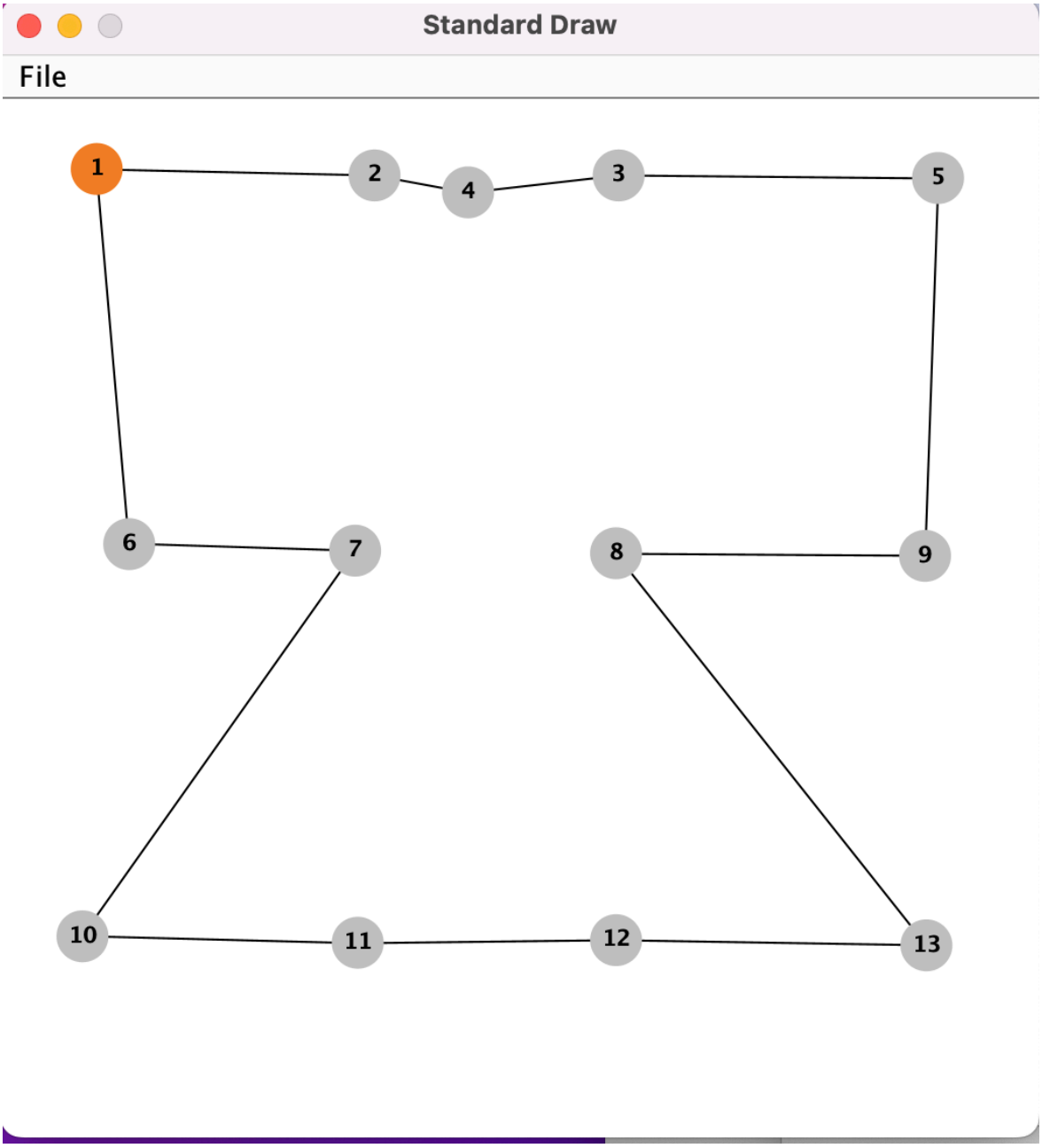
Ant Colony Method with Input2




```
Method: Ant Colony Optimization
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: 1.21
```

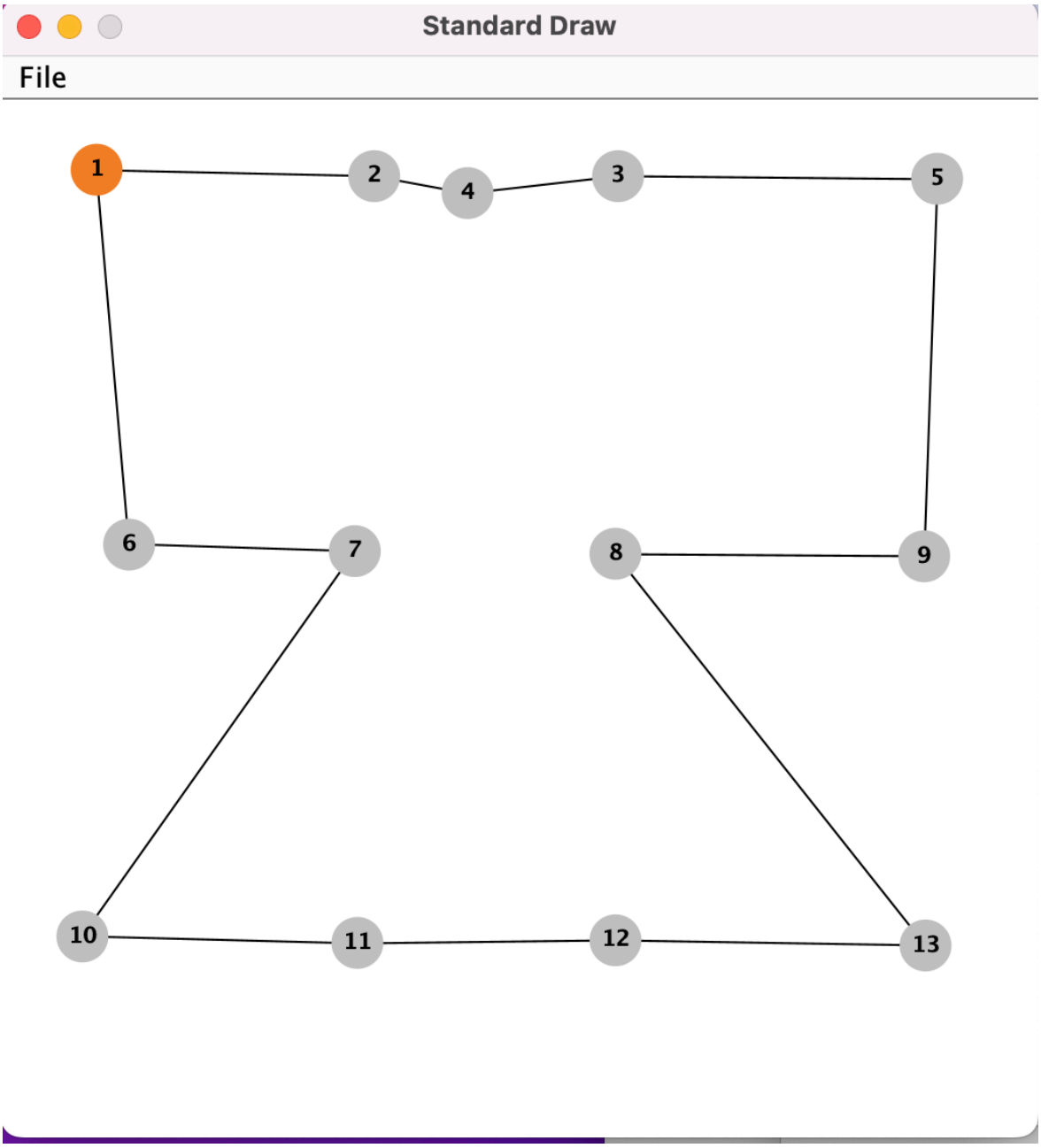

Standard Draw
File
7
11
8
10
9
1
2
6
3
5
12
4

Brute-force Method with Input3

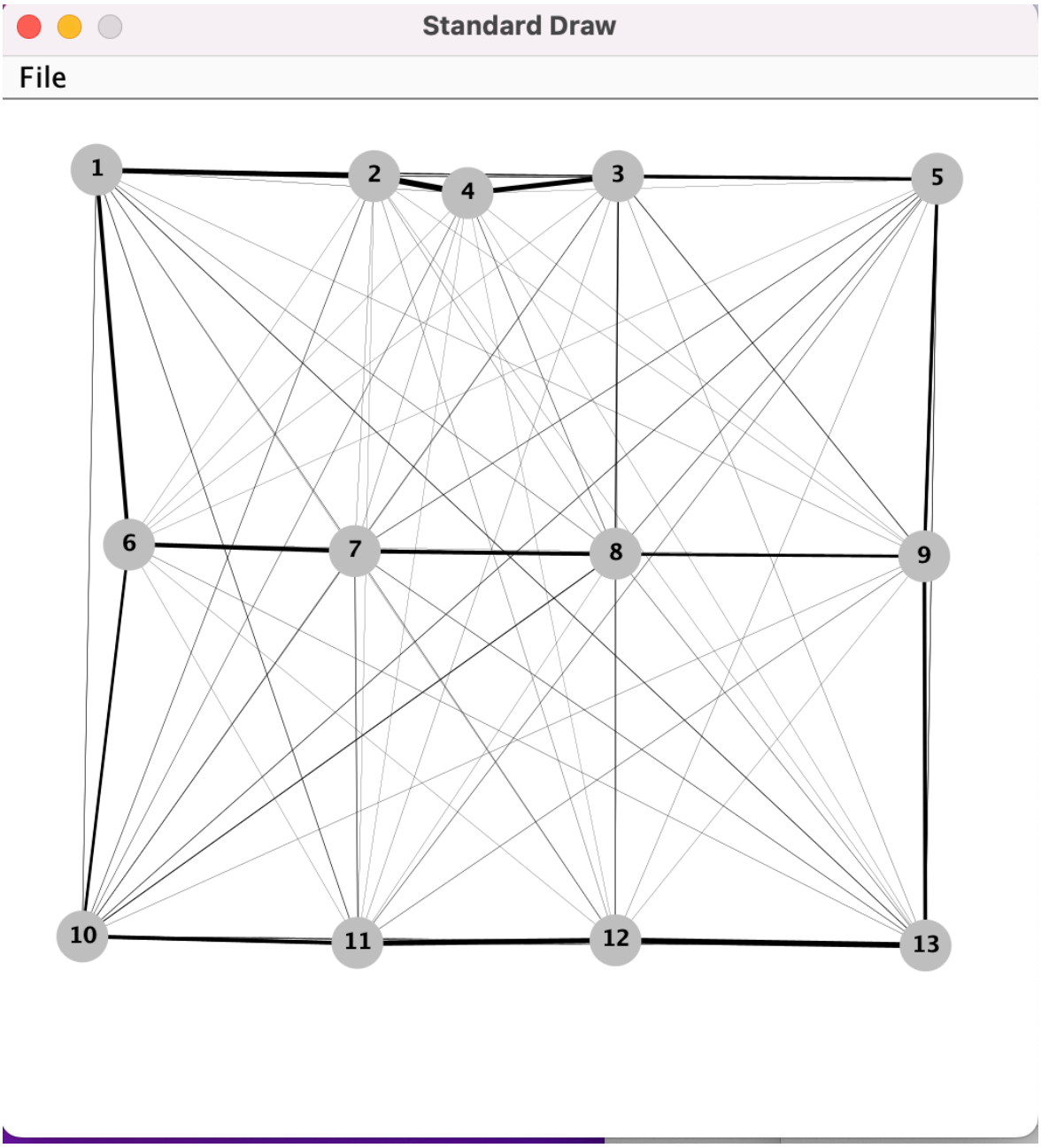



```
Method: Brute-Force Method
Shortest Distance: 3.80292
Shortest Path: [1, 2, 4, 3, 5, 9, 8, 13, 12, 11, 10, 7, 6, 1]
Time it takes to find the shortest path: 106.52
```

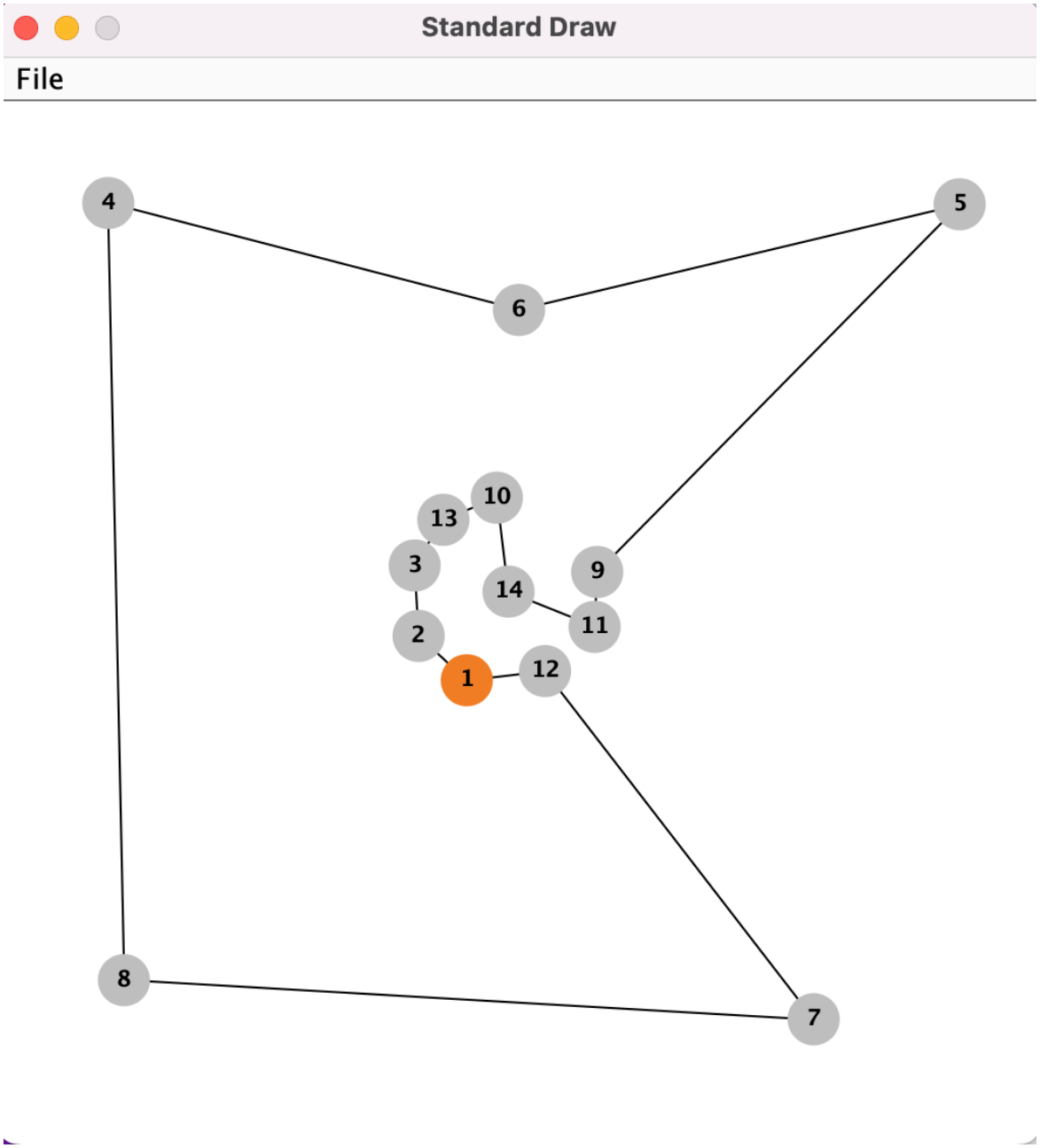
Ant Colony Method with Input3




```
Method: Ant Colony Optimization
Shortest Distance: 3.80292
Shortest Path: [1, 6, 7, 10, 11, 12, 13, 8, 9, 5, 3, 4, 2, 1]
Time it takes to find the shortest path: 1.26
```

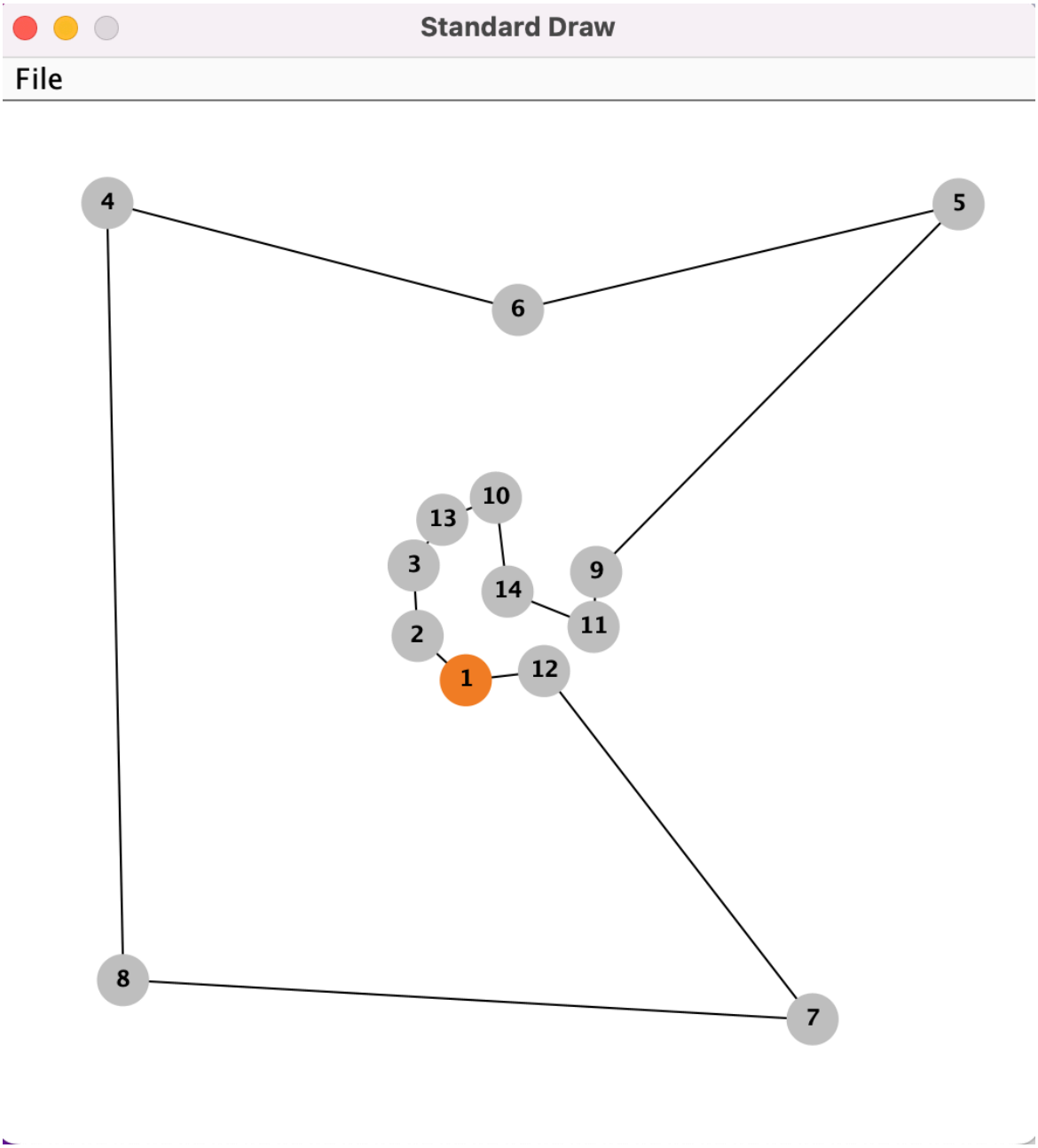

Standard Draw
File
1
2
4
3
5
6
7
8
9
10
11
12
13

Brute-force Method with Input4

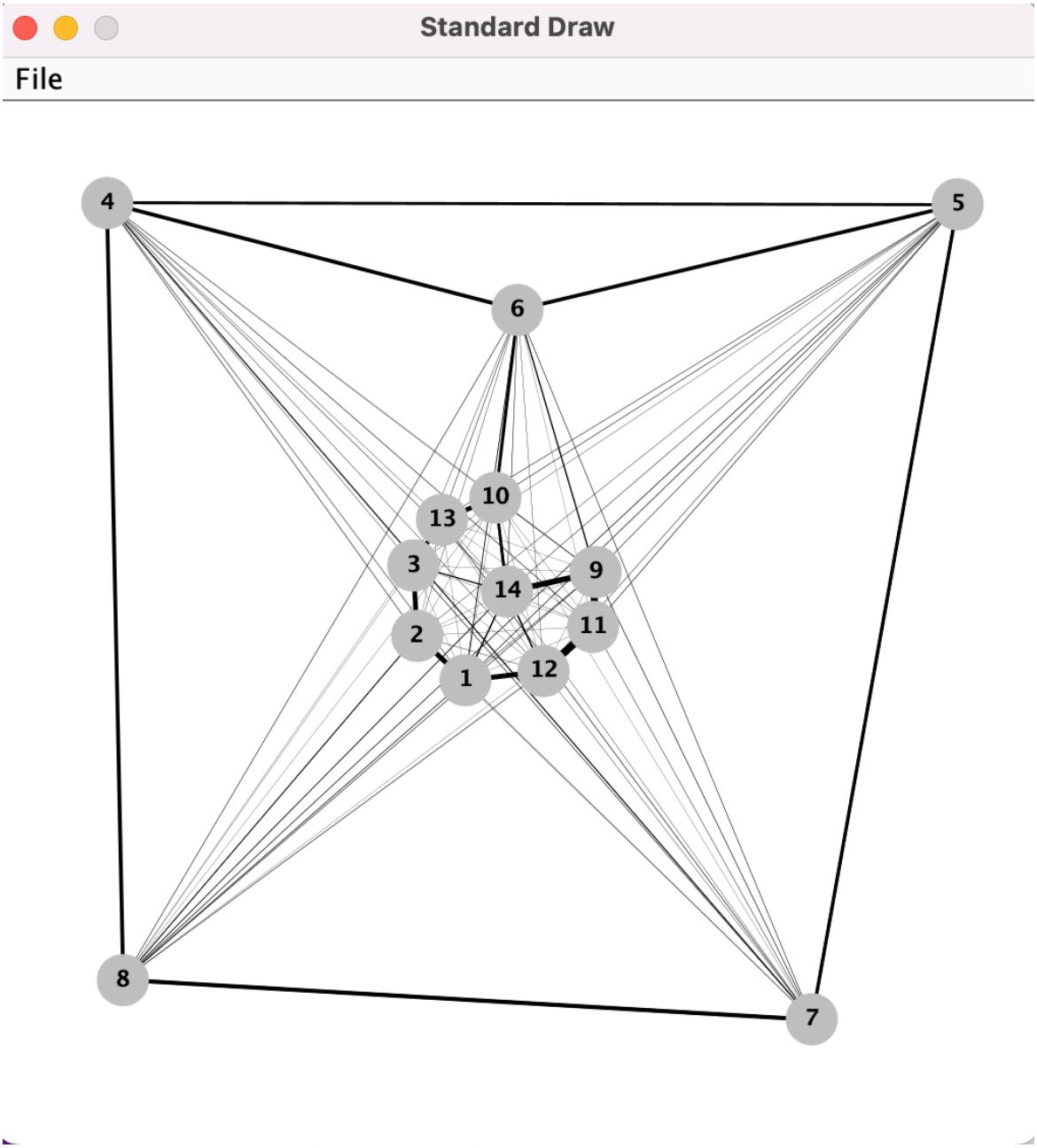



```
Method: Brute-Force Method
Shortest Distance: 3.71091
Shortest Path: [1, 2, 3, 13, 10, 14, 11, 9, 5, 6, 4, 8, 7, 12, 1]
Time it takes to find the shortest path: 759.28
```

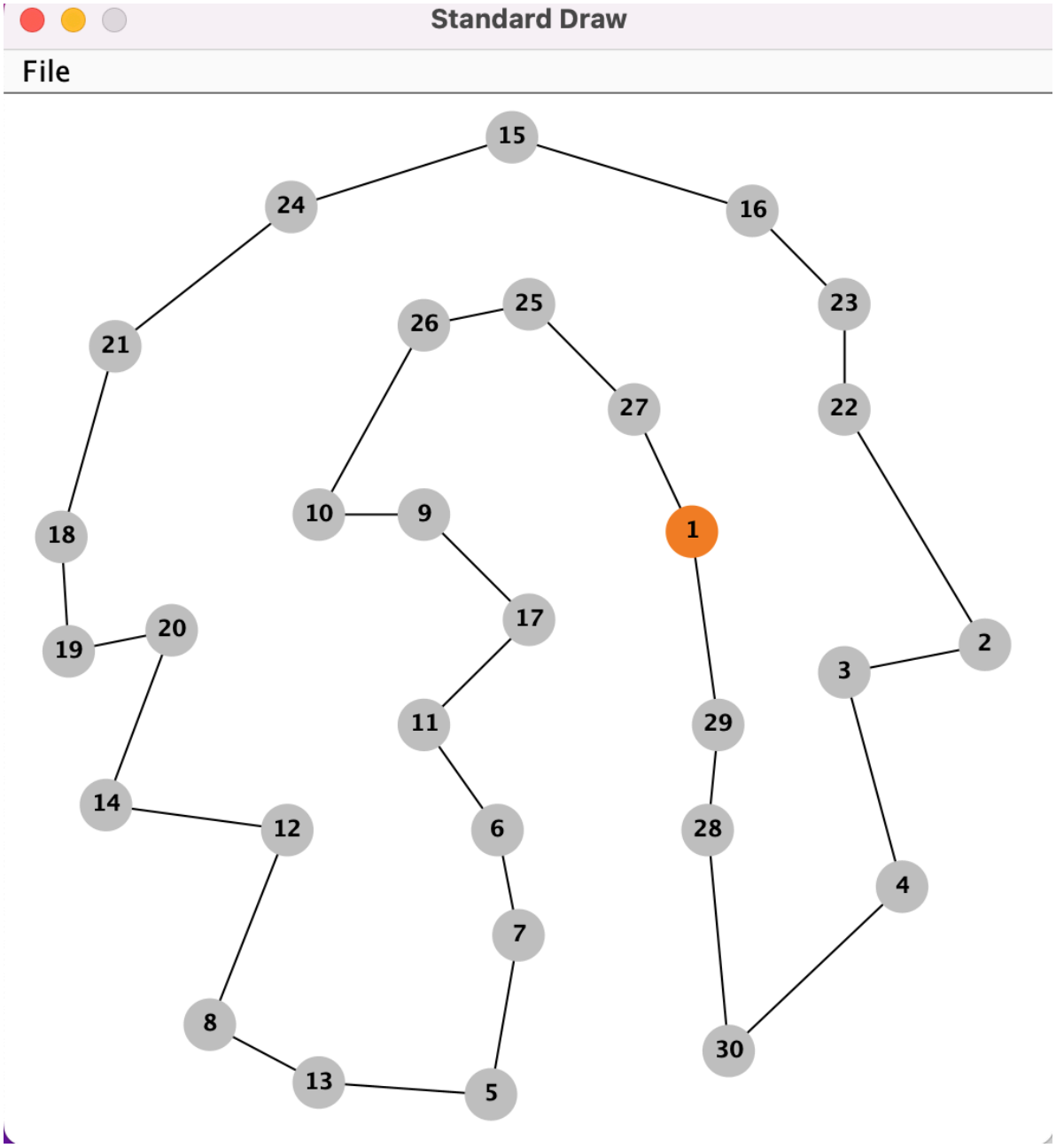
Ant Colony Method with Input4



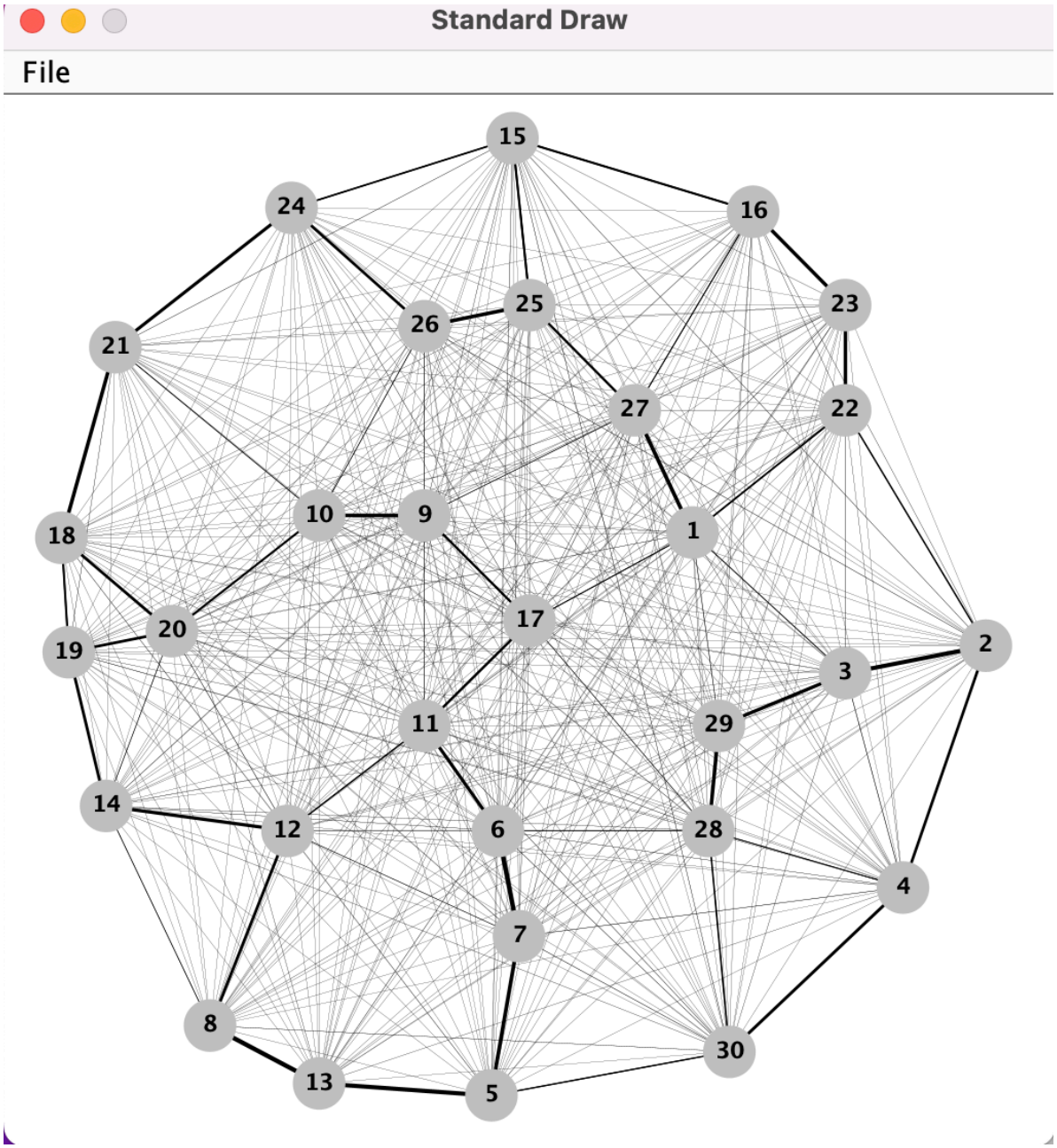

```
Method: Ant Colony Optimization
Shortest Distance: 3.71091
Shortest Path: [1, 12, 7, 8, 4, 6, 5, 9, 11, 14, 10, 13, 3, 2, 1]
Time it takes to find the shortest path: 1.55
```


Standard Draw
File
4
5
6
10
13
3
14
9
11
2
12
1
8
7

Ant Colony Method with Input5




```
Method: Ant Colony Optimization
Shortest Distance: 4.79805
Shortest Path: [1, 29, 28, 30, 4, 3, 2, 22, 23, 16, 15, 24, 21, 18, 19, 20, 14, 12, 8, 13, 5, 7, 6, 11, 17, 9, 10, 26, 25, 27, 1]
Time it takes to find the shortest path: 3.19
```

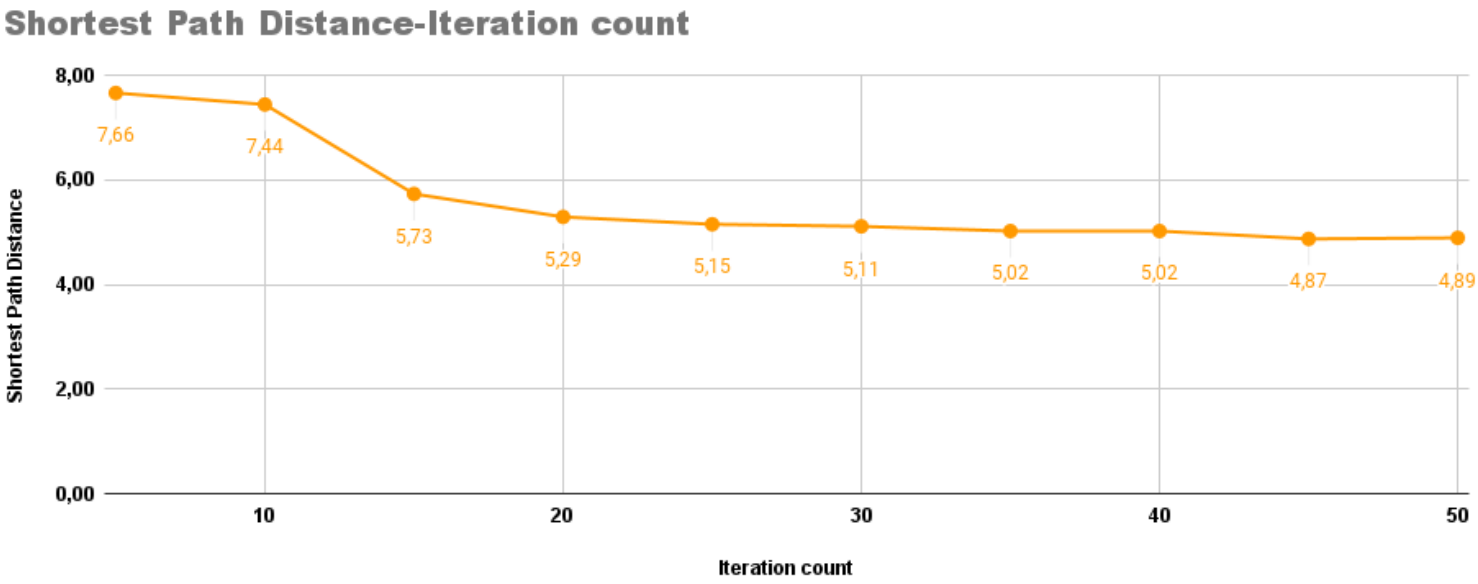

Standard Draw
File
15
24
16
25
26
23
21
27
22
10
9
1
18
17
20
19
2
3
11
29
14
12
6
28
4
7
8
30
13
5

My Best Ant Colony Hyperparameters:
Maximum iteration count: 100
Ant count per iteration: 50
Degradation factor: 0.5
Alpha: 0.9
Beta: 1.8
Initial pheromone intensity: 0.1
Q value: 0.0001

Comparison of Methods:

Input File	Number of Houses+ Migros	Brute-Force Time(seconds)	Ant Colony Time(seconds)	Speed Up Factor
Input1	11	1.35	1.20	1.13 times faster
Input2	12	7.40	1.21	6.12 times faster
Input3	13	106.52	1.26	84.54 times faster
Input4	14	759.28	1.55	489.86 times faster
Input5	30	Too long to compute	3.19	Too much to compute

Best Distance vs. Iteration count Graph for Ant Colony Method:



Advantages and Disadvantages of the Ant Colony Optimization Method:

Disadvantages:

- **Dependency on Parameters:** The ACO algorithm has several parameters that need to be fine-tuned for optimal results. Fine-tuning such parameters may require several iterations of the algorithm, which can be time-consuming.
- **Unstability of Algorithm Performance:** The performance of ACO algorithms can be unstable when the problem size increases. ACO might not provide the best solution with larger problem sizes as the algorithm runs out of time while trying different combinations.
- **The Need for Large-Scale Memory:** The ACO algorithm requires large-scale memory storage to store the various probabilities used to calculate the selection of the next state.

Advantages:

- **Faster Processing:** The ACO algorithm is designed to solve complex problems in less time compared to traditional methods. It effectively searches for the optimal solution in a short time.
- **A Metaheuristic Approach:** ACO is a metaheuristic approach that can be applied to various optimization problems, making it an eligible choice for solving any optimization problem.
- **Easy Implementation and Maintenance:** The ACO algorithm is easy to implement and maintain, requiring only a few iterations to converge. Unlike other conventional methods, which may require more complex algorithms, ACO is easy to maintain and update.
- **An Efficient Solution:** ACO algorithms provide an efficient solution to optimization problems by developing good-quality solutions faster in less computation time.

Despite all the disadvantages of the Ant Colony Algorithm listed above, it provides optimal solutions for most problems much faster than the conventional methods. As a result, Ant Colony Optimization is a popular and useful algorithm in the computer since community.

References:

https://www.google.com/url?sa=t&source=web&rct=j&opi=89978449&url=https://www.geeksforgeeks.org/introduction-to-ant-colony-optimization/&ved=2ahUKEwiI0rv7lYiGAxV4c_EDHdbiBWIQFnoECBMQAQ&usg=AOvVaw3WGWP8MP6xEeMXLRkKvvyK