**Experiment 10 Postlab**

**1. How to overcome combinatorial explosion in TSP?**

To overcome combinatorial explosion in TSP, several strategies and algorithms can be employed:

1. **Heuristic Methods**: Instead of exhaustively searching all possible routes, heuristic methods aim to find good solutions quickly by employing rules of thumb or strategies that exploit specific properties of the problem. Examples include Nearest Neighbor, Insertion, and Genetic Algorithms.
2. **Approximation Algorithms**: These algorithms provide solutions that are guaranteed to be close to the optimal solution, but they may not always find the exact optimal solution. Examples include Christofides Algorithm, Lin-Kernighan Algorithm, and Ant Colony Optimization.
3. **Problem Decomposition**: Break down the problem into smaller subproblems that can be solved independently or more efficiently. For example, the problem can be divided into smaller clusters of cities, and solutions for each cluster can be found separately and then combined.
4. **Branch and Bound**: This technique systematically explores the search space by branching into subproblems and bounding the search to eliminate portions of the space that are guaranteed to not contain the optimal solution.
5. **Dynamic Programming**: If the problem exhibits overlapping subproblems, dynamic programming techniques can be used to efficiently solve it by storing and reusing solutions to subproblems.

**2. What is learning from travelling salesperson problem?**

The TSP provides insights into various concepts and techniques in optimization, algorithm design, and complexity theory:

1. **Algorithm Design**: TSP challenges algorithm designers to develop efficient algorithms for solving a notoriously difficult combinatorial optimization problem. This problem serves as a benchmark for evaluating algorithmic performance and scalability.
2. **Heuristics and Approximation**: TSP demonstrates the importance of heuristic methods and approximation algorithms in solving complex optimization problems when exact solutions are impractical or infeasible.
3. **Complexity Theory**: TSP belongs to the class of NP-hard problems, meaning that no polynomial-time algorithm is currently known to solve it optimally. Studying TSP contributes to understanding the boundaries of computationally tractable problems.
4. **Metaheuristic Optimization**: TSP motivates the development and application of metaheuristic optimization techniques, such as genetic algorithms, simulated annealing, and tabu search, which can be generalized to solve a wide range of optimization problems.
5. **Real-world Applications**: Although TSP is a theoretical problem, it has practical applications in various fields, such as logistics, transportation, circuit design, and network routing. Learning from TSP helps in developing efficient solutions to real-world routing and scheduling problems.