```python
import numpy as np
import random

NUM_CITIES = 5  # Number of cities for the TSP
NUM_ANTS = 10   # Number of ants
NUM_ITERATIONS = 100
EVAPORATION_RATE = 0.5
ALPHA = 1.0     # Influence of pheromone
BETA = 2.0      # Influence of distance
Q = 100         # Constant related to pheromone deposit

dist_matrix = np.random.randint(1, 100, size=(NUM_CITIES, NUM_CITIES))

np.fill_diagonal(dist_matrix, 0)

pheromone_matrix = np.ones((NUM_CITIES, NUM_CITIES))

def ant_colony_optimization():
    global pheromone_matrix  # Explicitly declare global usage
    best_route = None
    best_length = float('inf')

    for iteration in range(NUM_ITERATIONS):
        all_routes = []
        all_lengths = []

        for ant in range(NUM_ANTS):
            route = [random.randint(0, NUM_CITIES - 1)]
            while len(route) < NUM_CITIES:
                current_city = route[-1]
                probabilities = []
                for next_city in range(NUM_CITIES):
                    if next_city not in route:
                        pheromone = pheromone_matrix[current_city][next_city] ** ALPHA
                        distance = dist_matrix[current_city][next_city]
                        if distance > 0:
                            distance = distance ** (-BETA)
                        else:
                            distance = 0
                        probabilities.append(pheromone * distance)
                    else:
                        probabilities.append(0)

                total_prob = sum(probabilities)
                if total_prob == 0:
                    break  # No valid move
                probabilities = [p / total_prob for p in probabilities]

                next_city = np.random.choice(range(NUM_CITIES), p=probabilities)
                route.append(next_city)

            # Calculate the length of the route
            route_length = sum([dist_matrix[route[i]][route[i + 1]] for i in range(NUM_CITIES - 1)]) + \
                           dist_matrix[route[-1]][route[0]]  # Closing the loop back to start

            all_routes.append(route)
            all_lengths.append(route_length)

            if route_length < best_length:
                best_length = route_length
                best_route = route

        # Pheromone update
        pheromone_matrix *= (1 - EVAPORATION_RATE)
        for route, route_length in zip(all_routes, all_lengths):
            for i in range(NUM_CITIES - 1):
                pheromone_matrix[route[i]][route[i + 1]] += Q / route_length
            pheromone_matrix[route[-1]][route[0]] += Q / route_length  # Closing the loop

    return best_route, best_length

# Run the ACO
best_route, best_length = ant_colony_optimization()
print(f"Best Route: {best_route}, Length: {best_length}")
```

```
Best Route: [1, 3, 4, 0, 2], Length: 184
```