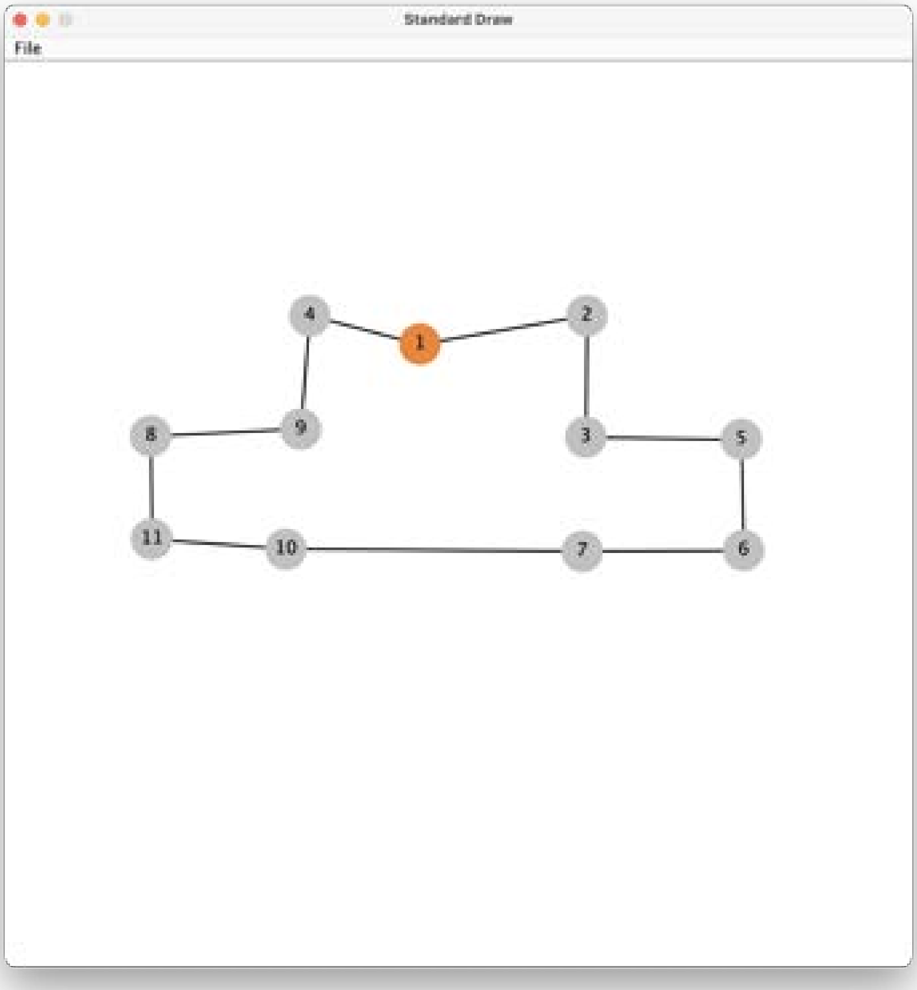
Assignment 3 Report

Mehmet Kaan Ünsel

<u>Input 1:</u>

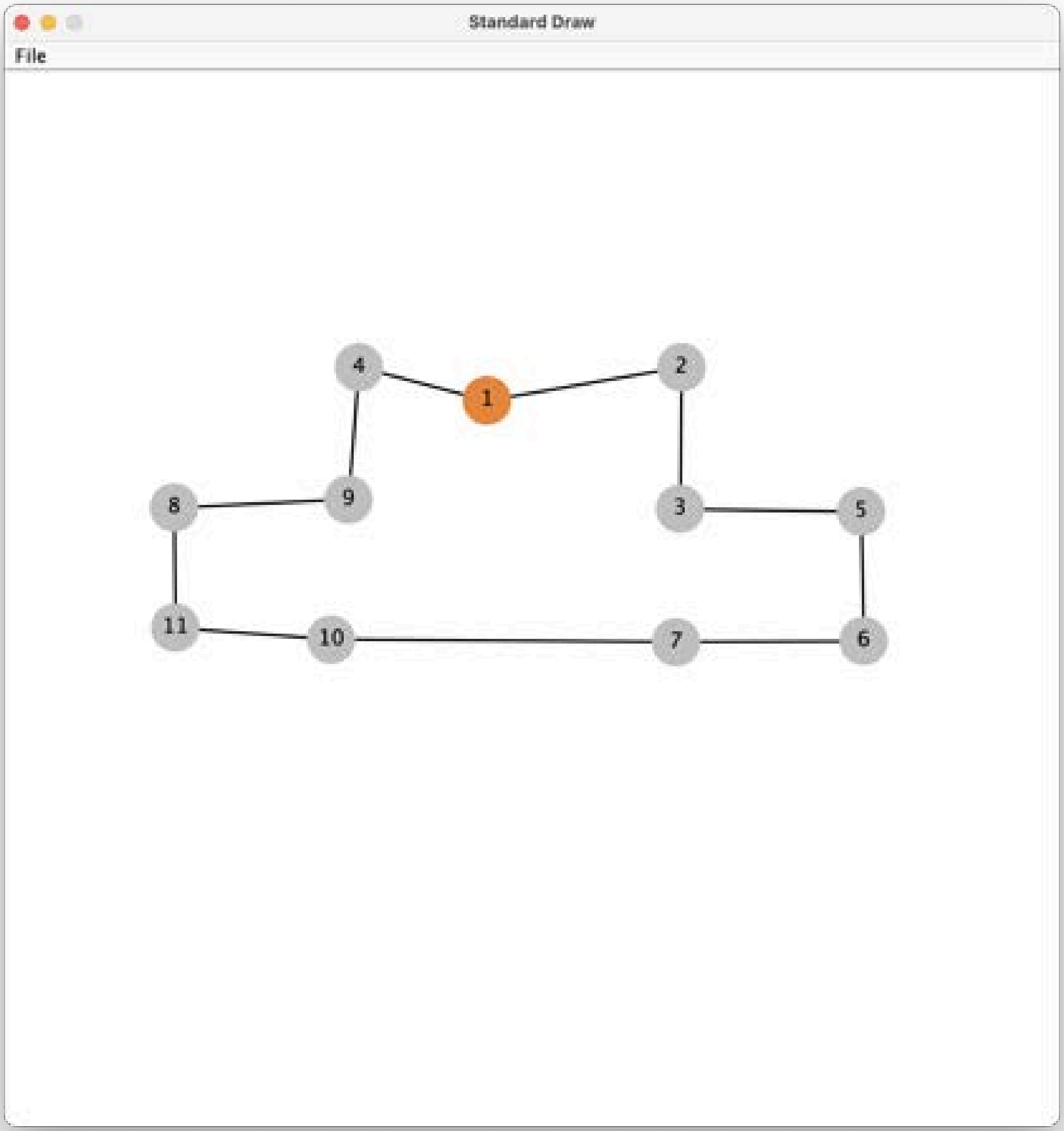
Brute Force:

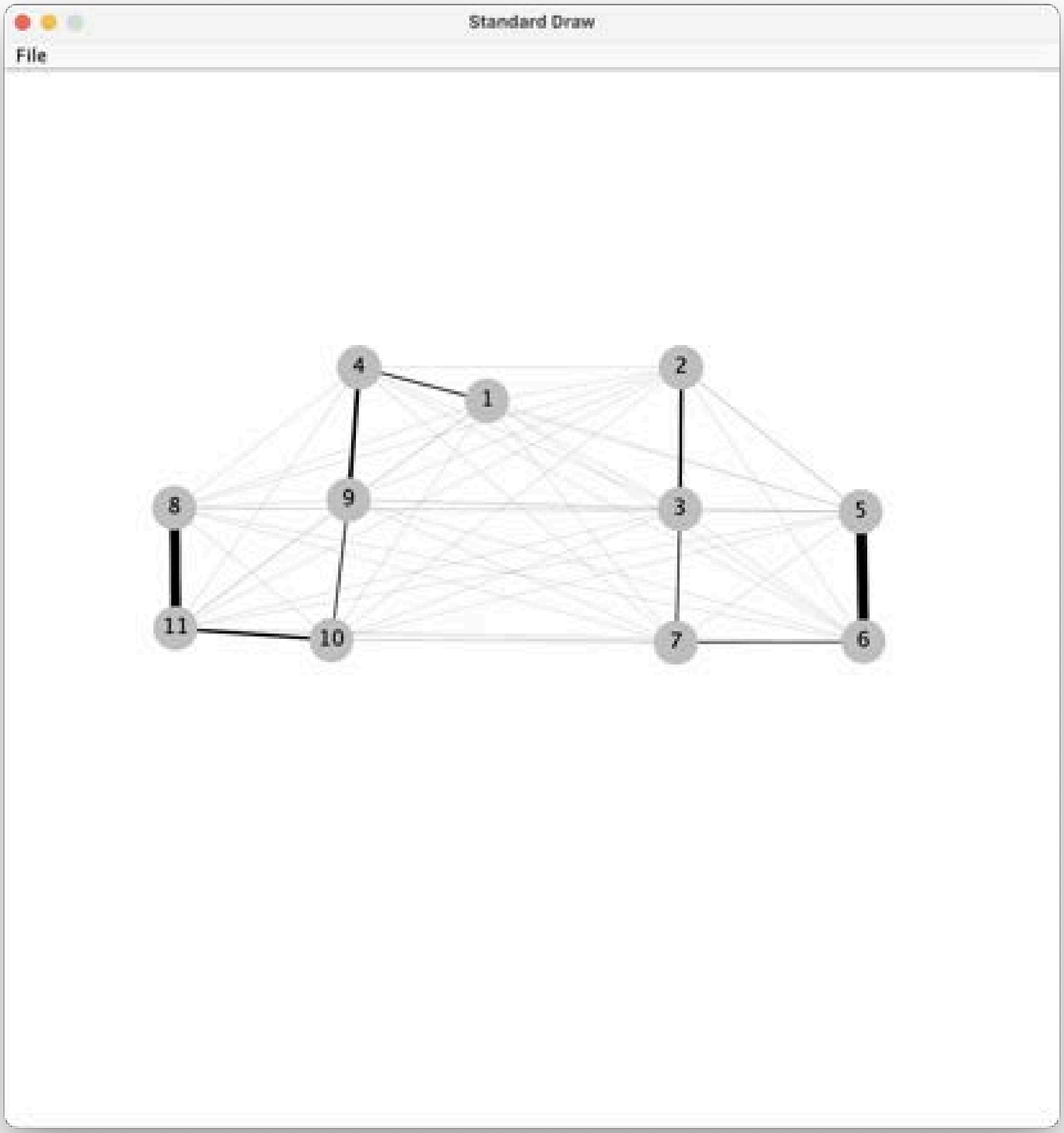


StdOut:

```
Method: Brute-Force Method
Shortest Distance: 1.7952913856772432
Shortest Path: [1, 4, 9, 8, 11, 10, 7, 6, 5, 3, 2, 1]
Time it takes to find the shortest path: 0.175 seconds.
```

Ant Colony Optimization:



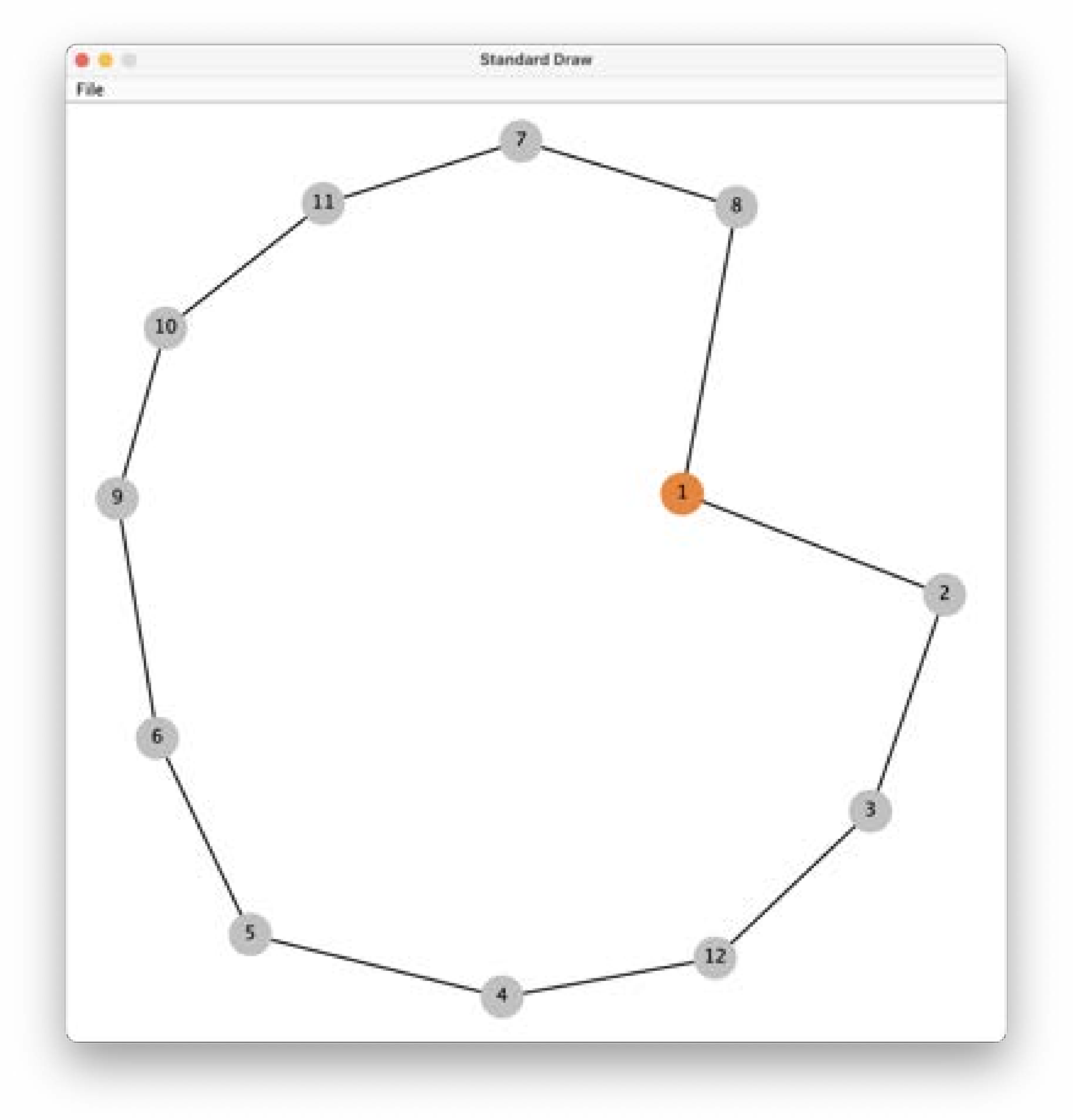


StdOut:

```
Method: Ant Colony Optimization
Shortest Distance: 1.7952913856772432
Shortest Path: [1, 4, 9, 8, 11, 10, 7, 6, 5, 3, 2, 1]
Time it takes to find the shortest path: 0.055 seconds.
```

Input 2:

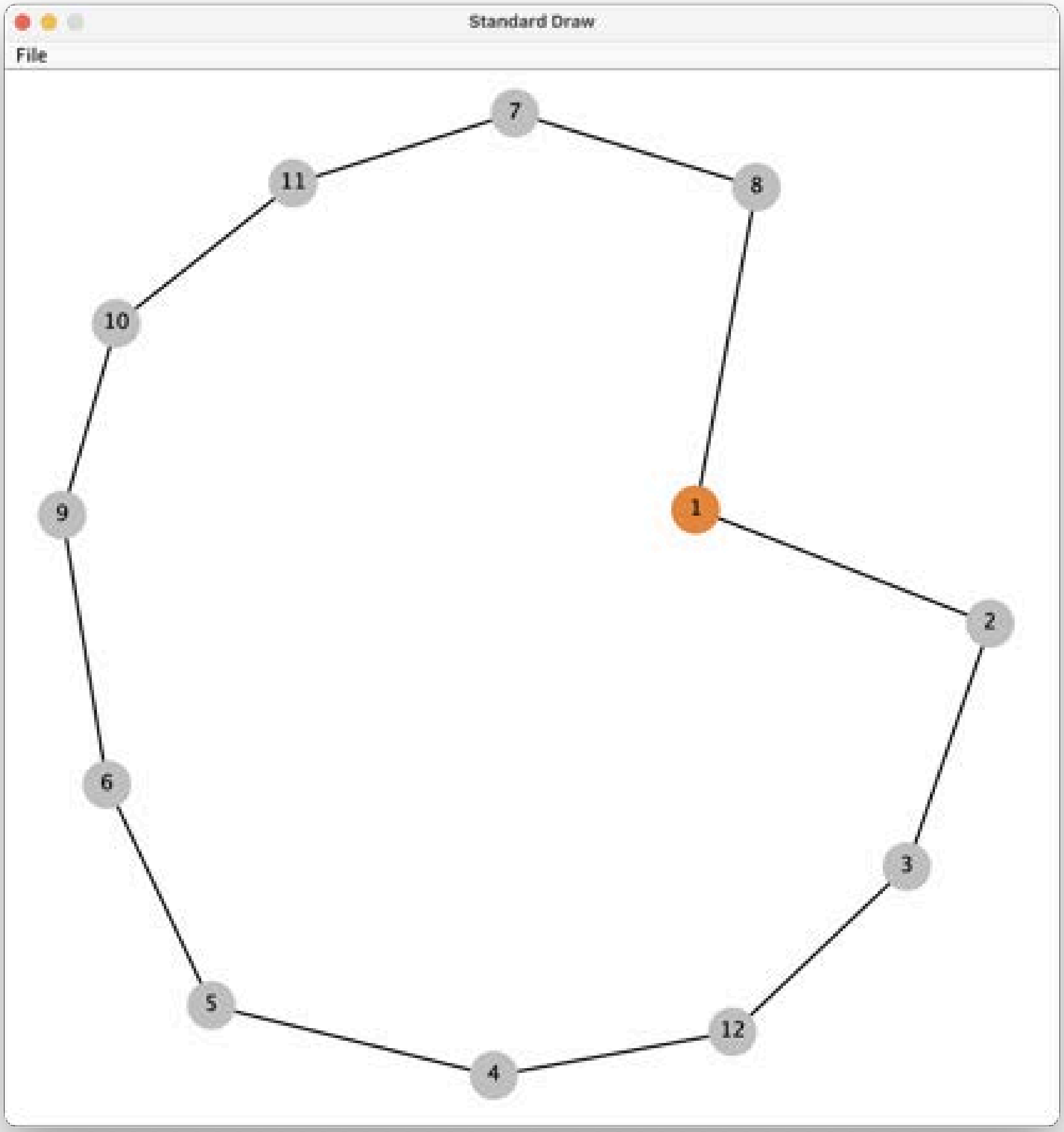
Brute Force:

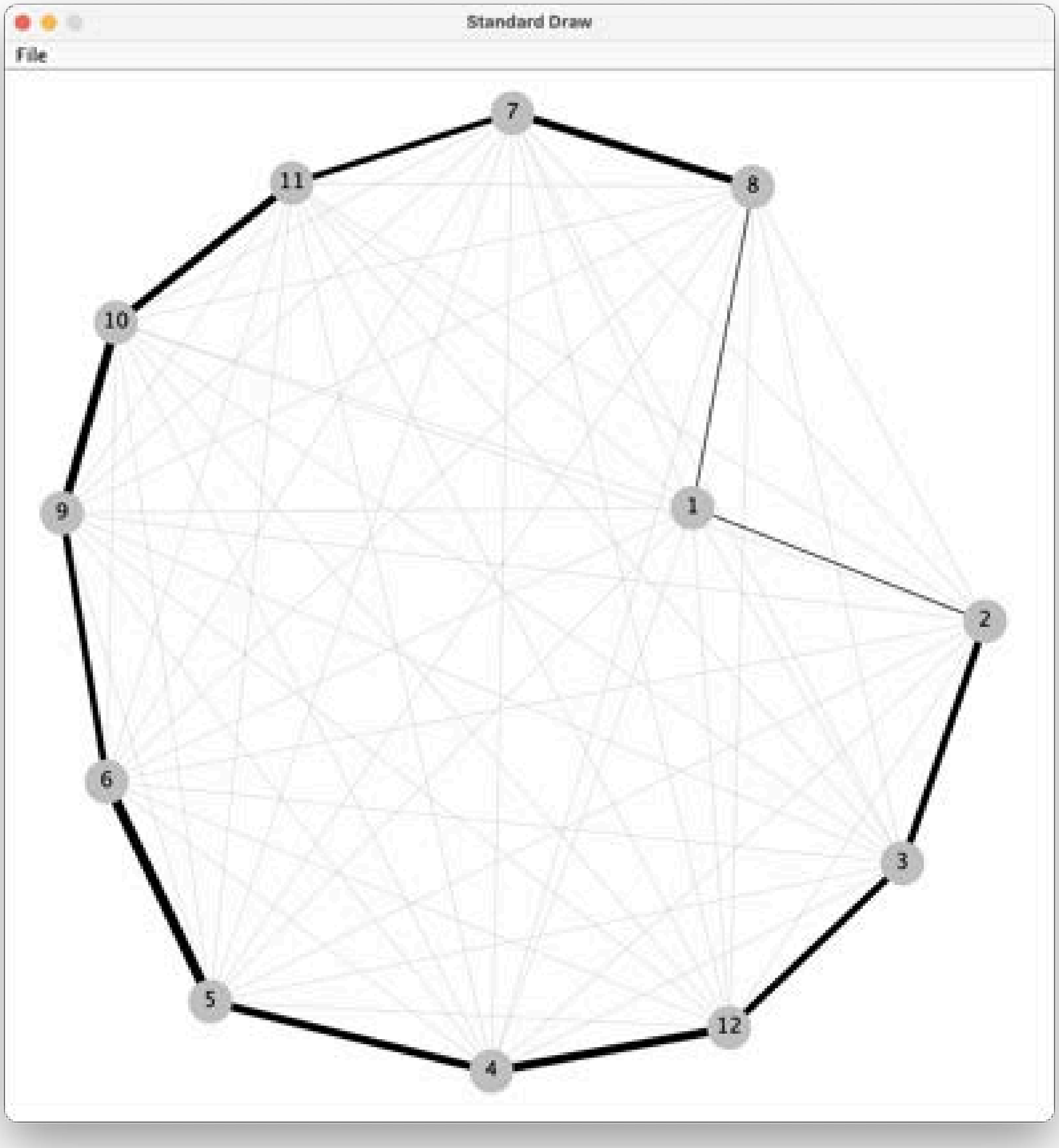


StdOut:

```
Method: Brute-Force Method
Shortest Distance: 2.935877143237598
Shortest Path: [1, 8, 7, 11, 10, 9, 6, 5, 4, 12, 3, 2, 1]
Time it takes to find the shortest path: 1.708 seconds.
```

Ant Colony Optimization:



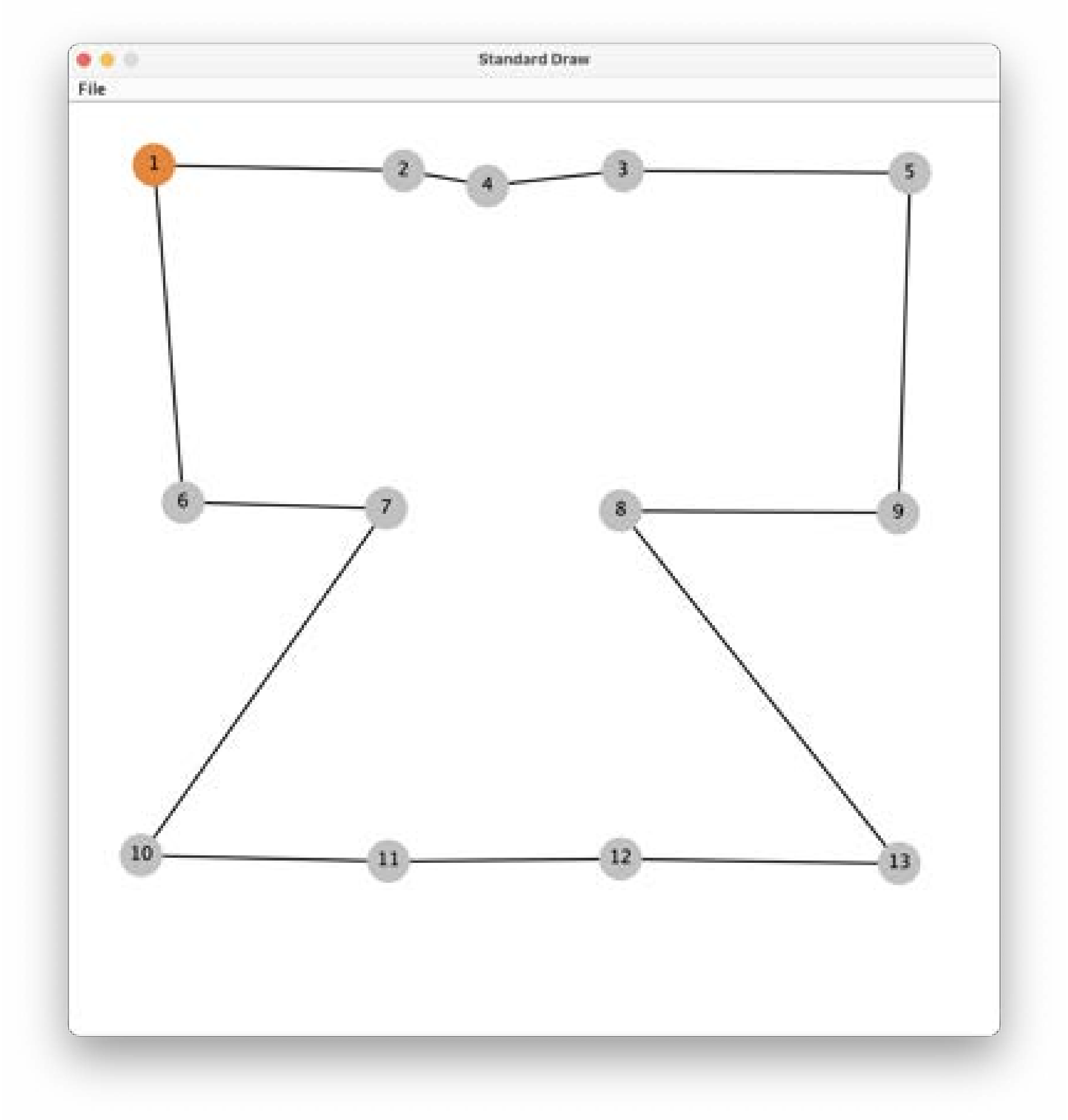


StdOut:

```
Method: Ant Colony Optimization
Shortest Distance: 2.935877143237598
Shortest Path: [1, 8, 7, 11, 10, 9, 6, 5, 4, 12, 3, 2, 1]
Time it takes to find the shortest path: 0.062 seconds.
```

Input 3:

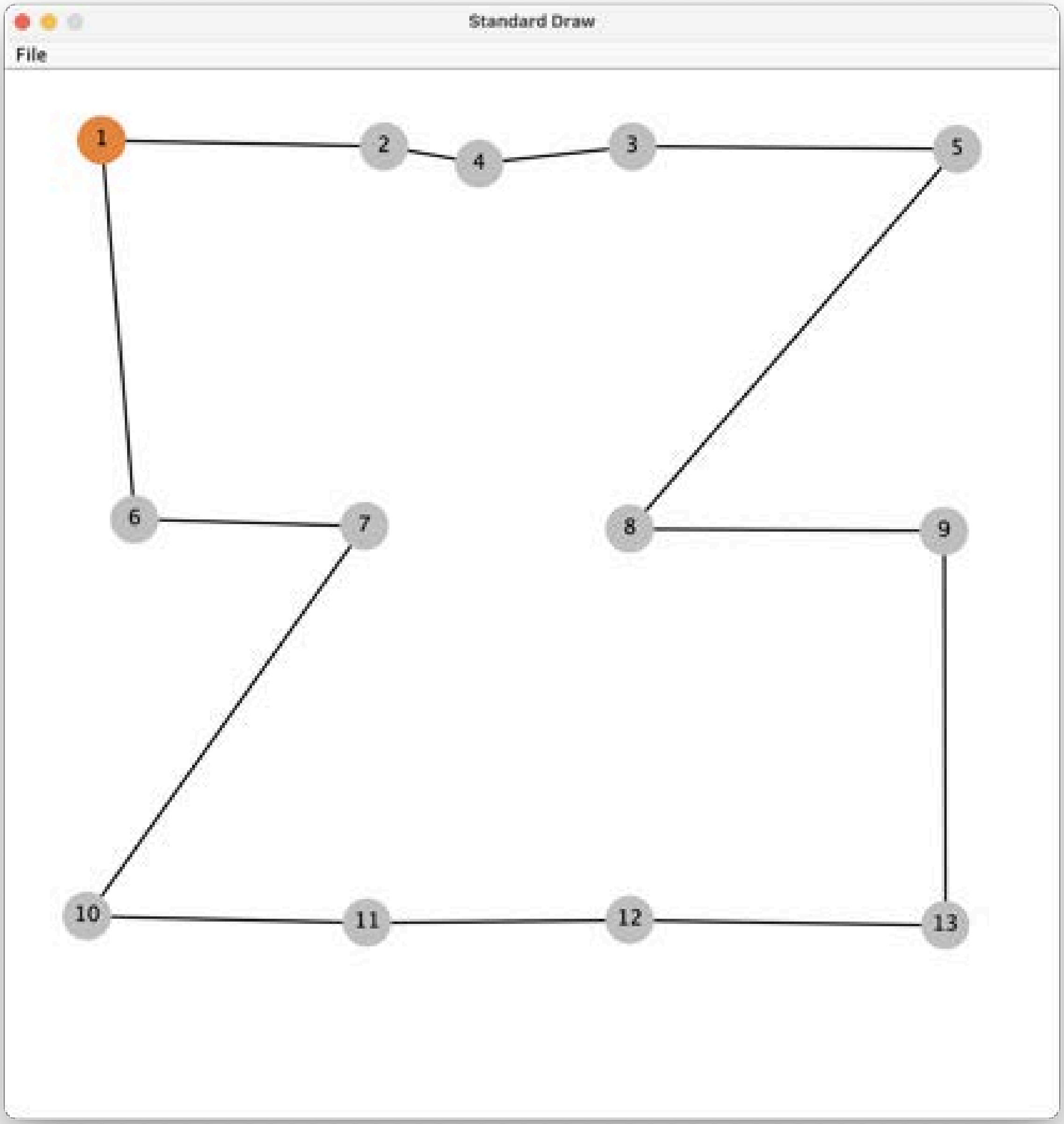
Brute Force:

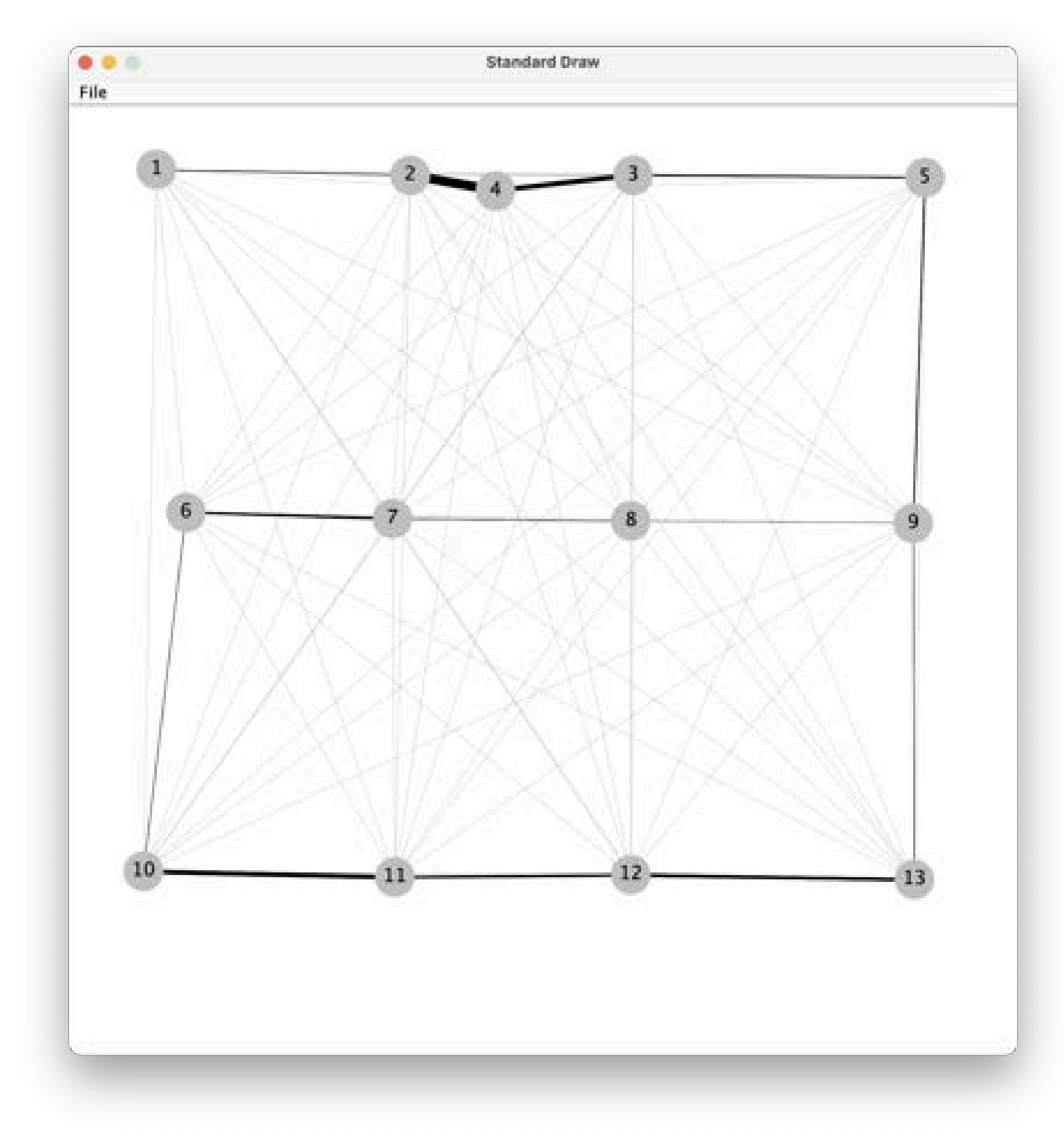


StdOut:

Method: Brute-Force Method
Shortest Distance: 3.802919361826042
Shortest Path: [1, 2, 4, 3, 5, 9, 8, 13, 12, 11, 10, 7, 6, 1]
Time it takes to find the shortest path: 21.482 seconds.

Ant Colony Optimization:



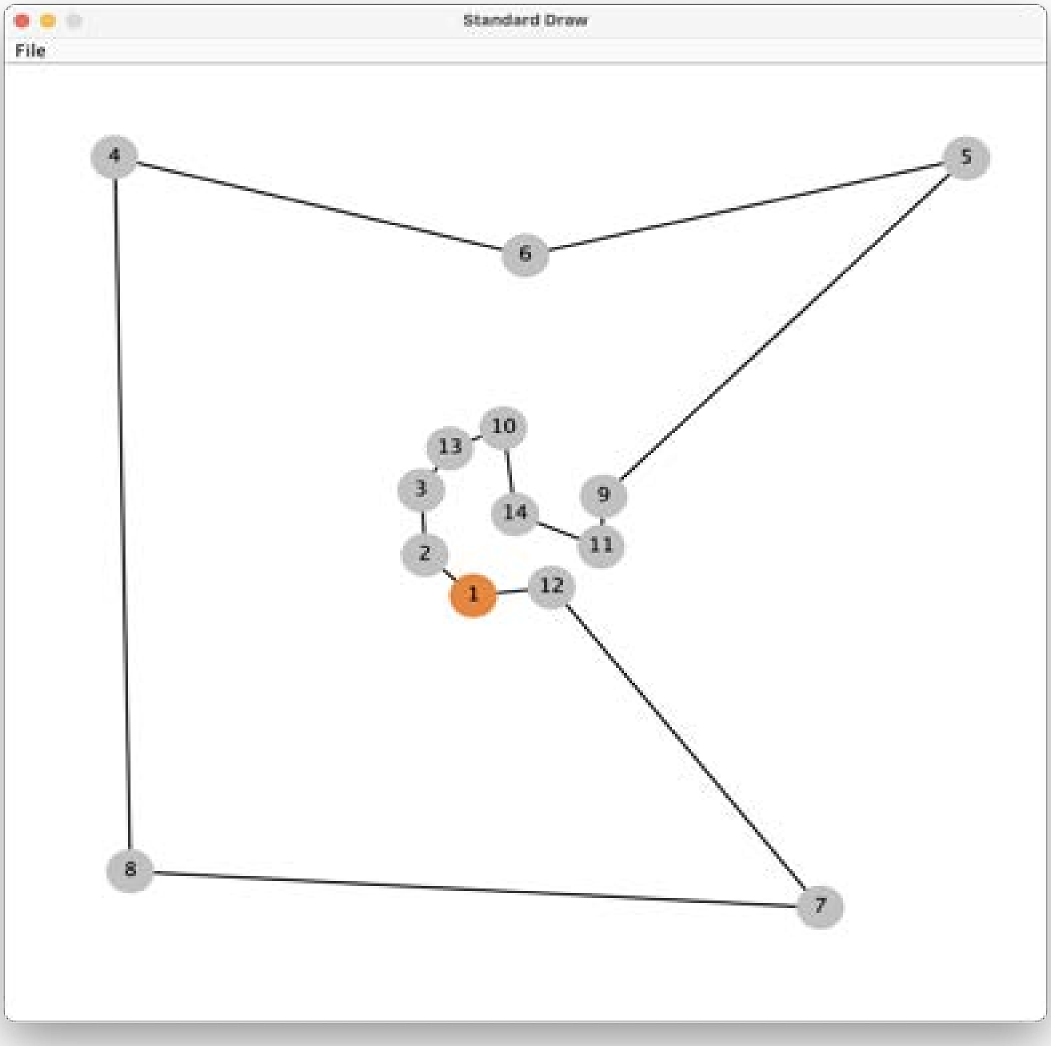


StdOut:

```
Method: Ant Colony Optimization
Shortest Distance: 3.8085283870089066
Shortest Path: [1, 2, 4, 3, 5, 8, 9, 13, 12, 11, 10, 7, 6, 1]
Time it takes to find the shortest path: 0.072 seconds.
```

Input 4:

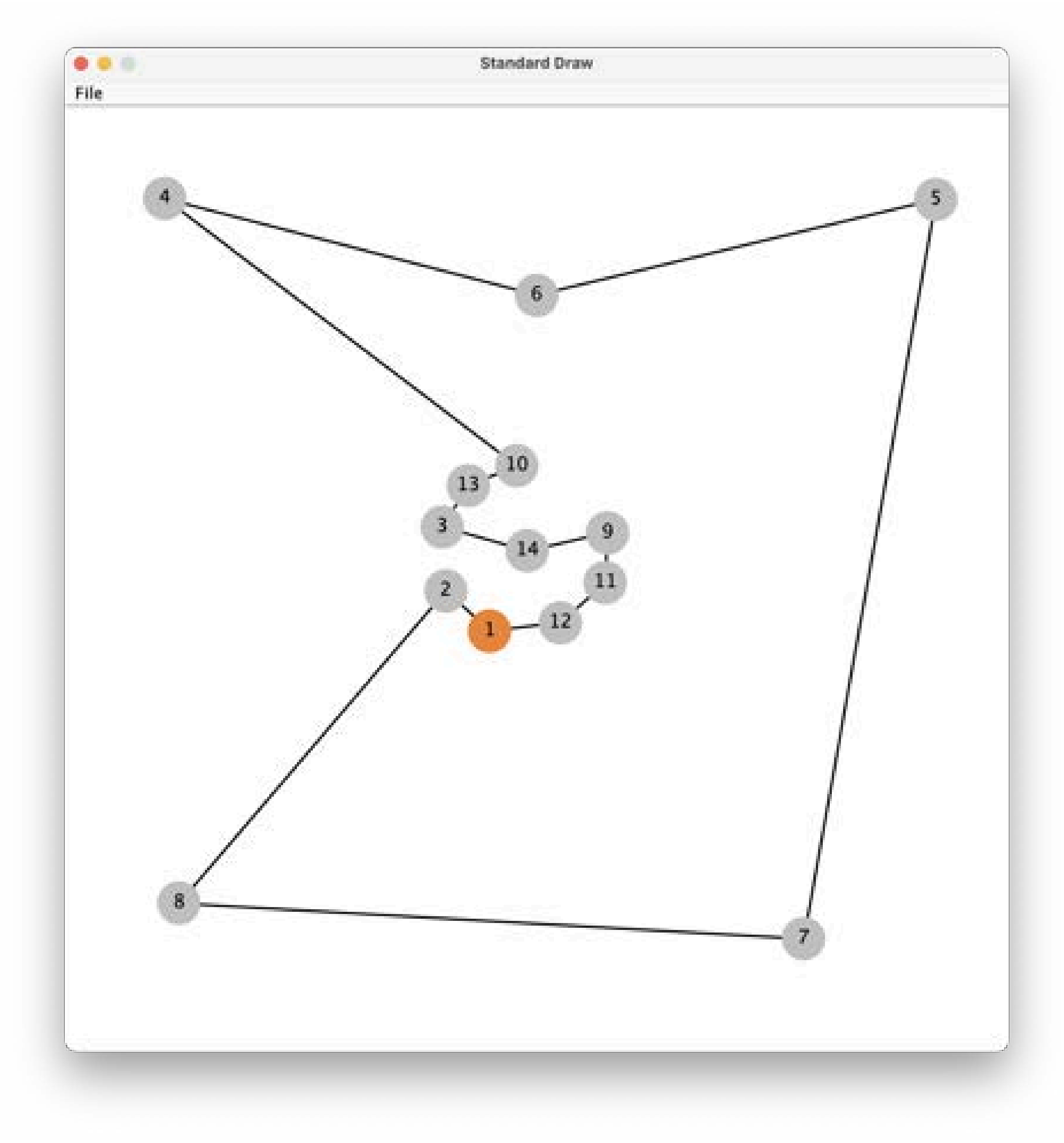
Brute Force:

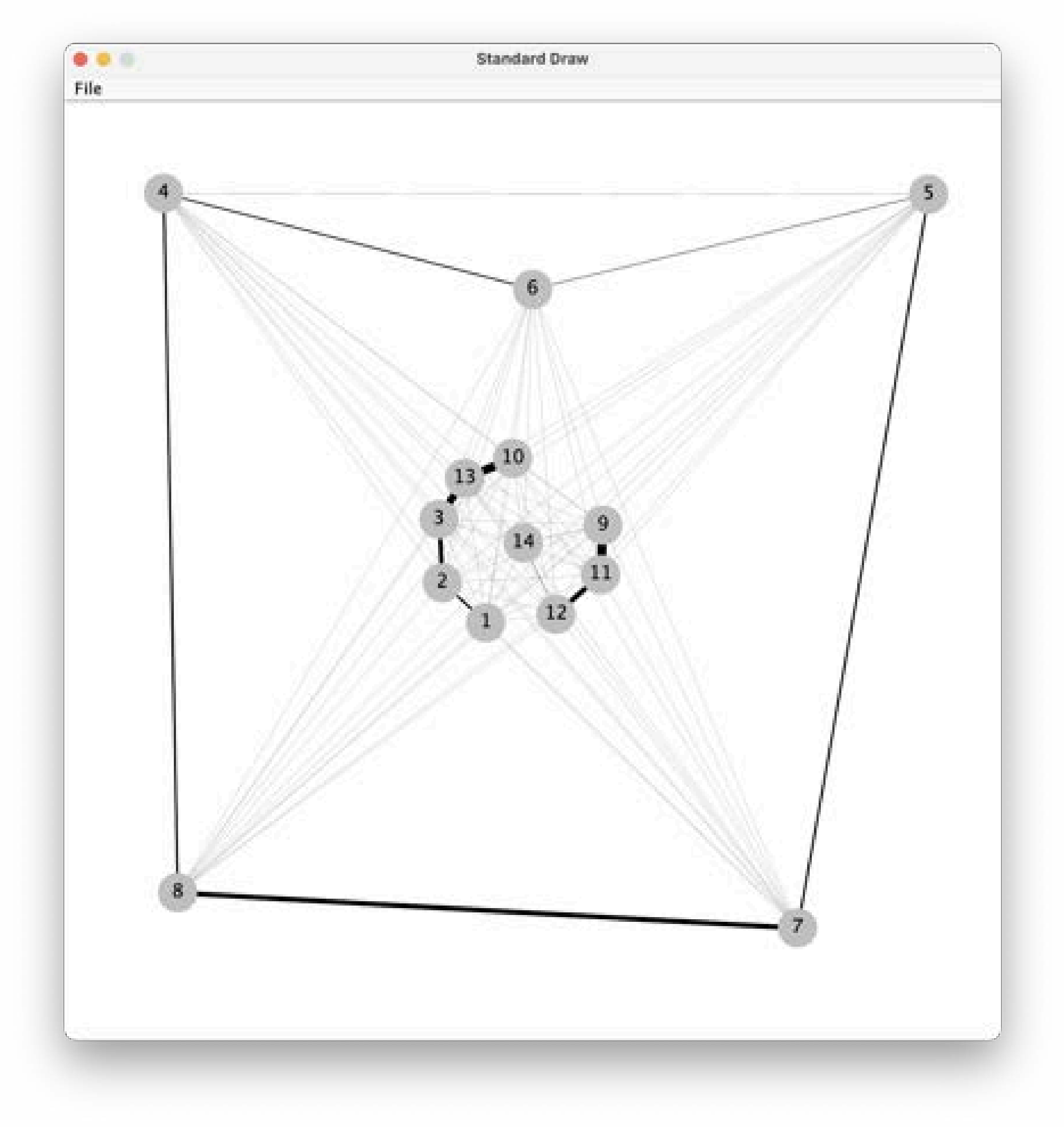


StdOut:

Method: Brute-Force Method
Shortest Distance: 3.710908906673479
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: 310.61 seconds.

Ant Colony Optimization:





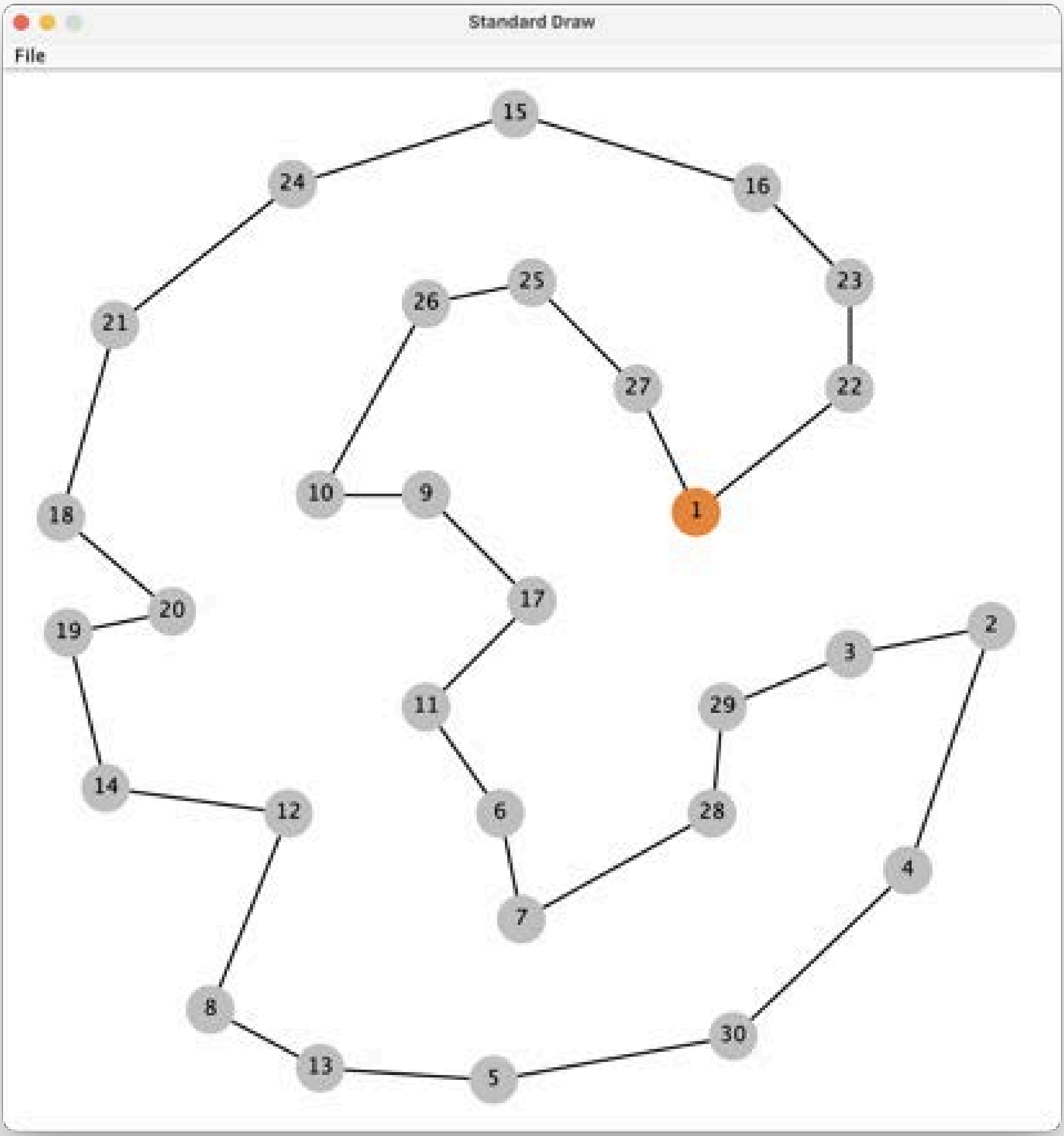
StdOut:

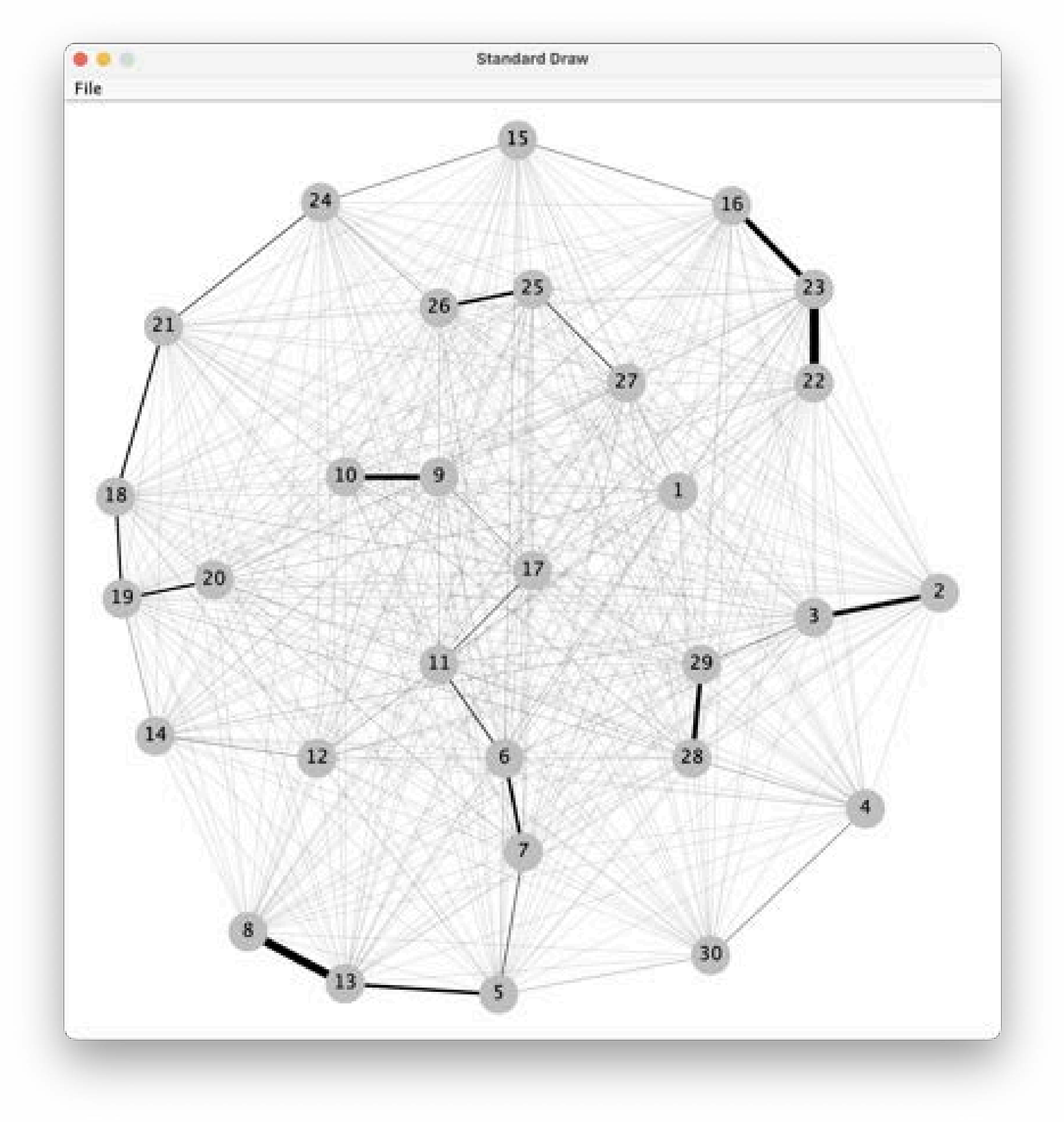
```
Method: Ant Colony Optimization
Shortest Distance: 3.7415223787369336
Shortest Path: [1, 12, 11, 9, 14, 3, 13, 10, 4, 6, 5, 7, 8, 2, 1]
Time it takes to find the shortest path: 0.082 seconds.
```

<u>**Input 5:**</u>

Brute Force: ***Taking too long to compute.***

Ant Colony Optimization:





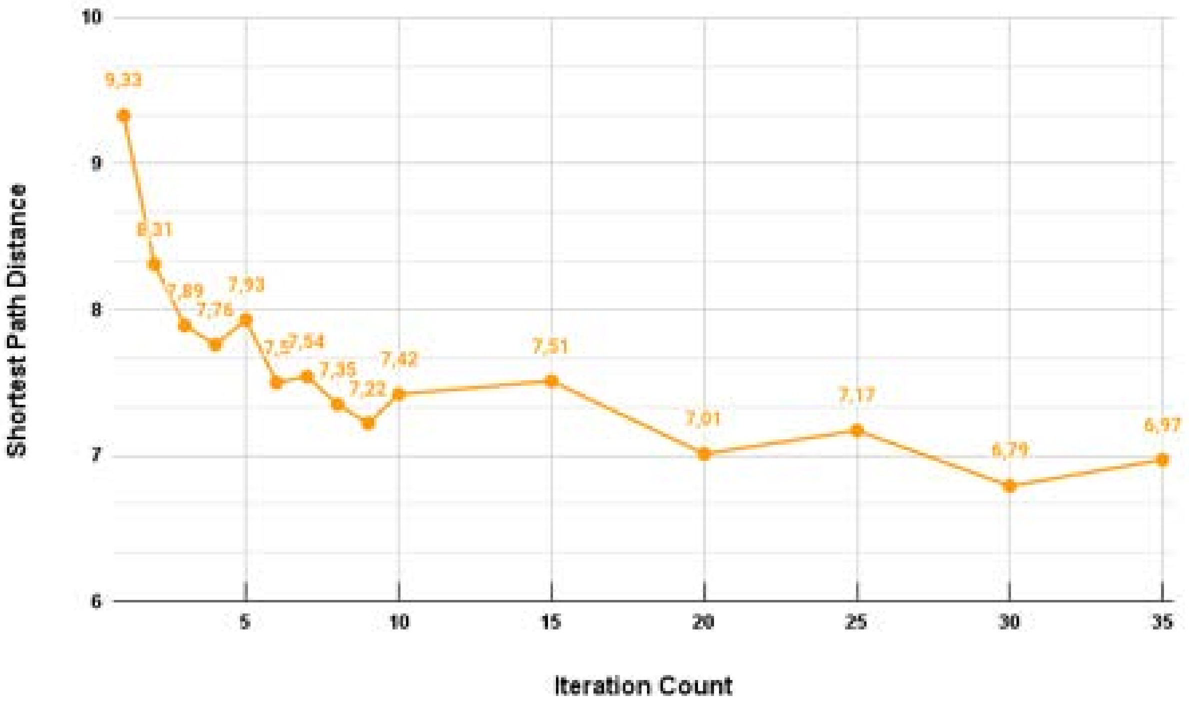
StdOut:

```
Method: Ant Colony Optimization
Shortest Distance: 4.913386292902409
Shortest Path: [1, 3, 29, 28, 7, 6, 11, 17, 9, 10, 26, 24, 21, 18, 19, 20, 14,
12, 8, 13, 5, 30, 4, 2, 22, 23, 16, 15, 25, 27, 1]
Time it takes to find the shortest path: 1.562 seconds.
```

Best Colony Hypermeters:

maximum iteration count = 400
ant count per iteration = 60
degradation factor = 0.7
alpha = 0.8
beta = 1.5
initial pheromone intensity = 0.1
Q value = 0.0001

Input File	***Number of Houses + Migros***	***Brute-Force Time (seconds)***	***Ant Colony Time (seconds)***	***Speed Up Factor***
Input1	11	0.172 (Distance: 1.7952)	0.056 (Distance: 1.7952)	3 times faster
Input2	12	1.707 (Distance: 2.9358)	0.062 (Distance: 2.9358)	28 times faster
Input3	13	21.509 (Distance: 3.8029)	0.072 (Distance: 3.8085)	299 times faster
Input4	14	298.896 (Distance: 3.7109)	0.083 (Distance: 3.7371)	3453 times faster
Input5	30	Too Long (Est. 8,6e23)	0.346 (Distance: 4.8978)	Too big



Ant Colony Optimization (ACO) is a technique used in optimization problems based on the behavior of ants finding the shortest path to food. It's effective for problems like network routing and task scheduling. Here are its simplified advantages and disadvantages:

Advantages:

-**Robustness:** Adapts to changes by updating pheromone levels, enhancing its stability.

-**Positive Feedback:** Promotes good behaviors through pheromone mechanisms, speeding up optimal solution finding.

-**Decentralized Approach:** Operates without central control, using local information to solve complex issues collectively.

-**Parallelism:** Supports multiple independent agents, beneficial for parallel computing and faster results.

-**Versatility:** Applicable to various static and dynamic problems, adaptable to different optimization challenges.

-**Component Interaction:** Manages interactions between various problem components effectively.

Disadvantages:

- **Slow Convergence:** This can be slow on complex problems as it stabilizes pheromone levels over many iterations.
- **Overhead Costs:** Resource-intensive, requiring significant maintenance of pheromone trails.
- **Dependence on Parameters:** Performance heavily relies on precise parameter settings, needing experimentation or expertise.
- **Stagnation:** Risk of premature convergence to suboptimal solutions if diversity in search isn't maintained.
- **Scalability Issues:** Computational and memory demands increase with problem size, limiting use in very large-scale scenarios.
- **Problem-Specific Tweaks**: Often requires specific modifications for optimal performance, complicating its implementation.

External Resources

-https://www.baeldung.com/java-ant-colony-optimization
-https://youtu.be/u7bQomllcJw?si=mndOEkpAZrO4rZNa
-https://chatgpt.com
-https://www.researchgate.net/figure/The-advantages-and-disadvantages-for-ACO-PSO-and-GA-algorithm_tbl2_338141997

Thank you.