ESKİŞEHİR OSMANGAZİ UNIVERSITY

DEPARTMENT OF COMPUTER ENGINEERING

2023-2024 Academic Year

152117127 – Introduction to Heuristic Algorithms

Dr. Öğr. Üyesi Emrah ATILGAN

Prepared by

Göknur Sevinç	152120191013
Hüseyin Emir Leylek	152120191067
Kerem Demircan	152120191089

January 2024

CONTENTS

1. Introduction

Combinatorial optimization problems are often complex and large-scale problems with solutions that are challenging to find. Heuristic algorithms developed to cope with such difficulties have a wide range of practical applications. These algorithms attempt to find effective solutions in their search space by providing specific strategies and intuitive approaches for a particular optimization problem. In this context, this project focuses on the Traveling Salesman Problem over the provinces of the Central Anatolia Region using Tabu Search, Simulated Annealing, and Ant Colony Optimization heuristic algorithms.

The Routing Problems with Optimization Algorithms involve finding the optimal route for a traveler to visit specific cities in the shortest path possible. This algorithm, to be executed over the provinces of the Central Anatolia Region, aims to start from a designated city and traverse all provinces, finding the shortest path and returning to the starting city. This process will be separately conducted with the Tabu Search, Simulated Annealing, and Ant Colony Optimization algorithms, and their performances will be compared.

2. Objective and Significance of the Project

The primary objective of this project is to use different heuristic algorithms to determine the shortest path between cities in the Central Anatolia Region. In the context of Routing Problems with Optimization Algorithms, the goal is to start from a specified city and traverse other cities, finding the shortest path and returning to the starting city. This project aims to evaluate the performance of heuristic algorithms by focusing on a challenging optimization problem frequently encountered in real-world applications.

3. General Information on Route Optimization in the Central Anatolia Region

The decision to exclude Karaman from the route due to its only bordering with Konya poses a challenge for our algorithm. The choice to remove Karaman arises as a practical solution to this difficulty since planning without excluding Karaman leads to the situation of passing through Konya twice in the route. This situation complicates the algorithm's ability to produce accurate and optimized solutions and may even lead to unrealistic results. Therefore, the decision to exclude Karaman from the route is a crucial step in enabling the algorithm to generate more realistic and useful results.

In this context, this project on route optimization in the Central Anatolia Region demonstrates how the algorithm is adapted and results are optimized to address practical challenges. The adjustments made to ensure the algorithm works effectively among cities in this region will contribute to achieving the project's success criteria.


Çankırı
Ankara
Eskişehir
Kırıkkale
Yozgat
Sivas
Kırşehir
Nevşehir
Kayseri
Aksaray
Konya
Niğde
Karaman


Çankırı
Ankara
Eskişehir
Kırıkkale
Yozgat
Sivas
Kırşehir
Nevşehir
Kayseri
Aksaray
Konya
Niğde

4. Algorithm Selection and Motivation

Tabu Search, Simulated Annealing, and Ant Colony Optimization algorithms are commonly preferred heuristic methods for solving optimization problems like Routing Problems with Optimization Algorithms. The selection of these algorithms is based on their different approaches and their successes in the literature.

1) Tabu Search

Tabu Search has shown promising results for Routing Problems. It has been applied not only to problems in the literature but also to real-life problems. The method implemented by Semet and Taillard (1993) and Rochat and Semet (1994) resulted in a 15% reduction in costs. The Tabu Search method with parallel implementations, as proposed by Taillard (1993), successfully solved large-scale problems. Semet and Taillard (1993) addressed the distribution routing problem for 45 different markets in Switzerland, achieving a 10-15% reduction in distribution costs. Rochat and Semet (1994) solved the distribution problem for a large company distributing animal feed and flour in Switzerland, leading to significant cost reduction. The adjustments made using the Tabu Search method in these projects demonstrated improved results in terms of distribution costs and the number of vehicles used.

Basic Principle: Tabu Search is a metaheuristic search algorithm known for its ability to avoid local minima during the problem-solving process. The basic idea can be summarized as moving in the solution space at each step while adhering to specific constraints (tabu list). The algorithm aims to find a better solution than the current one while employing a flexible strategy to explore the solution space effectively.

Parameter Settings:

Tabu List: Determines which solutions will be marked as tabu.

Aspiration Criterion: A tabu solution can cease to be tabu when a specific condition is met.

Move Strategies: Specifies the type and distance of moves to be made.

2) Simulated Annealing

Simulated Annealing was introduced in 1983 by physicists Scott Kirkpatrick, professor Charles Daniel Gelatt, and engineer Mario Vecchi. These researchers discovered this method while trying to develop a metaheuristic algorithm to optimize an energy function.

Scott Kirkpatrick, working at IBM, was exploring the connection between computer science and physics. Inspired by the simulated annealing process in physics, they decided to develop an algorithm that could be used to minimize an energy function.

Basic Principle: Simulated Annealing navigates the solution space using a strategy similar to the heating and cooling process of matter in physics. It starts with a high temperature, accepting possible poor solutions and exploring the solution space more extensively. As the temperature decreases, accepted solutions become more selective, and the algorithm focuses on searching for a closer solution. Simulated Annealing stands out for its ability to explore a wide solution space without getting stuck in local minima.

Parameter Settings:

Initial Temperature: Starting with a high temperature increases the number of acceptable poor solutions.

Cooling Rate: A factor that reduces the temperature over time.

Move Strategies: Strategies that generate random solutions and modify the current solution.

3) Ant Colony Optimization

Marco Dorigo first introduced Ant Colony Optimization (ACO) in his doctoral thesis in 1992. Inspired by the idea that ants follow a certain logic in their movements, Dorigo applied ant colony optimization to optimization problems such as the traveling salesman problem and vehicle routing problem. He conducted experiments using 200 ants, placing various food sources and obstacles to observe each ant's movement.

Basic Principle: Ant Colony Optimization (ACO) draws inspiration from the foraging strategy of ant colonies. Ants can find short paths and most efficient routes by communicating with each other. ACO mimics this natural behavior to find optimal solutions in graph-based problems. ACO tends to optimize a solution path based on the pheromones left by ants. By exploring the solution space and evaluating possible solutions through pheromone release, ACO aims to find the optimal route.

Parameter Settings:

Pheromone Update Rules: Define how the pheromone trails left by ants will be updated.

Number of Ants: Determines the number of ants navigating the solution space and leaving pheromone trails.

Evaporation Rate: Determines the rate at which pheromone trails evaporate over time.

These algorithms, known for their powerful heuristic approaches in solving optimization problems like Routing Problems with Optimization Algorithms, were applied to the provinces of the Central Anatolia Region during the project. The details of these algorithms, including their features and parameter settings, were thoroughly discussed in the project process.

5. Used Technologies

- NumPy (Numerical Python): A powerful library for scientific calculations and working with multi-dimensional arrays. NumPy was used in this program for operations on the distance matrix. NumPy provides vector and matrix operations to optimize mathematical calculations and improve performance.
- Pandas: A library used for data analysis and manipulation. Pandas DataFrame was used in this program to store and organize results. DataFrame was utilized to handle results more neatly and write results to an Excel file.
- Random: The random module from Python's standard library is used for generating random numbers. In this program, the random module was used for creating the initial solution and random selections during the tabu list removal process.
- Sys: The sys module from Python's standard library is used for system-related operations. In this program, sys.maxsize constant was used to set the initial best distance value to infinity.

6. Results

The results of the three algorithms that we optimized within the scope of our project are available in the report. The purpose of the algorithms is to travel all the provinces in the region, starting from the starting city received from the user, and return to the starting city by the shortest route. When the tables are examined, the best results in three different algorithms are seen for each province. According to these results, in the shortest path optimization. It has been observed that the most successful algorithm is the Tabu Search algorithm. If we were to rank the success, it can be listed as Tabu Search>Simulated Annealing>Ant Colony. When the execution time averages are calculated for each algorithm, they can be listed from fast to slow as Ant Colony<Tabu Search<Simulated Annealing. The resulting execution time The averages are 0.53 seconds for Ant Colony, 0.95 seconds for Tabu Search and 1.15 seconds for Simulated Annealing.

SIMULATED ANNEALEING		
Route	**Total Distance**	**Best**
Aksaray -> Niğde -> Nevşehir -> Kayseri -> Sivas -> Yozgat -> Kırşehir -> Kırıkkale -> Çankırı -> Ankara -> Eskişehir -> Konya -> Aksaray	1886	Yes
Ankara -> Çankırı -> Kırıkkale -> Kırşehir -> Yozgat -> Sivas -> Kayseri -> Nevşehir -> Niğde -> Aksaray -> Konya -> Eskişehir -> Ankara	1885	Yes
Çankırı -> Kırıkkale -> Kırşehir -> Yozgat -> Sivas -> Kayseri -> Nevşehir -> Niğde -> Aksaray -> Konya -> Eskişehir -> Ankara -> Çankırı	1885	Yes
Eskişehir -> Ankara -> Çankırı -> Kırıkkale -> Kırşehir -> Yozgat -> Sivas -> Kayseri -> Nevşehir -> Niğde -> Aksaray -> Konya -> Eskişehir	1885	Yes
Kayseri -> Sivas -> Yozgat -> Kırşehir -> Kırıkkale -> Çankırı -> Ankara -> Eskişehir -> Konya -> Aksaray -> Niğde -> Nevşehir -> Kayseri	1886	Yes
Kırıkkale -> Çankırı -> Ankara -> Eskişehir -> Konya -> Aksaray -> Niğde -> Nevşehir -> Kayseri -> Sivas -> Yozgat -> Kırşehir -> Kırıkkale	1886	Yes
Kırşehir -> Yozgat -> Sivas -> Kayseri -> Nevşehir -> Niğde -> Aksaray -> Konya -> Eskişehir -> Ankara -> Çankırı -> Kırıkkale -> Kırşehir	1885	Yes
Konya -> Aksaray -> Niğde -> Nevşehir -> Kayseri -> Sivas -> Yozgat -> Kırşehir -> Kırıkkale -> Çankırı -> Ankara -> Eskişehir -> Konya	1886	Yes
Nevşehir -> Kayseri -> Sivas -> Yozgat -> Kırşehir -> Kırıkkale -> Çankırı -> Ankara -> Eskişehir -> Konya -> Aksaray -> Niğde -> Nevşehir	1886	Yes
Niğde -> Aksaray -> Konya -> Eskişehir -> Ankara -> Çankırı -> Kırıkkale -> Kırşehir -> Yozgat -> Sivas -> Kayseri -> Nevşehir -> Niğde	1885	Yes
Sivas -> Kayseri -> Nevşehir -> Niğde -> Aksaray -> Konya -> Eskişehir -> Ankara -> Çankırı -> Kırıkkale -> Kırşehir -> Yozgat -> Sivas	1885	Yes
Yozgat -> Sivas -> Kayseri -> Nevşehir -> Niğde -> Aksaray -> Konya -> Eskişehir -> Ankara -> Çankırı -> Kırıkkale -> Kırşehir -> Yozgat	1885	Yes

ANT COLONY OPTIMIZATION		
Route	**Total Distance**	**Best**
Aksaray -> Niğde -> Nevşehir -> Kayseri -> Sivas -> Kırşehir -> Yozgat -> Çankırı -> Kırıkkale -> Ankara -> Eskişehir -> Konya -> Aksaray	1996	Yes
Ankara -> Çankırı -> Kırıkkale -> Kırşehir -> Yozgat -> Sivas -> Niğde -> Kayseri -> Nevşehir -> Aksaray -> Konya -> Eskişehir -> Ankara	2019	Yes
Çankırı -> Kırıkkale -> Ankara -> Eskişehir -> Konya -> Aksaray -> Kırşehir -> Kayseri -> Nevşehir -> Niğde -> Sivas -> Yozgat -> Çankırı	2025	Yes
Eskişehir -> Ankara -> Kırıkkale -> Çankırı -> Kırşehir -> Yozgat -> Sivas -> Kayseri -> Nevşehir -> Niğde -> Aksaray -> Konya -> Eskişehir	1937	Yes
Kayseri -> Kırşehir -> Yozgat -> Kırıkkale -> Çankırı -> Ankara -> Eskişehir -> Konya -> Aksaray -> Niğde -> Nevşehir -> Sivas -> Kayseri	2025	Yes
Kırıkkale -> Çankırı -> Yozgat -> Sivas -> Kayseri -> Nevşehir -> Kırşehir -> Aksaray -> Niğde -> Konya -> Eskişehir -> Ankara -> Kırıkkale	1971	Yes
Kırşehir -> Nevşehir -> Yozgat -> Sivas -> Kayseri -> Niğde -> Aksaray -> Konya -> Eskişehir -> Ankara -> Çankırı -> Kırıkkale -> Kırşehir	1986	Yes
Konya -> Niğde -> Aksaray -> Nevşehir -> Kayseri -> Sivas -> Yozgat -> Kırşehir -> Çankırı -> Kırıkkale -> Ankara -> Eskişehir -> Konya	2012	Yes
Nevşehir -> Kayseri -> Niğde -> Aksaray -> Konya -> Eskişehir -> Ankara -> Kırıkkale -> Çankırı -> Sivas -> Yozgat -> Kırşehir -> Nevşehir	2064	Yes
Niğde -> Nevşehir -> Kayseri -> Kırşehir -> Aksaray -> Konya -> Eskişehir -> Ankara -> Kırıkkale -> Çankırı -> Yozgat -> Sivas -> Niğde	2025	Yes
Sivas -> Kayseri -> Kırşehir -> Niğde -> Nevşehir -> Aksaray -> Konya -> Eskişehir -> Ankara -> Kırıkkale -> Çankırı -> Yozgat -> Sivas	1963	Yes
Yozgat -> Kırşehir -> Kırıkkale -> Çankırı -> Ankara -> Eskişehir -> Aksaray -> Konya -> Niğde -> Nevşehir -> Kayseri -> Sivas -> Yozgat	2075	Yes

TABU SEARCH ALGORITHMS		
Route	**Total Distance**	**Best**
Aksaray -> Eskişehir -> Ankara -> Çankırı -> Kırıkkale -> Kırşehir -> Yozgat -> Sivas -> Kayseri -> Nevşehir -> Niğde -> Konya -> Aksaray	1678	Yes
Ankara -> Eskişehir -> Konya -> Aksaray -> Niğde -> Nevşehir -> Kayseri -> Sivas -> Yozgat -> Kırşehir -> Kırıkkale -> Çankırı -> Ankara	1650	Yes
Çankırı -> Eskişehir -> Konya -> Aksaray -> Niğde -> Nevşehir -> Kayseri -> Sivas -> Yozgat -> Kırşehir -> Kırıkkale -> Ankara -> Çankırı	1621	Yes
Eskişehir -> Konya -> Aksaray -> Niğde -> Nevşehir -> Kayseri -> Sivas -> Yozgat -> Kırşehir -> Kırıkkale -> Çankırı -> Ankara -> Eskişehir	1556	Yes
Kayseri -> Sivas -> Yozgat -> Kırşehir -> Kırıkkale -> Çankırı -> Ankara -> Eskişehir -> Konya -> Aksaray -> Niğde -> Nevşehir -> Kayseri	1689	Yes
Kırıkkale -> Eskişehir -> Konya -> Aksaray -> Kırşehir -> Nevşehir -> Niğde -> Kayseri -> Sivas -> Yozgat -> Çankırı -> Ankara -> Kırıkkale	1693	Yes
Kırşehir -> Eskişehir -> Ankara -> Kırıkkale -> Çankırı -> Yozgat -> Sivas -> Kayseri -> Nevşehir -> Niğde -> Konya -> Aksaray -> Kırşehir	1667	Yes
Konya -> Eskişehir -> Ankara -> Çankırı -> Kırıkkale -> Kırşehir -> Yozgat -> Sivas -> Kayseri -> Nevşehir -> Niğde -> Aksaray -> Konya	1555	Yes
Nevşehir -> Eskişehir -> Ankara -> Çankırı -> Kırıkkale -> Kırşehir -> Yozgat -> Sivas -> Kayseri -> Niğde -> Konya -> Aksaray -> Nevşehir	1715	Yes
Niğde -> Eskişehir -> Konya -> Aksaray -> Kırşehir -> Kırıkkale -> Ankara -> Çankırı -> Yozgat -> Sivas -> Kayseri -> Nevşehir -> Niğde	1657	Yes
Sivas -> Eskişehir -> Ankara -> Kırıkkale -> Çankırı -> Yozgat -> Kırşehir -> Aksaray -> Konya -> Niğde -> Nevşehir -> Kayseri -> Sivas	1558	Yes
Yozgat -> Eskişehir -> Ankara -> Çankırı -> Kırıkkale -> Kırşehir -> Aksaray -> Konya -> Niğde -> Nevşehir -> Kayseri -> Sivas -> Yozgat	1659	Yes

7. References

- https://www.kgm.gov.tr/sayfalar/kgm/sitetr/root/uzakliklar.aspx
- https://www.buraksenyurt.com/post/floyd-warshall-algoritmasi-ile-en-kisa-yolu-bulmak
- https://dergipark.org.tr/tr/download/article-file/46729
- Kılıç, Ecem Nükte. Araç rotalama probleminin sezgisel bir algoritma ile çözülmesi: Bir boya fabrikasında uygulama. MS thesis. Sakarya Üniversitesi, 2020.
- Helshani, L. A. U. R. I. K. "Solving the traveling salesman problem using google services and simulated annealing algorithm." European Academic Research 4.3 (2016): 2321-30.
- Ali, Zeravan Arif. Concentric tabu search algorithm for solving traveling salesman problem. MS thesis. Eastern Mediterranean University (EMU)-Doğu Akdeniz Üniversitesi (DAÜ), 2016.