

EXPERIMENT 10: Travelling Salesman Problem

Aim: implement an Algorithm in Python for solving Travelling Salesman Problem.

Code:

```
import itertools
```

```
def distance(city1, city2):
```

```
    """Calculate Euclidean distance between two cities."""
```

```
    return ((city1[0] - city2[0])**2 + (city1[1] - city2[1])**2)**0.5
```

```
def total_distance(path, cities):
```

```
    """Calculate the total distance of a path."""
```

```
    return sum(distance(cities[path[i]], cities[path[i+1]]) for i in range(len(path)-1)) +  
    distance(cities[path[-1]], cities[path[0]])
```

```
def traveling_salesman(cities):
```

```
    """Find the shortest route that visits each city exactly once."""
```

```
    shortest_distance = float('inf')
```

```
    shortest_path = None
```

```
    for perm in itertools.permutations(range(len(cities))):
```

```
        d = total_distance(perm, cities)
```

```
        if d < shortest_distance:
```

```
            shortest_distance = d
```

```
            shortest_path = perm
```

```
    return shortest_path, shortest_distance
```

```
if __name__ == "__main__":
```

```
# Example usage
```

```
cities = [(0, 0), (1, 2), (3, 1), (5, 3)]
```

```
shortest_path, shortest_distance = traveling_salesman(cities)
```

```
print("Shortest Path:", shortest_path)
```

```
print("Shortest Distance:", shortest_distance)
```

Output:

Shortest Path: (0, 1, 2, 3)

Shortest Distance: 9.055385138137417

Result: Code has been Implemented successfully.