RELATIVE DISTANCES APPROACH FOR MULTI-TRAVELING SALESMEN PROBLEM

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

EMRE ERGÜVEN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

JANUARY 2024

Approval of the thesis:

RELATIVE DISTANCES APPROACH FOR MULTI-TRAVELING SALESMEN PROBLEM

submitted by **EMRE ERGÜVEN** in partial fulfillment of the requirements for the degree of **Master of Science in Computer Engineering Department, Middle East Technical University** by,

Date:15.01.2024

Name, Surname:	Emre Ergüven
Signature :	

ABSTRACT

RELATIVE DISTANCES APPROACH FOR MULTI-TRAVELING SALESMEN PROBLEM

Ergüven, Emre
M.S., Department of Computer Engineering
Supervisor: Prof. Dr. Faruk Polat

January 2024, 80 pages

This study aims to find a solution for the Multi-Traveling Salesman Problem (M-TSP). Within the problem, multiple tasks (e.g cargo delivery, warehouse placement) are executed by multiple agents (e.g traveling salesman, autonomous robots). There are two main objectives for these problems; the first one is minimizing the total path cost, and the second one is minimizing the maximum cost of salesmen (makespan).

We mainly focused on minimizing the total cost. But fully focusing on decreasing the total cost mostly results with an increase on the makespan. Our method keeps the makespan in a reasonable range. Due to the combinatorial structure of the problem, finding the cost-optimal solutions is impossible (with current conditions). Solutions must be found quickly in order to be applicable in real-life. So, it can be said that the third objective of the problem is reducing the complexity and time to find the solutions.

The MTSP problem is generally tried to be solved in two separate phases. In the first phase, tasks are assigned to salesmen with different approaches (e.g K-Means, DBSCAN). Second phase is finding optimal routes for each salesman. The problem within the second stage is identical to the Traveling Salesman Problem (TSP). Our relative distance model combines these phases within one method with a novel heuristic approach. With our model, tasks can be easily added and removed from the problem space and live-scheduling can be enabled.

All of these methods mentioned are implemented on C++ and visualized on Python

Keywords: Traveling Salesman Problem, Multi Traveling Salesman Problem, Combinatorial Optimization, Heuristic Methods, Task Assignment

ÖZ

ÇOKLU GEZGİN SATICI PROBLEMİ İÇİN GÖRELİ MESAFELER YAKLAŞIMI

Ergüven, Emre
Yüksek Lisans, Bilgisayar Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. Faruk Polat

Ocak 2024 , 80 sayfa

Bu çalışma, Çoklu Gezgin Satıcı Problemini çözmeyi amaçlamaktadır. Bu problemlerde, görevler (kargo teslimatı, depo yerleştirimi) birden çok etmen (gezgin satıcı, otonom robot) tarafından tamamlanmaya çalışılır. Bu gibi problemlerde iki ana hedef vardır; birincisi toplam kat edilen mesafeyi minimize etmektir, ikincisi de bir etmen tarafından kat edilen maksimum mesafeyi minimize etmektir.

Bu çözümdeki ana öncelik toplam kat edilen mesafeyi minimize etmektir. Fakat tamamen toplam kat edilen mesafeyi minimize etmeye odaklanmak, bir etmen tarafından kat edilen maksimum mesafeyi artırabilir. Bu çalışmada sunulan yöntem, bir etmen tarafından kat edilen maksimum mesafeyi de makul bir seviyede tutmaktadır. Problemin kombinatoryal yapıda olması sebebiyle maliyeti minmize eden bir çözümü bulmak imkansızdır (günümüz koşullarında). Gerçek hayatta uygulanabilirlik için çözümün hızlı bir şekilde bulunması gerekmektedir. Yani, şu söylenebilir ki; problemin üçüncü hedefi, çözüm bulunana kadarki karmaşıklığı ve harcanan zamanı düşürmektir.

Çoklu gezgin satıcı problemi, genellikle iki ayrı aşamada çözülmeye çalışılır. İlk aşamada görevler kullanıcılara farklı yaklaşımlarla verilir (k-ortalamalar kümesi, yoğunluk tabanlı mekansal uygulamaların gürültüyle kümelenmesi). İkinci aşama ise her gezgin için verilen görevlerin optimal sıralamasıdır. İkinci aşamadaki problem gezgin satıcı problemiyle aynıdır. Göreli Mesafe modelimiz bu fazları tek bir yöntemde özgün bir keşifsel yaklaşımla birleştirir. Modelimiz sayesinde görevler kolayca iptal edilebilir veya yeni görevler eklenebilir ve canlı planlama sağlanabilir.

Yukarıda belirtilen tüm metodlar C++'da çalıştırılmış ve Python'da görselleştirilmiştir.

Anahtar Kelimeler: Gezgin Satıcı Problemi, Çoklu Gezgin Satıcı Problemi, Kombinatoryal Optimizasyon, Buluşsal Yöntemler, Görev Tayini

To all victims of 6th February Earthquake

ACKNOWLEDGMENTS

I would like to thank Prof Dr. Faruk Polat for his constant support and guidance. He motivated me to study on this inspirational problem and showed his confidence on my skills.

Also, my deepest appreciation goes to my wife, Emine Aksel Ergüven. Without her, I couldn't have starting motivation to M.S and couldn't complete the thesis. She never make me feel alone. In these challenging times, her love and energy kept me going.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ABBREVIATIONS

TSP	Traveling Salesman Problem
MTSP	Multi Traveling Salesman Problem
RDA	Relative Distances Approach

CHAPTER 1

INTRODUCTION

1.1 Motivation and Problem Definition

Combinatorial Optimization[1] is one of the problems in Computer Science where the objective is to find the best combination, arrangement or picks of discrete elements from a finite set under problem-specific constraints. Some of the real-world combinatorial optimization problems are Traveling Salesman Problem[2], Job-Shop Scheduling Problem, Maximum Cut Problem and Task Assignment Problem.

Combinatorial Optimization Problems are hard to solve because of some reasons. Firstly; most of the combinatorial optimization problems are NP-hard, it means that the optimality of a solution cannot be guaranteed in a polynomial time. Secondly, the solution space is exponentially growing with the size of problem and it makes the problem unsolvable in a reasonable duration. Thirdly, there can be more than one objectives that should be satisfied but mostly these objectives are conflicting like minimizing the total cost and maximizing the resource utilization. Lastly, gathering and processing the data in real-life combinatorial optimization problems mostly involve the uncertainties, noises and the like.

Some combinatorial optimization problems are the mixture of other combinatorial optimization problems like Supply Chain Optimization (facility location problem and network optimization problem) or Multi Traveling Salesman Problem (task assignment problem and traveling salesman problem). The subject of this study is finding efficient solutions to Multi Traveling Salesman Problem with low levels of complexity.

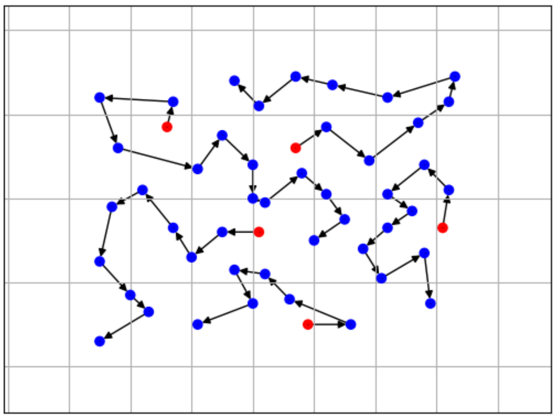


Figure 1.1: sample M-TSP instance with a solution

The Multi Traveling Salesman Problem (M-TSP) is an extended version of the Traveling Salesman Problem (TSP) in which the ideal visit order of destinations is tried to be determined. In this version, N destination points must be visited by k salesmen, regardless of whether they are in the same location before departure. The problem has similar objectives and constraints with the Multi Agent Path Finding with Multiple Deliveries (MAPF-MD) problem. The MAPF-MD is the generalized version of the Multi Agent Path Finding (MAPF) problem in which each agent is matched with a single task. (one to one correspondence between agents and target nodes). It can be said that the biggest difference between MAPF-MD and MTSP is that the paths in the MAPF-MD problem must be collision-free. This restriction is not prioritized in MTSP.

Various real-life scenarios that can be represented as an MTSP instance. Mobile e-charging stations, taxi networks, healthcare services and scenarios in which numerous goals have to be visited by multiple salesperson, vehicle etc. can be considered as a MTSP instance. Applying the MTSP concepts can be crucial for Disaster Management in different areas as follows:

- Reconnaissance activities to explore extent of the damage
- Search and Rescue Organizations

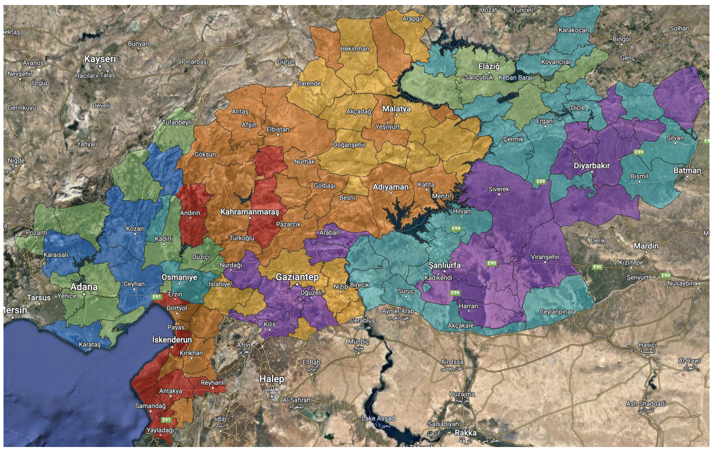


Figure 1.2: Towns affected from the 6 February Earthquake

- Distribution of Emergency Supplies (water, food etc.)
- Evacuation Planning

We lost many people in the 6 February Earthquake[3] due to lack of planning and organization. If the problem had been considered with MTSP principles and the solutions were tried in the earthquake drills, far fewer people would die.

Two ultimate motivations lie behind this study. The first one is the improvable environment of the problem. MTSP literature is relatively newer than the other combinatorial optimization problems and this situation led us to think creative and find novel solutions. Second one is the real-life applicability of the problem. Creative ideas and novel approaches for this problem may result better solutions for the problems that people face in reality.

1.2 Proposed Methods and Models

Heuristic algorithms are well suited to deal with the challenges of Combinatorial Optimization Problems because heuristics quickly provide approximate solutions. As the size of the samples increases, the importance of speed will become more evident.

There are common heuristics for combinatorial optimization problems such as genetic algorithms or greedy algorithms. In this problem, we used a greedy heuristic to initialize the solution. After initialization, suitable improvement heuristics are applied and the final solution is ready.

The Multi Traveling Salesman Problem is generally tried to be solved in two separate stages. First, tasks are distributed to each salesman, and then corresponding paths are created with the given tasks. In this study, these two stages are combined with a new heuristic which will be presented in the following sections.

1.3 Contributions and Novelties

Our contributions are as follows:

- The Multi-Traveling Salesman Problem has been relatively received less attention than other combinatorial optimization problems. This study will contribute to the enrichment of the existing MTSP literature.
- Alongside the literature enhancement, the study will also applicable to the real-life problems because the solution method doesn't require a well structured and complex data.
- The study and the solution is extendable to the other complex variants of problem like Prioritized Multi Traveling Salesman Problem

The Novelties are as follows:

- The solution approach in the study can also be applied into TSP and it also results well.
- The methodology presented in this study is suitable to different kinds of MTSP scenarios.
  - All salesmen can start from different locations.
  - All salesmen can start from the same location.
  - While some salesmen start from same location, others can start from different locations.

- Most of the studies for the MTSP splits the problem into two stages, our approach finds the solution within only one stage.
- Most studies on combinatorial optimization problems involve randomness in their structure. But our method doesn't contain any random factor within its algorithm. It provides consistency.

1.4 The Outline of the Thesis

The thesis aims to introduce a novel heuristic method that efficiently finds solutions with near-optimal and relatively reasonable makespan values.

Chapter 2 starts with the brief introduction of problems which will be analyzed and tried to be solved in the next chapters. After the introduction step, backgrounds of the methods and algorithms are categorized and explained the reasons behind whether they used in our approach or not.

Chapter 3 presents the related articles and works that covers the problems, algorithms or approaches mentioned within the thesis. The chapter is categorized and organized according the topics they covered.

In the fourth chapter mathematical and verbal descriptions of the problem are represented. These descriptions cover the constraints and the objectives of the problem. After the description part, pseudo codes are given with the example instances of the problem. At the end of chapter, brute force algorithm's pseudo-codes are shown to provide benchmarks in the experimental results and evaluations part.

Performance of the solutions provided by Relative Distances Approach is evaluated by different methods in fifth chapter;

- First, outputs are compared with Wang's [4], Ndiaye's [5] and Lou's [6] research.
- For the smaller problem instances, the outputs are evaluated by the complete brute force algorithm which all possible task assignment and sequencing scenarios are examined.

- In the moderate size problem instances, the results are evaluated by the partial brute force algorithm which looks at the alternative sequencing options only.

- The extreme-size instances are visualized and evaluated by their intersection-status.

Chapter 6 provides the brief conclusion and the targeted future works that extended from this thesis are presented.

CHAPTER 2

BACKGROUND

In this section, TSP, MTSP and some other problems that closely related to MTSP are briefly described and visualized to make descriptions clearer. After the description, backgrounds of methods and algorithms used to solve MTSP are given.

2.1 TSP

The Traveling Salesman Problem (TSP) is well-known combinatorial optimization problem in Computer Science and Operations Research. In this problem, set of cities and the distances between them are given. The objective is to find the shortest possible path/tour that covers all cities starting from a determined city. The problem is an NP-Hard (Non-Deterministic Polynomial-Time Hard) problem, which means that the problem becomes extremely difficult to solve as the number of cities increases. Finding the best solutions for larger instances is computationally challenging.

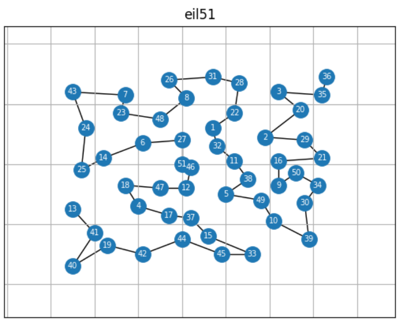


Figure 2.1: sample TSP instance proposed in TSP-LIB with a solution

Let n be the number of cities. d_{ij} is the distance between the cities i and j. x_{ij} is the Boolean variable that shows whether the path i to j discovered in the route of the Traveling Salesman. Mathematical formulation of the TSP is as follows:

$$\underset{i \neq j}{x_{ij} \in \{0, 1\}} \tag{2.1}$$

$$\sum_{\substack{i=1 \\ i \neq j}}^{n} x_{ij} = 1 \tag{2.2}$$

$$\sum_{\substack{j=1 \\ j \neq i}}^{n} x_{ij} = 1 \tag{2.3}$$

$$\sum_{\substack{i \in S \\ i \neq j}} \sum_{\substack{j \in S \\ S \subset N}} x_{ij} \leq \mid S \mid -1 \quad {\scriptstyle 2 \leq |S| \leq n-1} \tag{2.4}$$

Each decision variable that represents whether the path is selected or not can only be 0 or 1 (2.1).

Constraint 2.2 states that each city can have only one path which departs from itself. Analogous to 2.2, 2.3 states that each city can have only one path which arrives to itself.

Some extensions of TSP don't require the Hamiltonian Cycle. The salesman can finish the path at a node different from where s/he started. This kind of an extension will be the subject of the main problem of thesis.

2.4 is a constraint that prevents sub-tours within the problem space.

$$\sum_{i=1}^{n} \sum_{\substack{j=1 \\ i \neq j}}^{n} d_{ij} * x_{ij} \tag{2.5}$$

The cost function of the TSP can be formulated as shown in 2.5. Minimization of this function is the ultimate objective of the problem.

2.2 MTSP

The Multi Traveling Salesman Problem (MTSP) is an extension of the Traveling Salesman Problem (TSP). In the regular definition of the MTSP, there are multiple salesmen that starts from the same depot and the ultimate objective is to minimize the total distance covered without leaving any city unvisited. The approach presented in this thesis finds solutions for all types of MTSP problems in terms of depot distribution. Salesmen can start from whether the same depot or different depots or a combination of both scenarios (some from same some from different depots). The classical definition points the total distance minimization as an objective but minimizing the maximum distance covered by a salesman also targeted in some studies.

Related mathematical expressions of the MTSP will be given in the next sections with the possible variants.

2.3 MAPF

The Multi Agent Path Finding is a problem that aims to find collision-free paths from the agents' starting locations to their assigned targets in a shared space.

The problem can also have some additional constraints like avoiding collisions with the moving obstacles or speed/resource limitations.

The Multi Agent Path Finding with Multiple Delivery Locations (MAPF-MD) is an extension of the MAPF that agents have more than one target locations and decide the order of visit. In this variant, finding collision-free paths are also prioritized like MAPF.

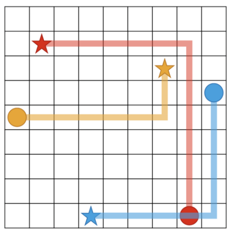


Figure 2.2: sample MAPF instance

2.4 VRP

The Vehicle Routing Problem is a combinatorial optimization problem that targets to optimize costs in parallel with the demand satisfaction goals. In the problem, vehicles start from the central depot and distributes the goods to the customers. There can be various constraints like truck capacities, delivery deadlines etc.

Both MTSP and VRP focus on the minimizing the total cost, their differences come from the constraints and application areas.

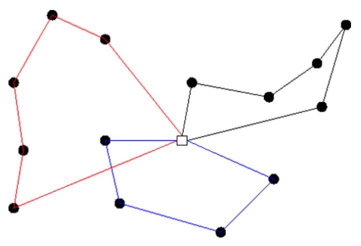


Figure 2.3: sample VRP instance

2.5 Brute Force Approach

The Brute Force Approach is a clear and exhaustive strategy to solve the problems. The approach systematically looks at the all possible combinations/permutations to find the optimal solution of a problem. It guarantees the optimality of a solution and can be beneficial problems with small size of problem instances. If the brute force approach is selected to solve MTSP, the complexity of solution becomes $(N!)^m$. (N is the number of cities and m is the number of salesmen). For a problem instance with 15 cities and 3 salesmen, approximate number of iterations will be $(15!)^3 = 2.2^{36}$ by brute force approach.

2.6 Exact Algorithms

Exact Algorithms are designed to solve a computational problem with the guarantee of optimality. These algorithms are also exploring the entire solution space to ensure that the optimal solution is found. These algorithms are closer to brute force approach than the heuristic methods. Here are the some exact algorithms can deployed to solve MTSP.

2.6.1 Integer Linear Programming

Integer Linear Programming is an approach that converts combinatorial optimization problems to linear programming models with some constraints to assign integer values to variables. This type of method requires an ILP solver like CPLEX or Gurobi. It guarantees the optimality of a solution but the complexity grows exponentially with the increasing number of cities and salesmen.

2.6.2 Branch and Bound

The Branch and Bound algorithm is an approach that uses the problem through divide and conquer approach. It systematically splits the problem into the smaller subproblems, finds optimal solutions these smaller ones and connects the sub-solutions to generate a solution to the main problem. If there is no time limitation, it will find the global optimal solution. It can not be applicable in real-world problems with larger instances.

2.7 Heuristic Methods

Heuristic methods are solution approaches which give importance to reducing complexity in parallel to reach optimality by using some practical rules or principles. For the instances which have larger sizes, heuristics are beneficial to find approximate solutions in a timely manner. Unlike exact algorithms that provide optimality, heuristics try to find quick & near optimal solutions instead of an exact optimal solution. You can find the well-known heuristic methods that can be deployed to solve the MTSP.

2.7.1 Local Search Algorithms

Local Search is a heuristic algorithm that tries to solve combinatorial optimization problems by maximizing the objective function. It requires an initial solution, after the initial solution is constructed the approach reduces cost by changes in an iterative way. The most-known examples of the local search are 2-opt, 3-opt and or-opt.

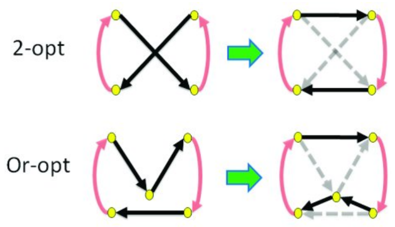


Figure 2.4: 2-opt and or-opt

2.7.2 Genetic Algorithms

Genetic Algorithms are the heuristics inspired by the natural selection and genetic processes. The aim of the genetic algorithms is to find approximate solutions in a timely manner. After the construction of initial solution, it tries to reduce cost by different steps like crossing-over or mutation.

2.7.3 Swarm Intelligence Algorithms

Swarm Intelligence algorithms are the optimization approach that inspired by the collective mechanisms in which the individuals are interacting to generate a collective strategy. These algorithms simulate the principles of collaborative organizations with decentralized decision-making systems. The most common Swarm Intelligence Algorithms are as follows; Ant colony Optimization, Bee Algorithm, Firefly Algorithm, Bat Algorithm.

2.7.4 Simulated Annealing

Simulated Annealing is a probabilistic approach inspired from the metallurgy. In the annealing process of metallurgy, materials are heated and cooled to finalize the structure. The analogy can be explained in the approach as; generated solutions converge to local optimum points but avoid to stay local optimums to explore the global optimum points.

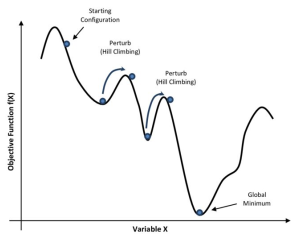


Figure 2.5: Simulated Annealing

2.7.5 Greedy Algorithms

Greedy algorithms promise the simple and intuitive methods to solve the combinatorial optimization problems. The idea behind the algorithm is to make decisions to reach local optima in each stage and try to converge globally optimal solution.

The ultimate difference between the greedy algorithms and the other heuristics is the no need for initial solution in greedy algorithms. It means that greedy algorithm can build a solution from nothing.

In the thesis, a novel greedy algorithm builds the initial solution and then the solution is improved by a genetic algorithm. Greedy algorithm is selected to obtain consistent result and avoid randomness.

CHAPTER 3

RELATED WORK

3.1 Traveling Salesman Problem

Traveling Salesman Problem has a history that covers several centuries.

The first written source about Traveling Salesman Problem was written in 1832 by an old commis-voyageur [7]. The book mentions the problem and gives examples about tours through Germany and Switzerland. But it doesn't propose any mathematical solution. The first problems related to Traveling Salesman Problem were discussed by Thomas Kirkman and also the Irish mathematician Rowan Hamilton who invented the Icosian game which targets to find Hamiltonian Cycle. Discussions about the work of Hamilton and Kirkman can be found in [8]. In 1930, Traveling Salesman Problem is mathematically formulated by Karl Menger [9]. His formulation aimed to find the shortest route within the cities which have to be visited once only.

Dantzig et al. [10] have tried to solve 49-state problem that aims to an optimal tour that covers all states. Their approach contains adding some inequalities to the problem and pruning the possible solution set. Their perspective was relatively close to the Branch and Bound Algorithm. In 1976, Nicos Christofides [11] proposed an approximation algorithm that guarantees that its output is at most 3/2 times worse than the optimal solution. His method had continued to provide the best "worst case scenario" until 2011. In this year; Gharan et al. [12] decreased the worst case ratio to 3/2-ϵ_0 ($\epsilon_0 > 0$) by a randomized rounding approach.

There were many methods and approaches that deployed to solve Traveling Salesman Problem. Integer Programming techniques to solve TSP were formulated by Miller et al. [13]. Diaby[14] proposed both linear and non-linear techniques to solve TSP. Imdat et al.[15] and Hungerlander [16] tried to solve TSP by an integer linear programming approach with time window constraints. Egon et al[17] presented Branch and Bound Techniques to solve Traveling Salesman Problem with different kinds of relaxations. Some vehicle-specific TSP's are also tried to be solved by branch and bound. Poikonen's study [18] is a good example on this.

Genetic Algorithms are also deployed to solve TSP's. Friesleben and Merz [19] tried to find near-optimum solutions to the TSP. Deng et al. [20] executed the genetic algorithm in an hybrid cellular way. Bat Algorithm[21] is executed by Osaba et al to solve symmetric and asymmetric TSPs. An adaptive Simulated Annealing[22] approach is deployed with a greedy search.

Studies on TSP have been reviewed so far. From that point, articles and studies related to MTSP will be presented.

3.2 Multi Traveling Salesman Problem

3.2.1 Brief History and Problem Variations

The Multi Traveling Salesman Problem (MTSP) field does not have as much research activity as the field of the Traveling Salesman Problem (TSP). Due to this fact, some closer problems to MTSP like VRP or MAPF-MD will also be mentioned for providing a broader perspective.

One of the first MTSP related effort is done by Dantzig [23] who tried to find a solution to the 49-state problem mentioned in TSP. They proposed a procedure based on a linear programming formulation to optimize the dispatching process executed by multiple trucks. Clarke and Wright[24] criticized the Dantzig's approach and modified it to solve the problem with their known Clarke & Wright Savings Algorithm.

Some different areas are also covered in MTSP area like Greenstein [25]'s early work at 1970. The aim of the study is optimizing the cost of multi-edition press printing process. The terms in the article is analogous to the MTSP problem like plate change costs - costs between cities. He formulated a mixed-integer program and stated that optimal solution can not be found due to the problem size. Then he proposed a heuristic to find a near-optimal solution. The press scheduling problem is mentioned as an MTSP problem also by Carter et al.[26]. They applied a genetic algorithm to solve the problem.

After Greenstein's effort, Angel et al. [27] tried to solve a bus scheduling problem as an MTSP problem with additional constraints. They aimed to minimize the number of routes and the total distance covered by buses with capacity and deadline constraints. There are also some studies about the school bus scheduling problems with different variants like single school - multiple schools or single loading - mixed loading. Park and Kim[28] provided a review of articles about school bus scheduling problem.

As mentioned in the introduction, disaster management processes can also be considered as an MTSP problem. Cheikhrouhou et al.[29] presented an analytical hierarchy based MTSP solution for the disaster management systems. Bodaghi et al.[30] proposed a mixed integer programming model to optimize the resource allocation in a stochastic environment. They also presented a case study about a Victorian Bushfire scenario. Bodaghi[31] has also provided a review of studies about multi resource emergency recovery operations.

3.2.2 Minimizing Different Objectives

Although some studies[32] made an effort to optimize alternative metrics, most of the studies[33],[34] aimed to minimize total distance/cost for Traveling Salesman Problems. When the number of salesman changes 1 to multiple, additional objective options will be available. So, it can be said that there are various objective options for Multi Traveling Salesman Problem.

The first option is minimizing the maximum distance covered by a salesman. In many real-life applications like collecting products within a depot by forklifts to a single truck, minimizing the maximum cost or time is important. The first min-max MTSP is proposed by França et al.[35]. They deployed the Tabu Search and K Nearest Neighbour methods and compared their performances. Variety of methods can be applied to solve the minmax MTSP problem like reinforcement learning[36] or neural networks[37].

Some studies tried to enhance the balance between salesmen. Minmax problem is also trying to minimize the maximum distance covered by a salesman, but maximizing the minimum distance is also aimed in this objective. In real life, ensuring the balance among workers is important to avoid possible problems among workers. Vandermeulen et al.[38] provided a solution to balanced task allocation.

Some studies tried to optimize the total distance covered by all salesmen for multi traveling salesman problems. Minimizing the total cost makes sense in real-life cases without deadlines or waiting customers. Tabu search [39] and integer programming [40] can be applied to solve the minsum MTSP.

Within the most studies on MTSP, multi objective minimization is the key target. That's the most applicable approach in real life because solutions both provide quickness and cost minimization. Zhen et al. [41] proposed a two stage algorithm to optimize two objectives. They initialized the solution with a clustering method and tried to improve the solution with a neighborhood search. Ant Colony[42] method is also used by Chen et al. to minimize the two objectives. Our thesis's objective is also minimizing both objectives.

3.2.3 Probabilistic vs. Deterministic

Two types of MTSP instances can be proposed, the first one is the probabilistic. In this type of problem; costs, number of salesmen or cities can be changed through time. Although the probabilistic environment receives less attention, it can be applicable in real life. Carreno et al.[43] presented the probabilistic Multi Robot Systems problem and tried to solve it with a model based reinforcement learning method. Dynamic routing can also be subject of MTSP problem like Garn's[44] study.

The second type is deterministic MTSP instance. Most MTSP studies are based on a deterministic environment like [45] and [46]. The thesis's effort is also done on a deterministic problem environment.

3.2.4 Depots & Types of Paths

The standard version of MTSP is defined with a single depot and closed paths. Closed path means that the salesman must finish the tour at the starting city. In terms of these two breakdowns, there can be four different problem variants. The first one is the single-depot and closed path variant, Thenepalle and Singamsetty[47] tried to solve an MTSP problem with a single depot and open paths but one of the salesmen should finish the tour in the depot. In Wang's[48] variant, all salesmen must return to the fixed-single depot. Ghafurian[49] presented an Ant Colony problem to solve the Multi Depot MTSP with closed paths. In Assaf & Ndiaye's study [5], there are multi depots and salesmen shouldn't return to their depots.

The thesis provides solutions for both single depot & multi depot open path MTSP's. Finding the collision free paths can also be an another variant for the problem like Semiz et al's [50] study. We didn't specifically avoid from collisions but our approach will provide solutions without any intersections whether in a path of salesman or among paths of salesmen.

CHAPTER 4

PROPOSED WORK

In this chapter, we will provide the mathematical description of the Multi Traveling Salesman Problem with objective and constraint formulations. After that, proposed method will be presented with the reasons and explanations in a verbal and visual way.

4.1 Problem Description

Let N be the number of cities and K be the number of salesmen. c_{ij} is the cost between the cities i and j. S_{ijk} is the Boolean variable that shows whether the path i to j discovered in the route of the Salesman k. f_{ik} represents whether the salesman k starts from city i.

4.1.1 Constraints

$$\underset{i \neq j\ \forall i\ \forall j}{c_{ij} > 0} \tag{4.1}$$

Costs between cities must be positive as been ruled in classical Traveling Salesman Problem.

$$\underset{i \neq j\ \forall i\ \forall j\ \forall k}{S_{ijk} \in \{0, 1\}} \tag{4.2}$$

Any salesman can traverse between any pair of cities. In real life problems there can be some additional constraints like this vehicle can not be used between given cities due to weather conditions etc.

$$\underset{\forall i\ \forall k}{f_{ik}} \in \{0,1\} \tag{4.3}$$

A salesman can start from any city.

$$\sum_{\forall i}\sum_{k=1}^{K} f_{ik} \leq K \tag{4.4}$$

A city can be a starting point of any salesman. And it can also be the starting points of multiple salesmen. If the problem is a single-depot MTSP, $\sum_{k=1}^{K} f_{ik}$ will be $\in \{0,1\}$. If all salesmen have to start from different cities $\sum_{k=1}^{K} f_{ik}$ will be $\in \{0,K\}$.

$$\underset{\forall k}{\sum_{i=1}^{N} f_{ik} = 1} \tag{4.5}$$

Each salesman have to start from one city. A Salesman can not start from multiple cities.

$$\underset{i\neq j\ \forall i}{\sum_{j=1}^{N}\sum_{k=1}^{K} \cancel{S_{ijk} = 1} \quad S_{ijk} \leq 1} \tag{4.6}$$

In the close path variants of MTSP, strikethroughed constraint have to be applied. But in the open path variant, final destinations of the salesmen can not be a starting point of any pairs.

$$\underset{i\neq j\ \forall k}{\sum_{i=1}^{N}\sum_{j=1}^{N} S_{ijk} \geq 1} \tag{4.7}$$

All salesmen have to traverse between any pair of cities at least once. The constraint keeps all salesmen active.

$$\sum_{i=1}^{N}\sum_{k=1}^{K} S_{ijk} + \boldsymbol{f_{jk}} = 1 \quad \underset{i \neq j \ \forall j}{} \tag{4.8}$$

If a city is not a starting point of any salesman, it has to be visited by a salesman in exactly one time.

$$\sum_{i=1}^{N}\sum_{k=1}^{K}(f_{ik} + \sum_{j=1}^{N} S_{ijk}) = N \quad \underset{i \neq j}{} \tag{4.9}$$

Total number of traverses in an MTSP instance should be equal to the difference between the number of cities and the number of unique starting cities.

Other than the mathematical formulations, a path of a salesman should not intersect itself in an Euclidean Traveling Salesman Problem. Because if an intersection occurs in a path, it can not be optimal and it should be transformed to a path without intersection.

As mentioned in the Urban Operations Research Book [51] the optimum traveling salesman tour does not intersect itself. Cost of two intersecting paths is always more than the non-intersecting alternatives due to the triangle inequality. As shown in Figure 4.1 $|ao| + |co| + |ob| + |od| > |ab| + |cd|$ because $|ao| + |ob| > |ab|$ and $|co| + |od| > |cd|$

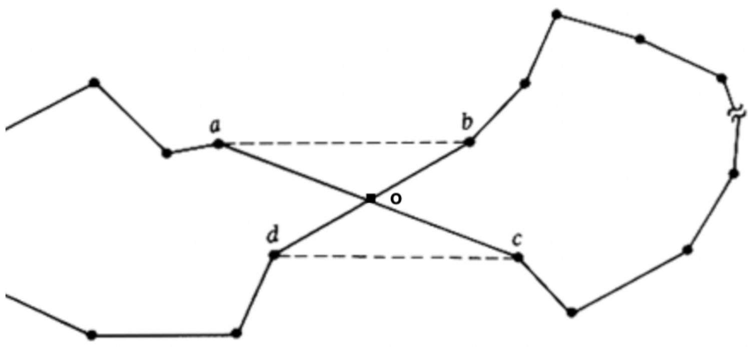


Figure 4.1: A self intersecting path

We can also state the same principle for the multi traveling salesman problem, if there exists any intersection between two salesmen, they can reduce the cost by swapping their cities.

4.1.2 Objectives

As stated in the previous chapters, there can be multiple objectives for a multi traveling salesman problem. The first one is the minimizing the total distance covered by salesmen. It can be represented as;

$$\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{\substack{k=1\\ i\neq j}}^{K} S_{ijk} * c_{ij} \tag{4.10}$$

Minimizing the maximum distance covered by a salesman can also be aimed. The objective function can be formulated as;

$$\max\sum_{i=1}^{N}\sum_{\substack{j=1\\ i\neq j\ \forall k}}^{N} S_{ijk} * c_{ij} \tag{4.11}$$

4.2 Relative Distance Approach

Relative Distance Approach redefines the distance description for the multi traveling salesman problem.

Classical description of distance is dependent to only the two objects that the distance been measured between them. In MTSP, this type of distance definition is insufficient because there are multiple salesmen and cities in the environment.

Let's define m_{ik} as the optimal marginal insertion cost of city i to the path of salesman k, M_i as the $\sum_{k=1}^{K} m_{ik}$ and p_k is the current distance of salesman k'th path. d is the minimum distance between all pairs of cities. The Relative Distance Coefficient (rd_{ik}) for each city can be formulated as follows;

$$rd_{ik} = M_i / (\sqrt{\underset{\forall i}{m_{ik}}} * \sqrt{d + p_k}) \tag{4.12}$$

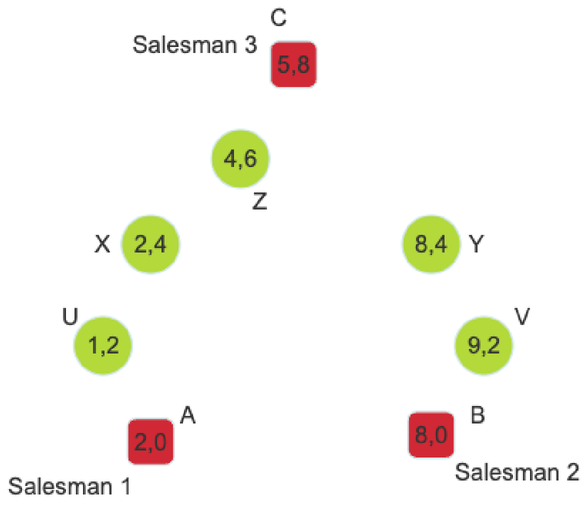


Figure 4.2: An example MTSP instance

In Figure 4.2; salesman 1 starts from city A, salesman 2 starts from city B, salesman 3 starts from C. The minimum distance between cities is $\sqrt{5} = 2,24$.

Table 4.1: Regular Distance Matrix

Regular Distances Between Salesmen and Cities					
Salesman	U	V	X	Y	Z
1	2,24	7,28	4	7,21	6,32
2	7,28	2,24	7,21	4	7,21
3	7,21	7,21	5	5	2,24
Total	16,73	16,73	16,21	16,21	15,77

Relative distance coefficients can be calculated by the 4 parameters;

- M_i (16,73 for City U)
- m_{ik} (7,21 for Salesmen 2 & City X)
- $d + p_k$ (2,24+0 for each salesman because there is no assignment yet)

Table 4.2: Relative Distance Coefficients

Relative Distances Between Salesmen and Cities	
Salesman & City	Relative Distance
rd_{1U}	$16,73/(\sqrt{2,24} * \sqrt{2,24+0}) = 7,47$
rd_{1V}	$16,73/(\sqrt{7,28} * \sqrt{2,24+0}) = 4,14$
rd_{1X}	$16,21/(\sqrt{4} * \sqrt{2,24+0}) = 5,42$
rd_{1Y}	$16,21/(\sqrt{7,21} * \sqrt{2,24+0}) = 4,03$
rd_{1Z}	$15,77/(\sqrt{6,32} * \sqrt{2,24+0}) = 4,19$
rd_{2U}	$16,73/(\sqrt{7,28} * \sqrt{2,24+0}) = 4,14$
rd_{2V}	$16,73/(\sqrt{2,24} * \sqrt{2,24+0}) = 7,47$
rd_{2X}	$16,21/(\sqrt{7,21} * \sqrt{2,24+0}) = 4,03$
rd_{2Y}	$16,21/(\sqrt{4} * \sqrt{2,24+0}) = 5,42$
rd_{2Z}	$15,77/(\sqrt{7,21} * \sqrt{2,24+0}) = 3,92$
rd_{3U}	$16,73/(\sqrt{7,21} * \sqrt{2,24+0}) = 4,16$
rd_{3V}	$16,73/(\sqrt{7,21} * \sqrt{2,24+0}) = 4,16$
rd_{3X}	$16,21/(\sqrt{5} * \sqrt{2,24+0}) = 4,84$
rd_{3Y}	$16,21/(\sqrt{5} * \sqrt{2,24+0}) = 4,84$
rd_{3Z}	$15,77/(\sqrt{2,24} * \sqrt{2,24+0}) = 7,04$

The Relative Distance Coefficients between Salesman 1 & City U and Salesman 2 & City V have the maximum values among other coefficients. We can assign U to salesman 1 as shown in Figure 4.3.

When looked at the distances carefully, closeness of Z to salesman 3 is also equal to U-1 or V-2 but relative distance coefficient of Z-3 is smaller than these two. Because Z is more closer to 1 and 2 than U-2, U-3 and V-1, V-3.

After the assignment of city U to Salesman 1, new relative distance coefficients are shown in the Table 4.3;

The cities V, X, Y and Z should be inserted after city U to optimize the total distance covered by salesman 1. It is not a general case. If city X was inserted first, city U would be inserted between city A and city X.

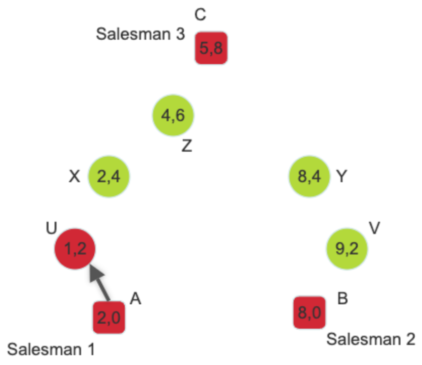


Figure 4.3: Example MTSP instance after first task assignment

Table 4.3: Relative Distance Coefficients after U is Assigned to Salesman 1

Relative Distances Between Salesmen and Cities	
Salesman & City	Relative Distance
$\cancel{rd_{1U}}$	City U is assigned to Salesman 1
rd_{1V}	$16,73/(\sqrt{8} * \sqrt{2,24+2,24}) = 2,79$
rd_{1X}	$16,21/(\sqrt{2,24} * \sqrt{2,24+2,4}) = 5,12$
rd_{1Y}	$16,21/(\sqrt{7,62} * \sqrt{2,24+2,24}) = 2,77$
rd_{1Z}	$15,77/(\sqrt{5} * \sqrt{2,24+2,24}) = 3,33$
$\cancel{rd_{2U}}$	City U is assigned to Salesman 1
rd_{2V}	$16,73/(\sqrt{2,24} * \sqrt{2,24+0}) = 7,47$
rd_{2X}	$16,21/(\sqrt{7,21} * \sqrt{2,24+0}) = 4,03$
rd_{2Y}	$16,21/(\sqrt{4} * \sqrt{2,24+0}) = 5,42$
rd_{2Z}	$15,77/(\sqrt{7,21} * \sqrt{2,24+0}) = 3,92$
$\cancel{rd_{3U}}$	City U is assigned to Salesman 1
rd_{3V}	$16,73/(\sqrt{7,21} * \sqrt{2,24+0}) = 4,16$
rd_{3X}	$16,21/(\sqrt{5} * \sqrt{2,24+0}) = 4,84$
rd_{3Y}	$16,21/(\sqrt{5} * \sqrt{2,24+0}) = 4,84$
rd_{3Z}	$15,77/(\sqrt{2,24} * \sqrt{2,24+0}) = 7,04$

With the relative distances approach, all cities can be assigned to salesmen iteratively as shown in Figure 4.4.

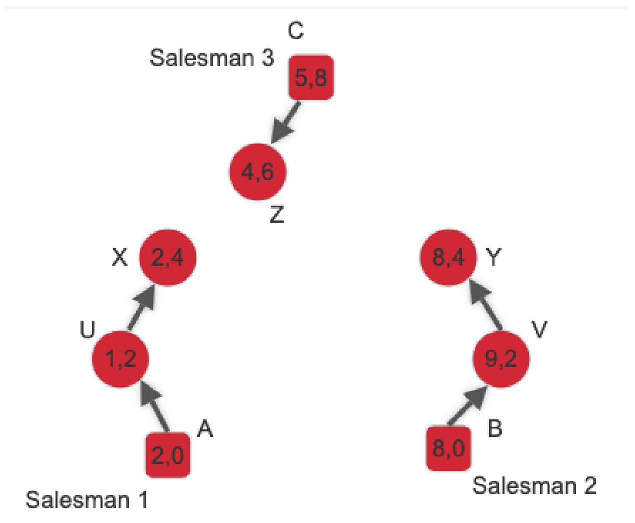


Figure 4.4: Example MTSP instance after all assignments

Actually, the output of the Relative Distances Approach is not different from the classical approach (assign cities to salesmen first with clustering then find optimum paths among assigned cities). But, what if the city Q is added to the map?

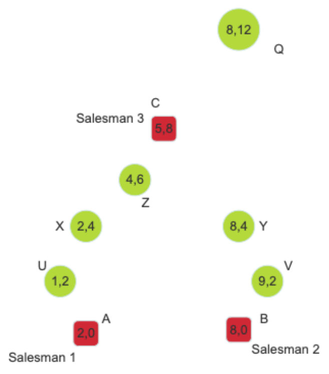


Figure 4.5: Example MTSP instance with the city Q

The top relative distance coefficient is measured between Salesman 3 and City Q as shown in Table 4.5. The pair is not the closest salesman-city pair on the map. Moreover, City Q is not the closest city to Salesman 3 (Z is closer). But Q is significantly far away from the other salesmen. Because of this, Q should be assigned to Salesman 3 **first**.

Table 4.4: Regular Distance Matrix with City Q

Regular Distances Between Salesmen and Cities						
Salesman	U	V	X	Y	Z	Q
1	2,24	7,28	4	7,21	6,32	13,42
2	7,28	2,24	7,21	4	7,21	12
3	7,21	7,21	5	5	2,24	5
Total	16,73	16,73	16,21	16,21	15,77	30,41

Table 4.5: Relative Distance Coefficients with City Q

Relative Distances Between Salesmen and Cities	
Salesman & City	Relative Distance
rd_{1U}	$16,73/(\sqrt{2,24} * \sqrt{2,24+0}) = 7,47$
rd_{1V}	$16,73/(\sqrt{7,28} * \sqrt{2,24+0}) = 4,14$
rd_{1X}	$16,21/(\sqrt{4} * \sqrt{2,24+0}) = 5,42$
rd_{1Y}	$16,21/(\sqrt{7,21} * \sqrt{2,24+0}) = 4,03$
rd_{1Z}	$15,77/(\sqrt{6,32} * \sqrt{2,24+0}) = 4,19$
rd_{1Q}	$30,42/(\sqrt{13,42} * \sqrt{2,24+0}) = 5,55$
rd_{2U}	$16,73/(\sqrt{7,28} * \sqrt{2,24+0}) = 4,14$
rd_{2V}	$16,73/(\sqrt{2,24} * \sqrt{2,24+0}) = 7,47$
rd_{2X}	$16,21/(\sqrt{7,21} * \sqrt{2,24+0}) = 4,03$
rd_{2Y}	$16,21/(\sqrt{4} * \sqrt{2,24+0}) = 5,42$
rd_{2Z}	$15,77/(\sqrt{7,21} * \sqrt{2,24+0}) = 3,92$
rd_{2Q}	$30,42/(\sqrt{12} * \sqrt{2,24+0}) = 5,87$
rd_{3U}	$16,73/(\sqrt{7,21} * \sqrt{2,24+0}) = 4,16$
rd_{3V}	$16,73/(\sqrt{7,21} * \sqrt{2,24+0}) = 4,16$
rd_{3X}	$16,21/(\sqrt{5} * \sqrt{2,24+0}) = 4,84$
rd_{3Y}	$16,21/(\sqrt{5} * \sqrt{2,24+0}) = 4,84$
rd_{3Z}	$15,77/(\sqrt{2,24} * \sqrt{2,24+0}) = 7,04$
rd_{3Q}	$30,42/(\sqrt{5} * \sqrt{2,24+0}) = 9,09$

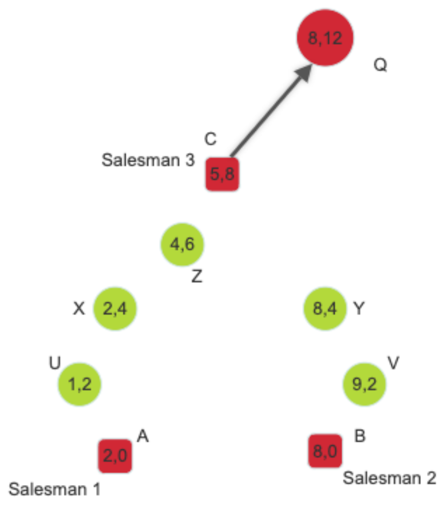


Figure 4.6: Q is Assigned to Salesman 3

Table 4.6: Relative Distance Coefficients after Q is Assigned to Salesman 3

Relative Distances Between Salesmen and Cities	
Salesman & City	Relative Distance
rd_{1U}	$16,73/(\sqrt{2,24}*\sqrt{2,24+0})=7,47$
rd_{1V}	$16,73/(\sqrt{7,28}*\sqrt{2,24+0})=4,14$
rd_{1X}	$16,21/(\sqrt{4}*\sqrt{2,24+0})=5,42$
rd_{1Y}	$16,21/(\sqrt{7,21}*\sqrt{2,24+0})=4,03$
rd_{1Z}	$15,77/(\sqrt{6,32}*\sqrt{2,24+0})=4,19$
rd_{2U}	$16,73/(\sqrt{7,28}*\sqrt{2,24+0})=4,14$
rd_{2V}	$16,73/(\sqrt{2,24}*\sqrt{2,24+0})=7,47$
rd_{2X}	$16,21/(\sqrt{7,21}*\sqrt{2,24+0})=4,03$
rd_{2Y}	$16,21/(\sqrt{4}*\sqrt{2,24+0})=5,42$
rd_{2Z}	$15,77/(\sqrt{7,21}*\sqrt{2,24+0})=3,92$
rd_{3U}	$16,73/(\sqrt{7,21}*\sqrt{2,24+2,24})=1,50$
rd_{3V}	$16,73/(\sqrt{7,21}*\sqrt{2,24+2,24})=1,60$
rd_{3X}	$16,21/(\sqrt{5}*\sqrt{2,24+2,24})=1,61$
rd_{3Y}	$16,21/(\sqrt{5}*\sqrt{2,24+2,24})=1,67$
rd_{3Z}	$15,77/(\sqrt{2,24}*\sqrt{2,24+2,24})=1,91$

After Q is assigned to Salesman 3, relative distance coefficients are recalculated and it can be seen that $rd_{1Z} > rd_{3Z}$ and $rd_{2Z} > rd_{3Z}$. That's an expected result because if Z is assigned to Salesman 3 after Q, it should traverse in a zigzag way.

According to new coefficients, U will be assigned to Salesman 1, V will be assigned to Salesman 2. At the end of the process, Z is assigned to Salesman 1.

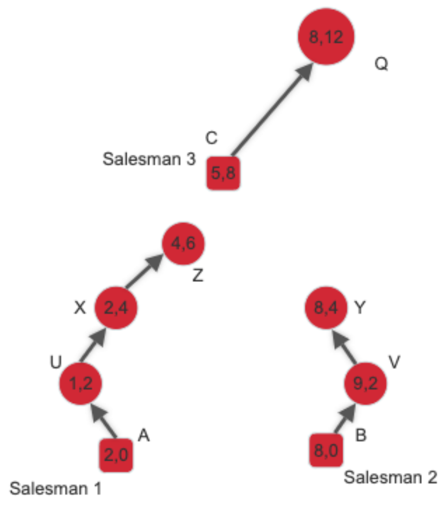


Figure 4.7: Final State of the Map

The total distance covered by salesmen is 16,79. The detailed numbers are shown in Table 4.7:

Table 4.7: Total Distance by Relative Distances Approach

Distances Within The Paths of Salesman		
Salesman	Cities	Relative Distance
1	A→U	2,24
1	U→X	2,24
1	X→Z	2,83
2	B→V	2,24
2	V→Y	2,24
3	C→Q	5
total	A→U→X→Z B→V→Y C→Q	16,79

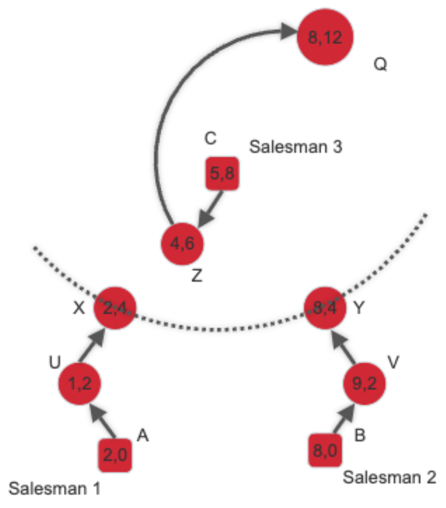


Figure 4.8: Task Assignment with Classical Approach

If the cities are assigned to the salesmen with the regular distance criteria, Z will be assigned to Salesman 1.

Table 4.8: Total Distance by Classical Approach

Distances Within The Paths of Salesman		
Salesman	Cities	Relative Distance
1	A→U	2,24
1	U→X	2,24
2	B→V	2,24
2	V→Y	2,24
3	C→Z	2,24
3	Z→Q	7,21
total	A→U→X B→V→Y C→Q→Z	18,41

The total distance of the solution generated by the regular distance criteria is more than the relative distance approach. Our approach performed 8.8% better than the classical approach in terms of the total distance covered by salesmen. If the objective is minimizing the maximum distance covered by a salesman, our approach performed 22,7% (9.45 vs. 7.31) better than the classical approach.

4.3 Improvement Heuristics

After the greedy task assignment, two improvement heuristics are applied to decrease the total cost. The first one is swapping the part of assigned cities among salesmen. Second one is reversing the orders of cities assigned to a salesman.

An example output of the relative distance approach and its total cost can be seen in Figure 4.14.

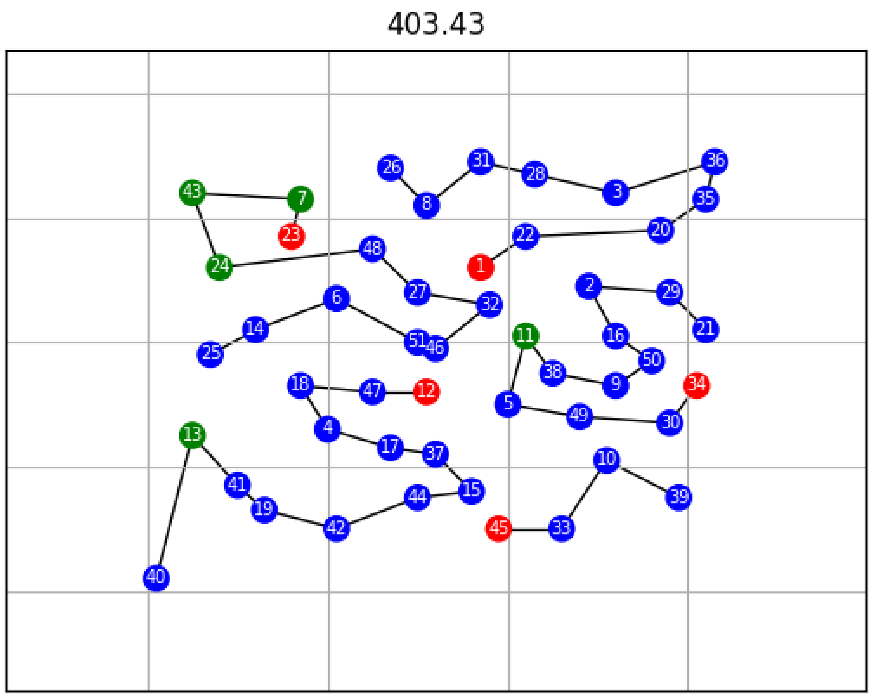


Figure 4.9: An example MTSP Output

The green-colored cities will be the subject of the task swapping procedure among salesmen. The first swapping operation is done for the city 13 between the salesmen starts from city 23 and city 12.

The salesman starts from 23 draws a G shaped path due to having the upper left cities (7,24,43) of the map in its schedule. These nodes are transferred to the salesman starts from 1 and the total distance is reduced.

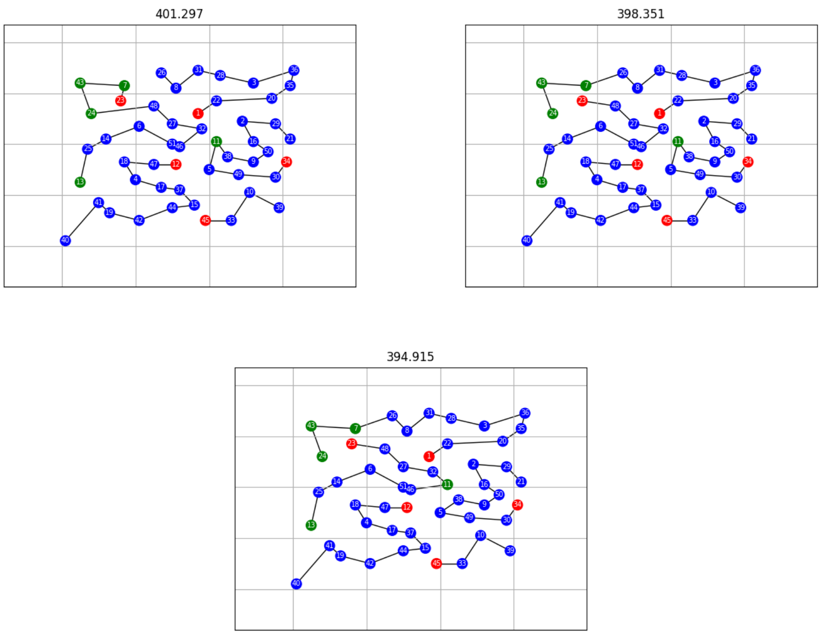


Figure 4.10: MTSP Output after Task Swapping

City 11 is transferred from the salesman starts from 23 to the salesman starts from 34.

After these swapping operations, the total distance covered by salesman is reduced from 403 to 394.

In Figure 4.11, a sample task reversing procedure can be seen. The salesman starts from City 5, reversed the 50-16 chain to 16-50 chain.

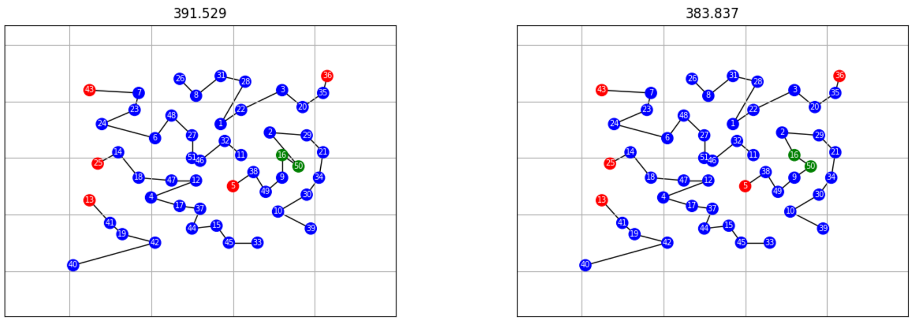


Figure 4.11: MTSP Output after the Task Reversing

Another task reversing procedure is visualized in Figure 4.12. The salesman starts from city 9 reversed the 29-21-34-30 chain to 30-34-21-29 chain.

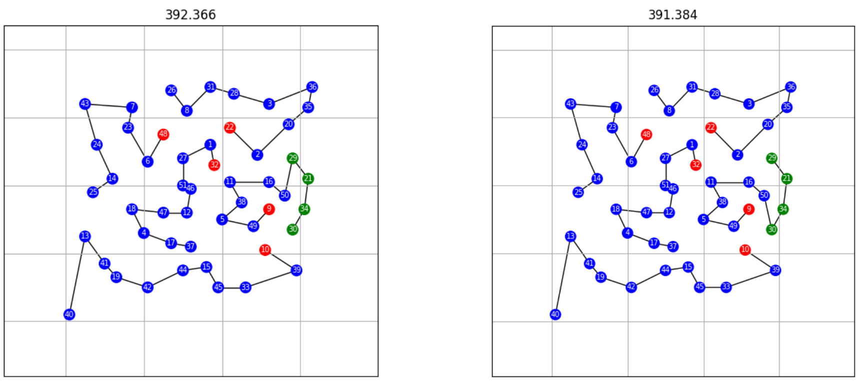


Figure 4.12: MTSP Output after the Task Reversing

In summary, the proposed solution method consists of two phases;

- Constructing an Initial Solution via Relative Distance Approach
- Improvement with Two Operations
  - Task Swapping between Salesmen
  - Reversing the Tasks of Salesman

4.4 Algorithm & Pseudo Code

The principles and the mechanism of the approach are given in the previous section. Here, you can find the pseudo code and the related explanations. After the pseudo code, complexity of each part will be shown with respect to the number of cities and the number of salesmen

At the initialization part of the problem, $(N - M) \times M \rightarrow \theta(N \times M - M^2)$ iterations were executed to find the initial relative distance coefficients (lines 2-12).

After that, the city-salesman pair with the highest relative distance coefficient is determined and the city is assigned to the salesman with the optimal position (lines 13-14).

Algorithm 1 Task Assignment & Ordering With Relative Distance Approach

```
1: $NumberofAssignedCities = NumberOfDistinctStartingCities$
2: for $i = 1$ to $N$ do
3:     for $k = 1$ to $M$ do
4:         $m_{ik} \leftarrow c_{ik}$
5:     end for
6:     $M_i \leftarrow M_i + m_{ik}$
7: end for
8: for $i = 1$ to $N$ do
9:     for $k = 1$ to $M$ do
10:        $rd_{ik} \leftarrow M_i/(\sqrt{m_{ik}} * \sqrt{d})$
11:    end for
12: end for
13: $(AssignedCity, AssigneeSalesman) \leftarrow \arg\max_{ik}(rd_{ik})$
14: $AssignTheCity(AssignedCity, AssigneeSalesman)$
15: $NumberofAssignedCities = NumberofAssignedCities + 1$
16: while $NumberofAssignedCities \leq N$ do
17:     for $i = 1$ to $N$ do
18:         $rd_{iAssigneeSalesman} \leftarrow M_i/(\sqrt{m_{iAssigneeSalesman}} * \sqrt{d + p_{AssigneeSalesman}})$
19:     end for
20:     $(AssignedCity, AssigneeSalesman) \leftarrow \arg\max_{ik}(rd_{ik})$
21:     $AssignTheCity(AssignedCity, AssigneeSalesman)$
22:     $NumberofAssignedCities = NumberofAssignedCities + 1$
23: end while
```

Within each iteration of while loop, the only relative distance coefficient updates are done for the salesman with the most recent task assignment. Because other relative distance coefficients didn't changed by the assignment. The fact can be observed from the differences between Table 4.2 & Table 4.3 and Table 4.5 & Table 4.6.

Before the while loop, M+1 cities are already assigned. In each iteration of while loop, one city is assigned to a salesman. It means that there will be N-M-1 iterations within the while loop. In x'th iteration, relative distance coefficients of N-M-x cities with the most recent salesman are re-calculated.

The complexity of each relative distance recalculation is $\theta(\lceil \frac{N}{M} \rceil)$ in x'th iteration of a while. The average-time complexity of the problem becomes $(N - M - 1) \times (N - M - x) \times \frac{N}{M} \rightarrow \theta(\frac{N^3}{M})$.

Before completing the Relative Distance Approach, the importance of the third parameter $\sqrt{d + p_{AssigneeSalesman}}$ should be explained. With this parameter, relative distance coefficient becomes inversely proportional to path length of a salesman. The inverse proportion makes difficult the assignment of new city to the salesmen with higher distance costs. Without this parameter unbalanced task assignments, moreover task-less salesmen would be observed.

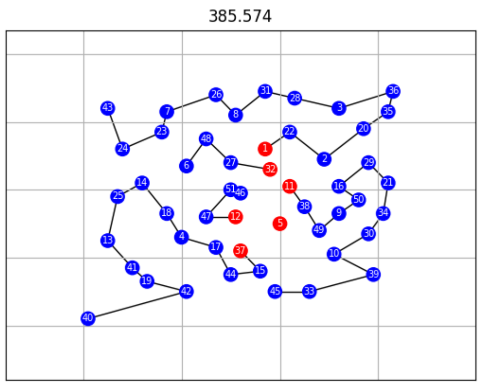


Figure 4.13: An Output Without Third Parameter

An example task allocation sequence by relative distance approach can be seen from Figure 4.13.

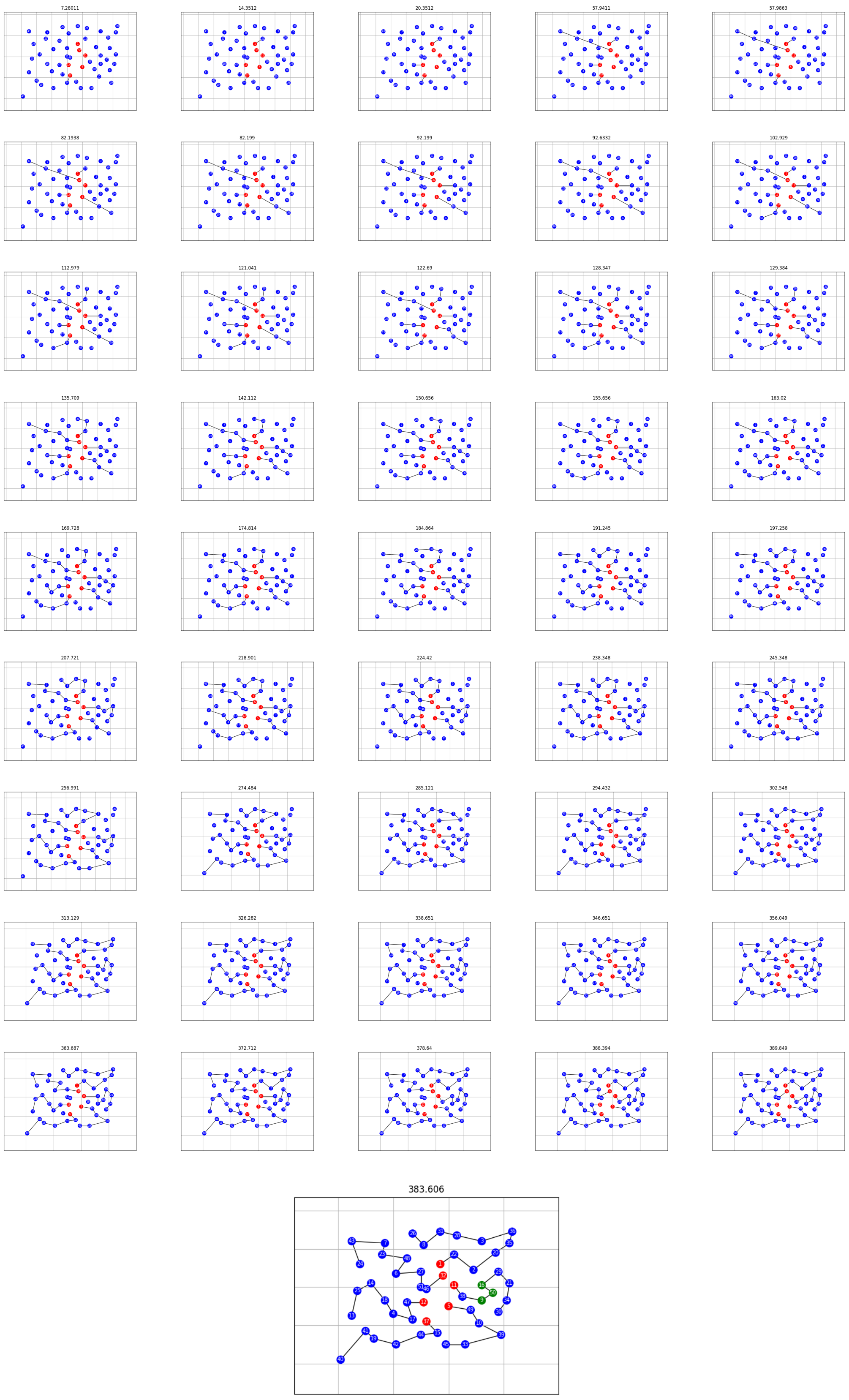


Figure 4.14: MTSP Output after the Task Reversing

4.5 Brute Force Method

In the MTSP literature, there are various researches with different categories like open path & close path or single depot & multiple depots. However, the absence of a standardized input instances makes it difficult to compare results across studies. Due to this situation, two different brute force algorithms are deployed to evaluate our algorithm's performance.

The first brute force method looks for the best solution among almost all possible scenarios.

This method can be used to evaluate our algorithm's performance for $N \leq 16$ and $M \leq 4$

Algorithm 2 Complete Brute Force Approach

```
1: $Number \leftarrow \frac{(M+1)^{N-M}-1}{M}$
2: $OptimalCost = INF$
3: for $i = Number$ to $Number \times M$ do
4:     $RadixedNumber = Radix(Number, M+1)$
5:     $TaskAssignmentArray = NumberToArray(RadixedNumber)$
6:     if $IsProper(TaskAssignmentArray)$ then
7:         while $NewPermutation(TaskAssignmentArray) exists$ do
8:             $TasksWithOrders = NewPermutation(TaskAssignmentArray)$
9:             $CostOfScenario = CalculateCost(TasksWithOrders)$
10:            if $CostOfScenario < OptimalCost$ then
11:                $OptimalCost = CostOfScenario$
12:                $OptimalSolution = TasksWithOrders$
13:            end if
14:        end while
15:    end if
16: end for
```

In this algorithm $\frac{(M+1)^{N-M}-1}{M}$ is treated as a magic number to explore different task assignments iteratively.

Assume that there are 14 cities and 3 salesmen that starts from different cities. $(11111111111)_4 = 4^0 + 4^1 + 4^2 + ... + 4^{10} = \frac{(4)^{11}-1}{3} \rightarrow \frac{(M+1)^{N-M}-1}{M}$. $(11111111111)_4$ represents the assignment array on base four. It means that all cities are assigned to salesman 1. There are 11 digits because of the number of unassigned cities. The next assignment arrays will become $(11111111112)_4$, $(11111111113)_4$ then $(11111111121)_4$ to $(33333333333)_4$.

Before proceeding to permutation part, the task assignment is examined according to the task allocation balance. If more than $\frac{2(N-M)}{M}$ cities assigned to a salesman, it doesn't considered proper and procedure continues to acquire a proper task assignment array. After determining task assignments, permutations within the task assignments are tried to find best solution.

Complexity of this procedure is $O(N!^M)$.

The second brute force algorithm is used to evaluate relative distance approach's outputs in terms of task orderings only. It doesn't seek for new assignment options due to the exponentially increasing execution times. It can be applied to instances with $\frac{N}{M} \leq 10$.

Let l_k be the number of cities assigned to salesman k. Complexity of the partial brute force method is $(max(l_k) - 1)!$

Algorithm 3 Partial Brute Force Approach

```
1: $NumberofAssignedCities = NumberOfDistinctStartingCities$
2: for $i = 1$ to $N$ do
3:     for $k = 1$ to $M$ do
4:         $m_{ik} \leftarrow c_{ik}$
5:     end for
6:     $M_i \leftarrow M_i + m_{ik}$
7: end for
8: for $i = 1$ to $N$ do
9:     for $k = 1$ to $M$ do
10:        $rd_{ik} \leftarrow M_i/(\sqrt{m_{ik}} * \sqrt{d})$
11:    end for
12: end for
13: $(AssignedCity, AssigneeSalesman) \leftarrow \arg\max_{ik}(rd_{ik})$
14: $AssignTheCity(AssignedCity, AssigneeSalesman)$
15: $NumberofAssignedCities \leftarrow NumberofAssignedCities + 1$
16: while $NumberofAssignedCities \leq N$ do
17:    for $i = 1$ to $N$ do
18:        $rd_{iAssigneeSalesman} \leftarrow M_i/(\sqrt{m_{iAssigneeSalesman}} * \sqrt{d + p_{AssigneeSalesman}})$
19:    end for
20:    $(AssignedCity, AssigneeSalesman) \leftarrow \arg\max_{ik}(rd_{ik})$
21:    $AssignTheCity(AssignedCity, AssigneeSalesman)$
22:    $NumberofAssignedCities \leftarrow NumberofAssignedCities + 1$
23: end while
24: TaskAssignmentArray is constructed
25: while $NewPermutation(TaskAssignmentArray)exists$ do
26:    $TasksWithOrders \leftarrow NewPermutation(TaskAssignmentArray)$
27:    $CostOfScenario \leftarrow CalculateCost(TasksWithOrders)$
28:    if $CostOfScenario < OptimalCost$ then
29:        $OptimalCost \leftarrow CostOfScenario$
30:        $OptimalSolution \leftarrow TasksWithOrders$
31:    end if
32: end while
```

CHAPTER 5

EXPERIMENTAL RESULTS AND EVALUATIONS

In this chapter, outputs of the relative distances approach and running times will be presented for different number of cities, number of salesmen and starting cities. Current studies and researches are prioritized to compare performances but most of studies tried to solve Single Depot MTSP[52] or Closed Path MTSP [53]. Most of the researches didn't evaluate their performances with the public data sets like TSPLIB[54]. Due to this situation, randomly generated instances are also used and we compared our performance with Brute Force methods' outputs. 3 Groups were determined to make precise evaluation and accurate analyses.

- In this group, Problem Instances with ≤ 16 cities and ≤ 4 salesmen are evaluated. Complete Brute Force method is deployed to determine the performance of Relative Distances Approach.
- Problem instances with > 16 and ≤ 60 cities and $\frac{N}{M} \leq 10$ are in this group. Partial Brute Force method is applied to observe the performance of Relative Distances Approach.
- Problem instances with > 60 cities are in this group. Other studies are used to compare our algorithm's performance. Also, visual outputs are provided to show the feasibility of our solutions.

5.1 Instances with Less Than 17 Cities

Here, we used ulysses16 instance from TSPLIB and E15, which is generated randomly.

For Ulysses16; 2-salesmen, 3-salesmen, 4-salesmen variants are tried. Within these variants, single-depot and multi-depot cases are examined.

You can find the compared stats of relative distances approach and complete brute force method for ulysses16 & 4 salesmen variant in Table 5.1.

Table 5.1: Overall Results for Ulysses16 - 4 Salesmen

Ulysses16 - 4 Salesmen		
Metric	Relative Distances	Brute Force
Average Total Cost	39,330	39,330
Average Makespan	21,3	21,3
Elapsed Time (ms)	0,5	526252

With the relative distances approach, the best solutions are explored within only 0,5 milliseconds (+1.000.000 times faster). In Figure 5.1, some sample outputs of relative distances approach for 4-salesmen variant are given.

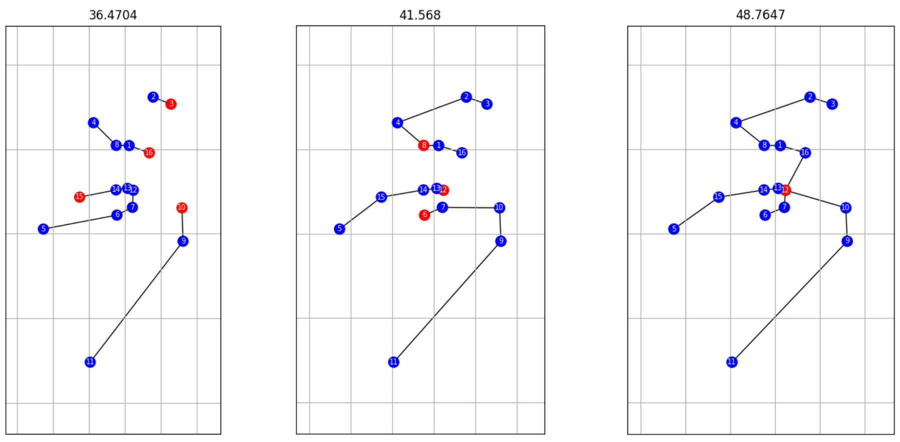


Figure 5.1: Outputs of Ulysses16 with 4 salesmen within different depot scenarios

The left one shows the instance that each salesmen start from different cities (3, 10, 15, 16). In the second version, 2 salesmen start from same city, others start from other cities (6, 8, 8, 12). The last version shows the single depot (12) case. For all types of depot scenarios, the optimal solutions are obtained.

For the 3-salesmen variant of the Ulysses16, the optimal solutions are also obtained by relative distances approach.

You can find the compared stats of relative distances approach and complete brute force method for ulysses16 & 3 salesmen variant in Table 5.2.

Table 5.2: Overall Results for Ulysses16 - 3 Salesmen

Ulysses16 - 3 Salesmen		
Metric	Relative Distances	Brute Force
Average Total Cost	42,583	42,583
Average Makespan	25,871	25,871
Elapsed Time (ms)	0,9	611108

Thanks to the relative distances approach, the best solutions are explored within only 0,9 milliseconds (+600.000 times faster). In Figure 5.2, some sample outputs of relative distances approach for 3-salesmen variant are given.

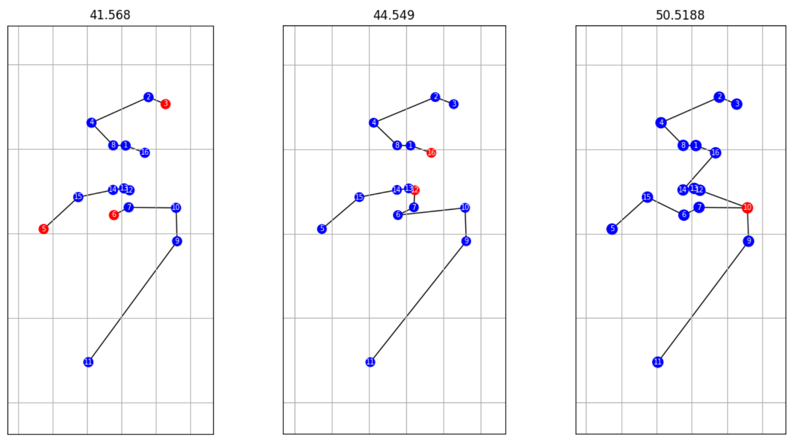


Figure 5.2: Outputs of Ulysses16 with 3 salesmen within different depot scenarios

On the left, all salesmen starts from different cities (3,5,6). Two salesmen starts from 12 and the other starts from 16 at the second instance. All salesmen starts from 10 at the right instance. For multi-depot, multi&single-depot and single-depot variants, optimal results are found in milliseconds.

The relative distances approach didn't produce the optimal result for all Ulysses16 - 2 Salesmen variants. You can see the stats of the experimental results in Table 5.3.

Table 5.3: Overall Results for Ulysses16 - 2 Salesmen

Ulysses16 - 2 Salesmen		
Metric	Relative Distances	Brute Force
Average Total Cost	47,962	47,585
Average Makespan	33,010	33,917
Elapsed Time (ms)	1,2	770993

First, instances that relative distances approach found the optimal solution are shown in Figure 5.3.

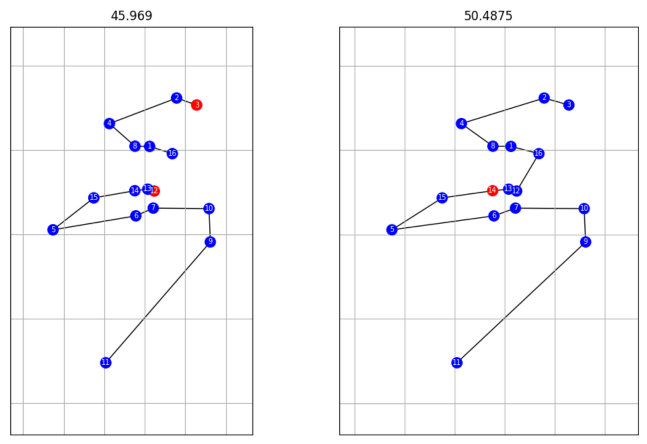


Figure 5.3: Outputs of Ulysses16 with 2 salesmen within different depot scenarios

As mentioned in previous variants, this approach can also find optimal results on single-depot instances and multi-depot instances.

But it's been observed that some instances in which our approach didn't find the optimal result.

Relative Distances Approach has more total cost (49,1281 & 47,6198) than Brute Force method but it has reduced makespan (27,5476 & 31,1749) for the example instance shown in Figure 5.4.

Table 5.4: Relative Distance Approach vs. Brute Force Method

Ulysses16 - 2 Salesmen		
Metric	Relative Distances	Brute Force
Total Cost	49,128	47,620
Makespan	27,547	31,174
Elapsed Time (ms)	1,2	875396

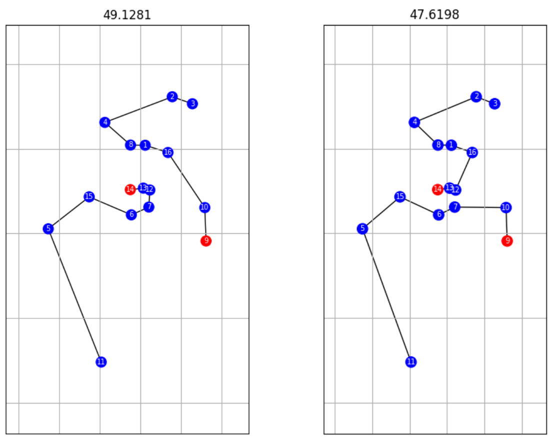


Figure 5.4: Relative Distances Approach vs Brute Force for Ulysses16 with 2 salesmen

Up to this point, it can be said that relative distances approach is not dominated by the brute force method in terms of two different objectives.

In TSPLIB, the minimum size instance is Ulysses16. Other instances have larger sizes and can not be examined by the complete brute force method. Due to this factor, we used a randomly generated instance, E15.

For E15; 1-salesman, 2-salesmen and 3 salesmen variants are observed. Within these options, single-depot and multi-depot cases are examined.

You can find the compared stats of the relative distance approach and complete brute force method used for E15 & 3 salesmen variant in Table 5.5.

Table 5.5: Relative Distance Approach vs. Brute Force Method

E15 - 3 Salesmen		
Metric	Relative Distances	Brute Force
Total Cost	51,005	50,380
Makespan	25,060	27,150
Elapsed Time (ms)	0,7	105696

In Figure 5.5, examples of E15 instances are shown whether the optimal solution is found or not by Relative Distance Approach.

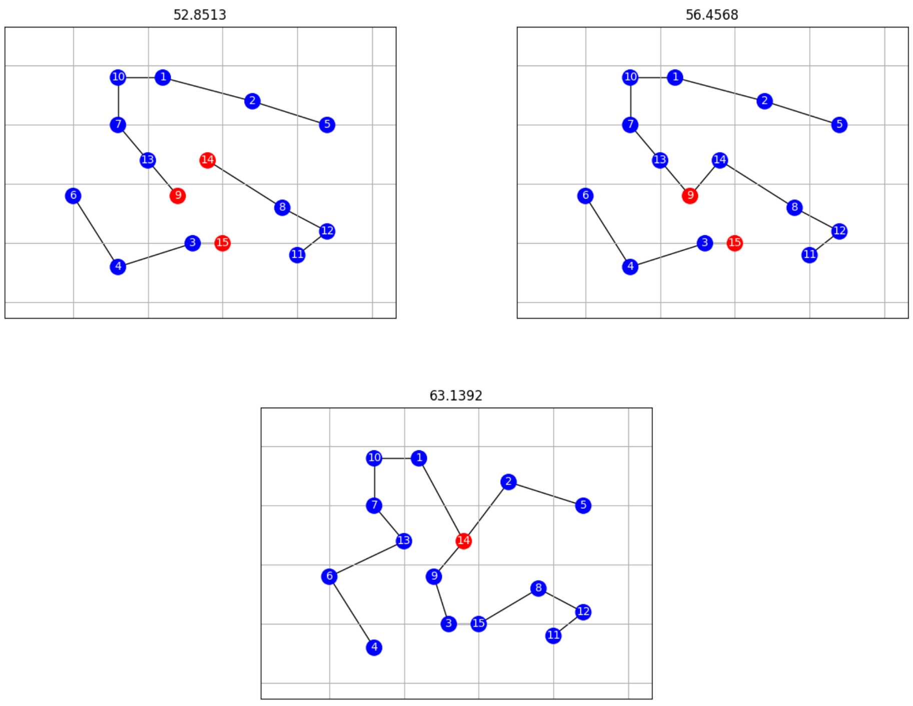


Figure 5.5: Outputs of E15 with 3 salesmen within different depot scenarios

The left one shows the multi depot case, second figure shows 2 starting depots (9,15) and the last one shows the single depot case (14).

The last output is not the cost-optimal solution. Cost-optimal solution has less total cost but more makespan (31,542 &30,761).

You can find the compared stats of the relative distance approach and complete brute

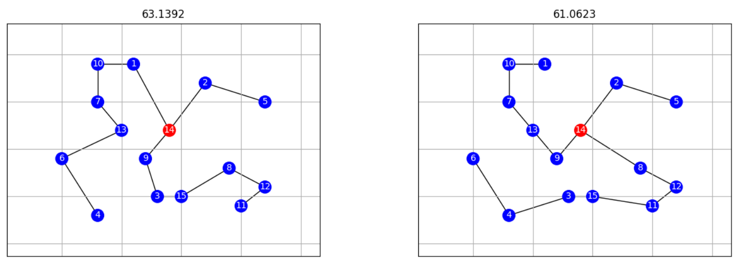


Figure 5.6: Relative Distances Approach vs Brute Force for E15 with 3 salesmen starts from same depot

force method used for E15 & 2 salesmen variant in Table 5.6.

Table 5.6: Relative Distance Approach vs. Brute Force

E15 - 2 Salesmen		
Metric	Relative Distances	Brute Force
Total Cost	58,919	58,644
Makespan	33,882	37,481
Elapsed Time (ms)	0,8	64791

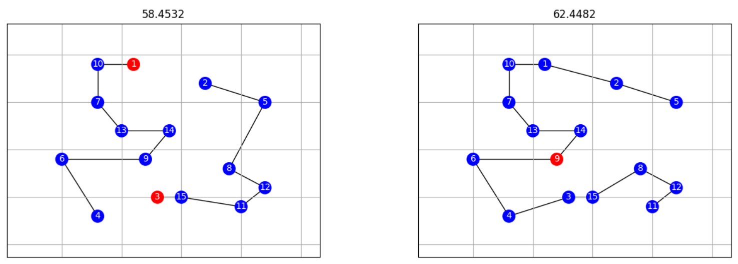


Figure 5.7: Outputs of E15 with 2 salesmen within different depot scenarios

Multi depot & single depot cases shown in Figure 5.7. The first solution is cost-optimal but the second solution is not.

The single depot 2 salesman variant of the E15 didn't solved in terms of the cost-optimality. But again, makespan of the cost-optimal solution is higher than the relative distance approach's output (35,635 vs. 32,527).

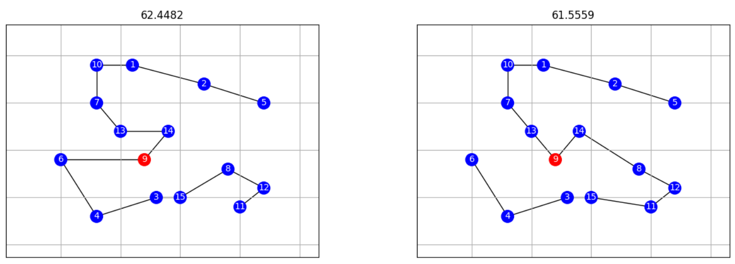


Figure 5.8: Relative Distances Approach vs Brute Force for E15 with 2 salesmen starts from same depot

Relative Distances Approach can also finds solutions to the Traveling Salesman Problem. In this case, it is not the cost-optimal but the 1.66% worse solution is obtained in milliseconds.

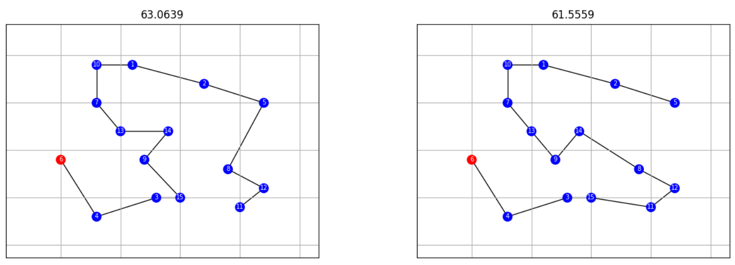


Figure 5.9: Relative Distances Approach vs Brute Force for E15 with 1 salesman

5.2 Instances with between 17 and 60 Cities

From this section, Complete Brute Force approach can not be applicable due to the exponentially growing solution spaces.

We will evaluate our work according to four references in this section and the next section;

- Wang et al's[4] work
- Ndiaye et al's [5] work
- Lou et al's [6] work

- Partial Brute Force Approach

In this section, we examine relative distances approach's performance on eil51 and berlin52 instances from TSPLIB. This evaluation will be based on task orders, alternative task assignment options will not be considered due to the exponentially growing solution space.

You can find the compared stats of the relative distance approach and partial brute force method used for eil51 & 5 salesmen variant in Table 5.7.

Table 5.7: Relative Distance Approach vs. Partial Brute Force

eil51 - 5 Salesman		
Metric	Relative Distances	Brute Force
Total Cost	394,602	394,431
Makespan	113,425	113,425
Elapsed Time (ms)	5	36340

Example problem instances that optimal orders can be found by relative distances approach are given in Figure 5.10.

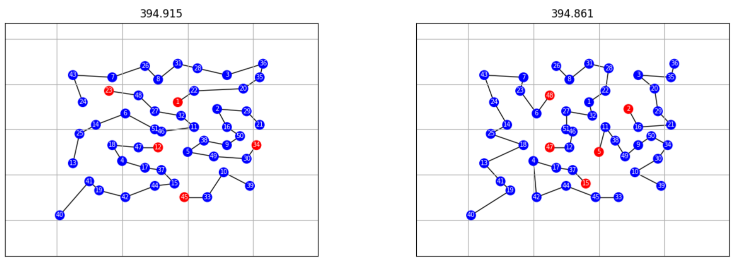
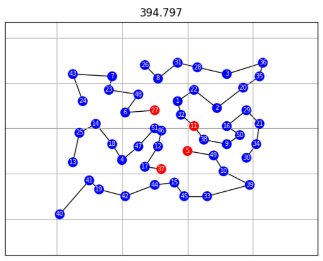


Figure 5.10: Outputs of eil51 with 5 salesmen

There also exist some instances that optimal orders didn't found. In the example shown from Figure 5.11, it can be seen that cities 23,7,43 and 24 re-ordered by the partial brute force approach and reduced cost by 0.513 (0.12%). To guarantee the optimality of orders, much more iterations should be executed and the elapsed time increases by 6800 times (5 to 34300). There is a trade-off between the elapsed time and the total cost here.



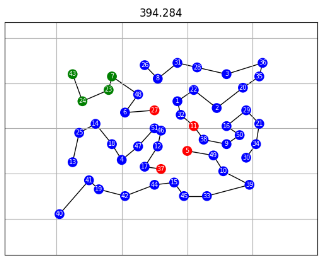


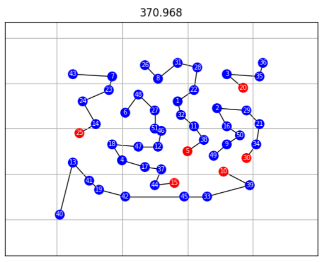
Figure 5.11: Relative Distance Approach vs. Partial Brute Force for 5 salesmen

Detailed stats for different eil51 instances with 6 salesmen are given in the table.

Table 5.8: Relative Distance Approach vs. Partial Brute Force

eil51 - 6 Salesmen		
Metric	Relative Distances	Brute Force
Total Cost	394,26	393,61
Makespan	90,35	89,86
Elapsed Time (ms)	4,7	39411

Example problem instances that optimal orders can be found by relative distances approach are given in Figure 5.12.



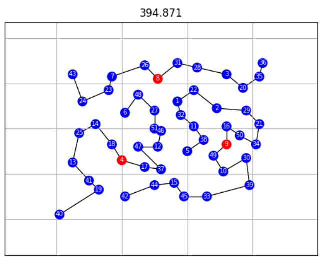


Figure 5.12: Outputs of eil51 with 6 salesmen

On the left, all salesmen started from different cities. Right figure shows the instance that 6 salesmen starts from 3 cities two by two.

There is also an instance that the relative distances approach didn't find the optimals.

In the left side of Figure 5.13, cities 7,23,24 and 43 are also not in an optimal order.

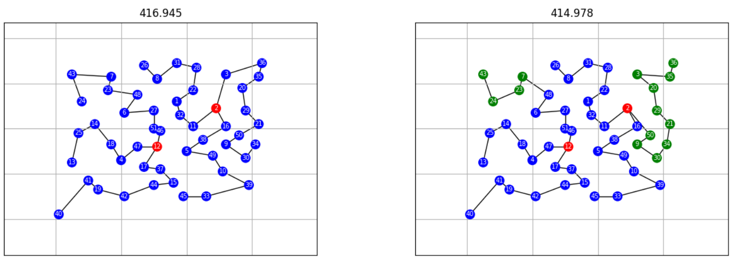


Figure 5.13: Relative Distance Approach vs. Partial Brute Force for 6 Salesman

Also; 50,9,30,34,41,26,20,3,35 and 36 should be re-ordered to find the optimal orders. The cost difference between two results is 1.967 (0.47%). To find the order-optimal solution, 14000 times slower (56700/4) process should be executed.

For the berlin52 instance, all executions obtained the order-optimal results. You can find the detailed stats in 5.9.

Table 5.9: Relative Distance Approach vs. Partial Brute Force for berlin52

berlin52			
Salesmen	Metric	Relative Distances	Brute Force
5	Total Cost	6404,44	6404,44
6	Total Cost	6140,84	6140,84
7	Total Cost	5851,25	5851,25
8	Total Cost	5919,63	5919,63
5	Makespan	2280,9	2280,9
6	Makespan	1489,6	1489,6
7	Makespan	1960,3	1960,3
8	Makespan	1686,9	1686,9
5	Elapsed Time (ms)	5	29526
6	Elapsed Time (ms)	4,3	1755
7	Elapsed Time (ms)	4	241
8	Elapsed Time (ms)	4	47

In Figure 5.14, multi-depot instances and multi&single instances are shown.

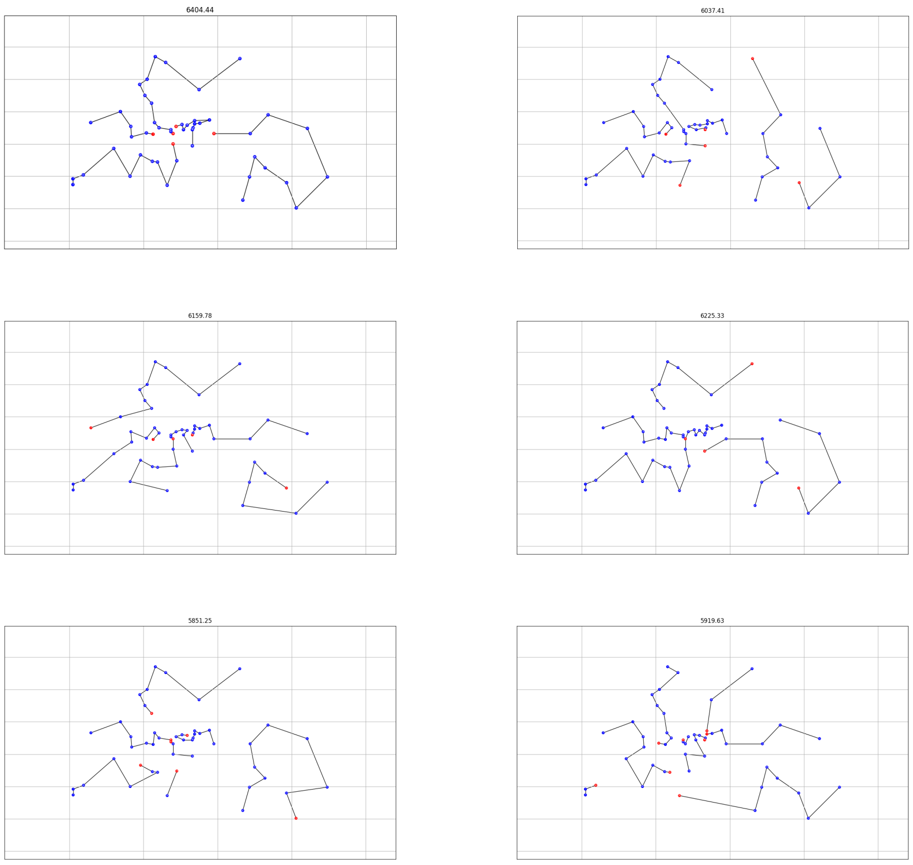


Figure 5.14: Outputs of Relative Distance Approach for berlin52

5.3 Instances with More Than 60 Cities

There doesn't exist any open-path multiple-depot MTSP research on the literature except Wang's, Ndiaye's and Lou's study. Under this condition, comparison tables between these studies and our RDA will be provided first. Then, solutions with larger problem sizes will be visualized and evaluated based on their shapes & intersections.

All these three studies didn't provide the starting cities of the salesmen. Due to this fact, it can be said that they presented the results by the most proper starting cities for them. Our starting cities are not selected to reach the optimality. Beating Wang's, Ndiaye's and Lou's results is sufficient to be presented here. Because, this problem doesn't seek for the optimal starting cities.

According to this table, it can be said that the relative distance method can easily

Table 5.10: Smodel vs RDA in terms of total cost and elapsed time

Comparison between Smodel and RDA			
Instance	Agents	Smodel (cost & ms)	RDA (cost & ms)
eil51	2	403 & 12	400 & 4
eil51	4	365 & 12	363 & 4
eil51	5	356 & 12	352 & 3
berlin52	2	7409 & 11	6705 & 4
berlin52	3	7019 & 11	6764 % 3
berlin52	7	5818 & 11	5607 & 3
pr136	2	101736 & 265	92316 & 21
pr136	3	99358 & 265	90681 & 17
pr136	8	88254 & 265	83745 & 17
st70	3	613 & 34	600 & 7
st70	4	579 & 34	571 & 7
pr439	5	105894 & 1428	102557 & 849
pr439	7	100341 & 1428	99329 & 982
Lin318	2	47065 & 1458	44096 & 610
Lin318	10	40986 & 1458	39539 & 356

provide solution to the different types of problem instances based on their sizes or salesman counts.

The RDA results in Tables 5.10, 5.11 and 5.12 don't show the completely optimal scenarios in the given conditions because MTSP doesn't aim to determine the starting points on the map. RDA finds the proper results in a given instance regardless of the depot counts or salesman distribution.

The results obtained by the relative distances approach are shown at Appendix.

There doesn't exist any study that presented an MTSP algorithm and shown their applications on pr1002 or usa13509 maps. Due to their enormous sizes, their optimalities can not be guaranteed.

As mentioned in the previous chapters, self intersections and intersections between salesmen are the signals of the inoptimal solutions. As can be seen from our applications, there doesn't exist any intersection on pr1002 and usa13509 maps. They can be seen from the Appendix.

Table 5.11: Ndiaye's Transformation Algorithm vs RDA in terms of total cost and elapsed time

Comparison between Transformation Algorithm and RDA			
Instance	Agents	Assaf & Ndiaye (cost & ms)	RDA (cost & ms)
gr24	2	1109 & 6700	1075 & 1
gr24	3	1011 & 3200	956 & 1
gr24	5	900 & 1800	848 & 1
gr48	2	4522 & 10000	4430 & 3
gr48	3	4394 & 10000	4289 % 3
gr48	5	4041 & 23200	3915 & 3
ftv33	2	1141 & 8200	1095 & 2
ftv33	3	1073 & 4600	964 & 2
ftv33	5	973 & 3400	845 & 2
ftv35	2	1283 & 20900	1257 & 2
ftv35	3	1193 & 16500	1128 & 2
ftv35	5	1094 & 3500	1013 & 2
ftv38	2	1345 & 9700	1311 & 2
ftv38	3	1259 & 51400	1240 & 2
ftv38	5	1182 & 12400	1118 & 2
ftv44	2	1460 & 184400	1450 & 3
ftv44	3	1407 & 14100	1355 & 2
ftv44	5	1317 & 5500	1224 & 2
ftv47	2	1623 & 290000	1567 & 3
ftv47	3	1540 & 13400	1483 & 3
ftv47	5	1471 & 36000	1330 & 2

Table 5.12: Lou's Partheno-Genetic Algorithm vs RDA in terms of total cost and elapsed time

Comparison between Partheno-Genetic Algorithm and RDA			
Instance	Agents	Lou (cost)	RDA (cost)
eil51	3	409	378
eil76	3	526	504

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

6.1 Conclusions

In this thesis, we worked on the Multi-Depot Open-Path Multi Traveling Salesman Problem. In general, there are two different objectives for the problem; minimizing the total distance covered by all salesmen and minimizing the maximum distance covered by a salesman. Our aim is to minimize the total distance and to keep the maximum distance (makespan) in a reasonable level.

Due to the combinatorial structure of the problem, optimal solutions can not be found in polynomial time. The fact makes the problem insolvable in real life applications. To reach the optimal or near-optimal solutions, greedy heuristics are deployed in the thesis.

Apart from the other researches, tour method didn't split the problem into two sections as task assignment and the arranging the assigned tasks in an order. We proposed the Relative Distances Approach that combines these two phases into one stage.

Relative Distances Approach assigns a city to a salesman with respect to the three parameters.

- Sum of marginal insertion costs to salesmen's paths.
- Minimum marginal insertion cost to a salesman's path.
- Calculated distance of the salesman with the minimum marginal insertion cost.

By the parameters given above, relative distance coefficients are calculated for each salesman & city pair in. The pair with the highest coefficient is matched and the calculation continues iteratively until all cities are assigned to salesmen. Complexity of this step is O($\frac{N^3}{M}$). It means that the problem is solved in polynomial time.

After the iterative assignment step, task swapping and order reversing heuristics are applied to reduce the total distance covered by the salesmen.

Performance of the solutions provided by Relative Distances Approach is evaluated by different methods.

- First, outputs are compared with Wang's [4], Ndiaye's [5] and Lou's [6] research.
- For the smaller problem instances, the outputs are evaluated by the complete brute force algorithm which all possible task assignment and sequencing scenarios are examined.
- In the moderate size problem instances, the results are evaluated by the partial brute force algorithm which looks at the alternative sequencing options only.
- The extreme-size instances are visualized and evaluated by their intersection-status.

6.2 Future Work

To make the problem more applicable in real-life, some constraints and coefficients will be added into the problem.

- Priority of the tasks will have an importance in the problem. As mentioned in the introduction, disaster management is an application area of the MTSP. Giving importance coefficients according to damage levels of the locations will ensure the applicability.
- Deadlines will be inserted into a problem, it can be important in real life scenarios like same-day delivery.

REFERENCES

[1] B. Korte and J. Vygen, *Combinatorial Optimization: Theory and Algorithms*. Algorithms and Combinatorics, Springer Berlin Heidelberg, 2012.

[2] *The Traveling Salesman Problem*, pp. 527–562. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008.

[3] Wikipedia contributors, "2023 turkey–syria earthquakes — Wikipedia, the free encyclopedia." `https://en.wikipedia.org/w/index.php?title=2023_Turkey%E2%80%93Syria_earthquakes&oldid=1190109763`, 2023. [Online; accessed 24-December-2023].

[4] X. Wang, D. Liu, and M. Hou, "A novel method for multiple depot and open paths, multiple traveling salesmen problem," in *2013 IEEE 11th International Symposium on Applied Machine Intelligence and Informatics (SAMI)*, pp. 187–192, 2013.

[5] M. Assaf and M. Ndiaye, "Solving an open path multiple depot multiple traveling salesman problem after transformation," *IEEE*, 2017.

[6] P. Lou, K. Xu, J. Yan, and Z. Xiao, "An improved partheno-genetic algorithm for open path multi-depot multiple traveling salesmen problem," *Journal of Physics: Conference Series*, vol. 1848, p. 012002, apr 2021.

[7] *Der Handlungsreisende wie er sein soll und was er zu thun hat, um Aufträge zu erhalten und eines glücklichen Erfolgs in seinen Geschäften gewiß zu sein ; Mit e. Titelkupf.* Voigt, 1832.

[8] N. Biggs, Lloyd, and R. J. Wilson, *Graph theory, 1736-1936*. Clarendon Press, Feb. 1986.

[9] H. Hahn, "Ergebnisse eines mathematischen kolloquiums," *Monatshefte für Mathematik und Physik*, vol. 39, p. A8, 1932.

[10] G. Dantzig, R. Fulkerson, and S. Johnson, “Solution of a large-scale traveling-salesman problem,” *Operations Research*, vol. 2, no. 4, pp. 393–410, 1954.

[11] N. Christofides, “Worst-case analysis of a new heuristic for the travelling salesman problem,” Technical Report 388, Graduate School of Industrial Administration, Carnegie Mellon University, 1976.

[12] S. O. Gharan, A. Saberi, and M. Singh, “A randomized rounding approach to the traveling salesman problem,” in *Proceedings of 52nd Annual IEEE Symposium on Foundations of Computer Science, FOCS 2011*, IEEE, January 2011. Awarded Best Paper.

[13] C. E. Miller, A. W. Tucker, and R. A. Zemlin, “Integer programming formulation of traveling salesman problems,” *J. ACM*, vol. 7, p. 326–329, oct 1960.

[14] M. Diaby, “The traveling salesman problem: A linear programming formulation of,” *ArXiv*, vol. abs/cs/0609005, 2006.

[15] I. Kara, O. N. Koc, F. Altıparmak, and B. Dengiz, “New integer linear programming formulation for the traveling salesman problem with time windows: minimizing tour duration with waiting times,” *Optimization*, vol. 62, no. 10, pp. 1309–1319, 2013.

[16] P. Hungerländer and C. Truden, “Efficient and easy-to-implement mixed-integer linear programs for the traveling salesperson problem with time windows,” *Transportation Research Procedia*, vol. 30, pp. 157–166, 2018. EURO Mini Conference on "Advances in Freight Transportation and Logistics".

[17] E. Balas and P. Toth, “Branch and bound methods for the traveling salesman problem,” 1983.

[18] S. Poikonen, B. Golden, and E. A. Wasil, “A Branch-and-Bound Approach to the Traveling Salesman Problem with a Drone,” *INFORMS Journal on Computing*, vol. 31, pp. 335–346, April 2019.

[19] B. Freisleben and P. Merz, “A genetic local search algorithm for solving symmetric and asymmetric traveling salesman problems,” in *Proceedings of IEEE International Conference on Evolutionary Computation*, pp. 616–621, 1996.

[20] Y. Deng, J. Xiong, and Q. Wang, “A hybrid cellular genetic algorithm for the traveling salesman problem,” *Mathematical Problems in Engineering*, 2021.

[21] E. Osaba, X.-S. Yang, F. Diaz, P. Lopez-Garcia, and R. Carballedo, “An improved discrete bat algorithm for symmetric and asymmetric traveling salesman problems,” *Engineering Applications of Artificial Intelligence*, vol. 48, pp. 59–71, 2016.

[22] X. Geng, Z. Chen, W. Yang, D. Shi, and K. Zhao, “Solving the traveling salesman problem based on an adaptive simulated annealing algorithm with greedy search,” *Applied Soft Computing*, vol. 11, no. 4, pp. 3680–3689, 2011.

[23] G. Dantzig and J. Hamser, “The truck dispatching problem,” *Management Science*, 1959.

[24] G. Clarke and J. W. Wright, “Scheduling of Vehicles from a Central Depot to a Number of Delivery Points,” *Operations Research*, vol. 12, pp. 568–581, August 1964.

[25] S. Gorenstein, “Printing press scheduling for multi-edition periodicals,” *Manage. Sci.*, vol. 16, pp. B–373–B–383, feb 1970.

[26] A. E. Carter and C. T. Ragsdale, “Scheduling pre-printed newspaper advertising inserts using genetic algorithms,” *Omega*, vol. 30, pp. 415–421, 2002.

[27] R. D. Angel, W. L. Caudle, R. Noonan, and A. Whinston, “Computer-assisted school bus scheduling,” *Management Science*, vol. 18, no. 6, pp. B279–B288, 1972.

[28] J. Park and B.-I. Kim, “The school bus routing problem: A review,” *European Journal of Operational Research*, vol. 202, no. 2, pp. 311–319, 2010.

[29] O. Cheikhrouhou, A. Koubaa, and A. Zarrad, “A cloud based disaster management system,” *Journal of Sensor and Actuator Networks*, vol. 9, no. 1, 2020.

[30] B. Bodaghi, E. Palaneeswaran, S. Shahparvari, and M. Mohammadi, “Probabilistic allocation and scheduling of multiple resources for emergency operations; a victorian bushfire case study,” *Computers, Environment and Urban Systems*, vol. 81, p. 101479, 2020.

[31] B. Bodaghi, S. Shahparvari, M. Fadaki, K. H. Lau, P. Ekambaram, and P. Chhetri, “Multi-resource scheduling and routing for emergency recovery operations,” *International Journal of Disaster Risk Reduction*, vol. 50, p. 101780, 2020.

[32] M. Kaspi, M. Zofi, and R. Teller, “Maximizing the profit per unit time for the travelling salesman problem,” *Comput. Ind. Eng.*, vol. 135, p. 702–710, sep 2019.

[33] P. Stodola, P. Otrísal, and K. Hasilová, “Adaptive ant colony optimization with node clustering applied to the travelling salesman problem,” *Swarm Evol. Comput.*, vol. 70, p. 101056, 2022.

[34] Q. M. Ha, Y. Deville, Q. D. Pham, and M. H. Hà, “A hybrid genetic algorithm for the traveling salesman problem with drone,” *Journal of Heuristics*, vol. 26, pp. 219–247, April 2020.

[35] P. M. FRANÇA, M. GENDREAU, G. LAPORTE, and F. M. MÜLLER, “The m-traveling salesman problem with minmax objective,” *Transportation Science*, vol. 29, no. 3, pp. 267–275, 1995.

[36] J. Park, S. Bakhtiyarov, and J. Park, “Schedulenet: Learn to solve minmax m{tsp} using reinforcement learning with delayed reward,” 2021.

[37] M. Takafumi and N. Kazumiti, “Solving min-max multiple traveling salesman problems by chaotic neural network,” *IEICE Proceeding Series*, vol. 46, pp. 237–240, 09 2014.

[38] I. Vandermeulen, R. Groß, and A. Kolling, “Balanced task allocation by partitioning the multiple traveling salesperson problem,” in *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS ’19, (Richland, SC), p. 1479–1487, International Foundation for Autonomous Agents and Multiagent Systems, 2019.

[39] J.-K. Hao, “Hybrid search with neighborhood reduction for the multiple traveling salesman problem,” *Computers Operations Research*, 2022.

[40] J. A. Cornejo-Acosta, J. García-Díaz, J. C. Pérez-Sansalvador, and C. Segura, "Compact integer programs for depot-free multiple traveling salesperson problems," *Mathematics*, vol. 11, no. 13, 2023.

[41] J. Zheng, Y. Hong, W. Xu, W. Li, and Y. Chen, "An effective iterated two-stage heuristic algorithm for the multiple traveling salesmen problem," *Computers Operations Research*, vol. 143, p. 105772, 2022.

[42] X. Chen, P. Zhang, G. Du, and F. Li, "Ant colony optimization based memetic algorithm to solve bi-objective multiple traveling salesmen problem for multi-robot systems," *IEEE Access*, vol. 6, pp. 21745–21757, 2018.

[43] Y. Carreno, J. H. A. Ng, Y. Petillot, and R. Petrick, "Planning, execution, and adaptation for multi-robot systems using probabilistic and temporal planning," in *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '22, (Richland, SC), p. 217–225, International Foundation for Autonomous Agents and Multiagent Systems, 2022.

[44] W. Garn, "Balanced dynamic multiple travelling salesmen: Algorithms and continuous approximations," *Computers Operations Research*, vol. 136, p. 105509, 2021.

[45] Y. Harrath, A. F. Salman, A. Alqaddoumi, H. Hasan, and A. Radhi, "A novel hybrid approach for solving the multiple traveling salesmen problem," *Arab Journal of Basic and Applied Sciences*, vol. 26, no. 1, pp. 103–112, 2019.

[46] A. T. Al-Taani, "Solving the multiple traveling salesman problem using memetic algorithm," *Artificial Intelligence Evolution*, 2022.

[47] J. K. Thenepalle and P. Singamsetty, "An open close multiple travelling salesman problem with single depot," *Decision Science Letters*, 2019.

[48] Y. Wang, Y. Chen, and Y. Lin, "Memetic algorithm based on sequential variable neighborhood descent for the minmax multiple traveling salesman problem," *Comput. Ind. Eng.*, vol. 106, p. 105–122, apr 2017.

[49] S. Ghafurian and N. Javadian, "An ant colony algorithm for solving fixed destination multi-depot multiple traveling salesmen problems," *Appl. Soft Comput.*, vol. 11, pp. 1256–1262, 2011.

[50] F. Semiz, M. A. Yorganci, and F. Polat, “Solving an industry-inspired generalization of lifelong MAPF problem including multiple delivery locations,” *Adv. Eng. Informatics*, vol. 57, p. 102026, 2023.

[51] R. C. Larson and A. R. Odoni, *Urban Operations Research - 6.4.7 Euclidean TSP*. Prentice-Hall, 1981.

[52] S. Y. Yang Shuai and Z. Kai, “An effective method for solving multiple travelling salesman problem based on nsga-ii,” *Systems Science & Control Engineering*, vol. 7, no. 2, pp. 108–116, 2019.

[53] M. Burger, Z. Su, and B. De Schutter, “A node current-based 2-index formulation for the fixed-destination multi-depot travelling salesman problem,” *European Journal of Operational Research*, vol. 265, no. 2, pp. 463–477, 2018.

[54] G. Reinelt, “TSPLIB–a traveling salesman problem library,” *ORSA Journal on Computing*, vol. 3, no. 4, pp. 376–384, 1991.

APPENDIX A

APPENDIX FROM SMODEL & RDA COMPARISON

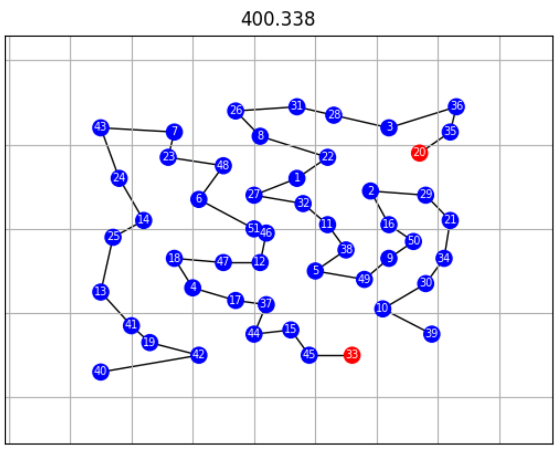


Figure A.1: Output from the 1st line of Table 5.10

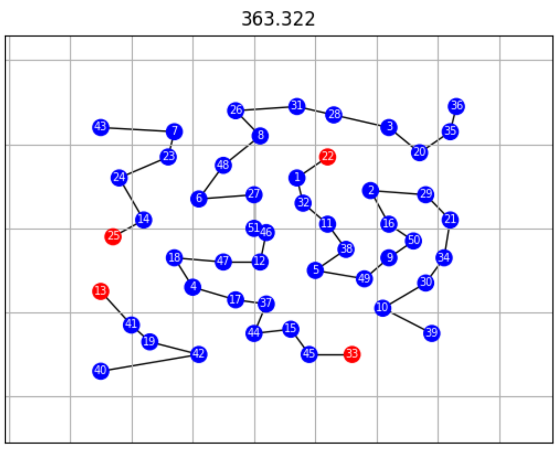


Figure A.2: Output from the 2nd line of Table 5.10

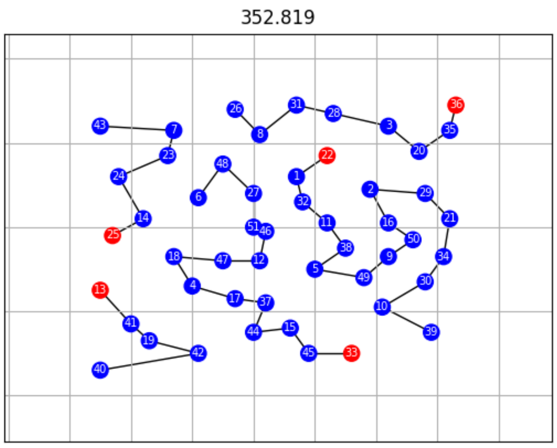


Figure A.3: Output from the 3rd line of Table 5.10

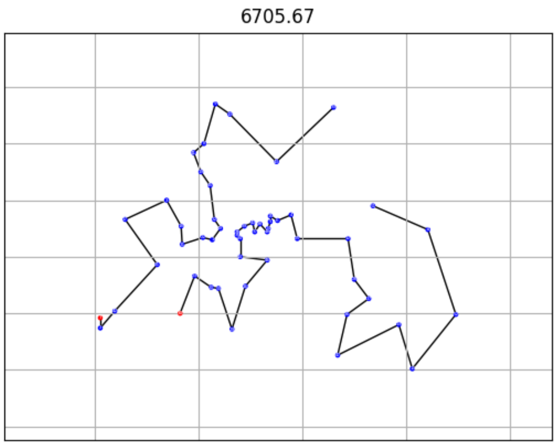


Figure A.4: Output from the 4th line of Table 5.10

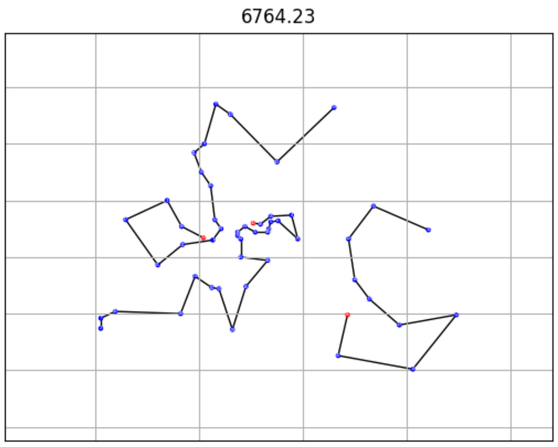


Figure A.5: Output from the 5th line of Table 5.10

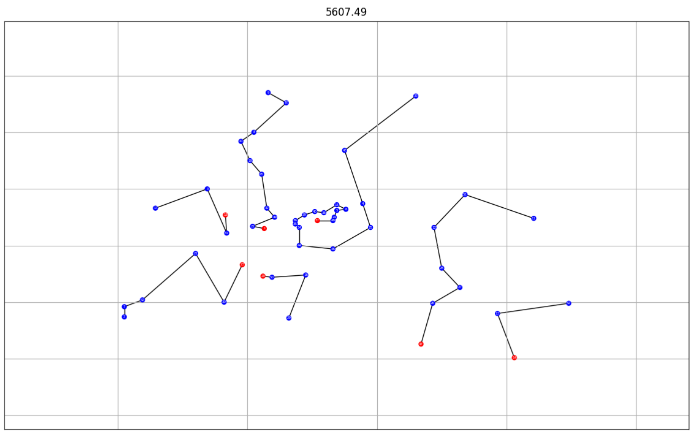


Figure A.6: Output from the 6th line of Table 5.10

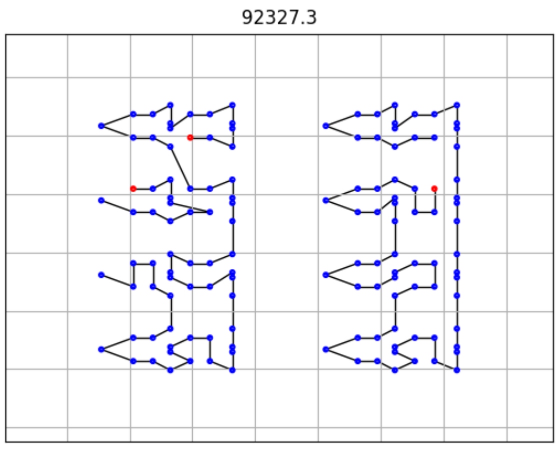


Figure A.7: Output from the 7th line of Table 5.10

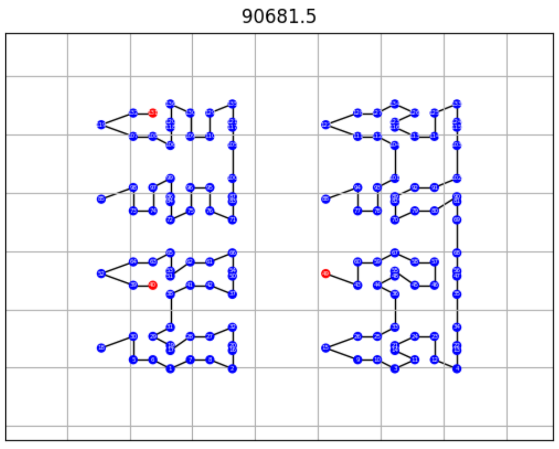


Figure A.8: Output from the 8th line of Table 5.10

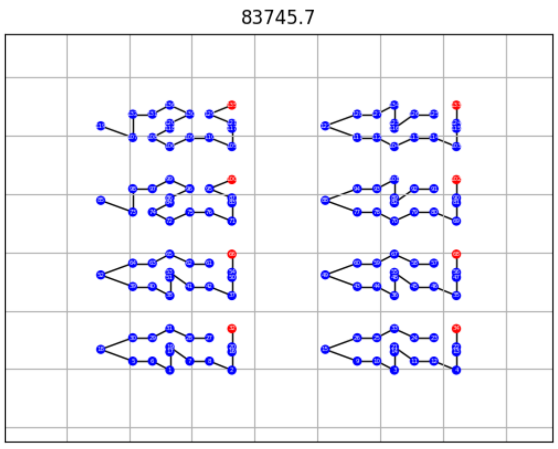


Figure A.9: Output from the 9th line of Table 5.10

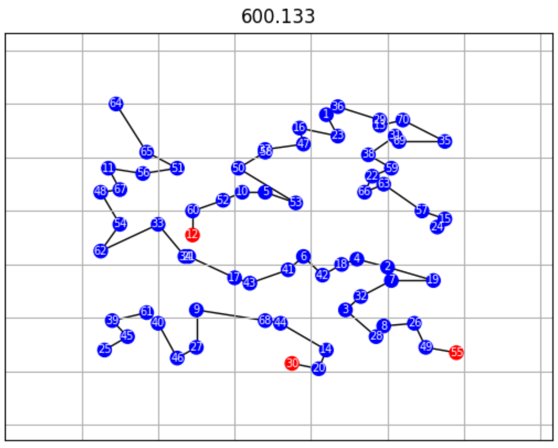


Figure A.10: Output from the 10th line of Table 5.10

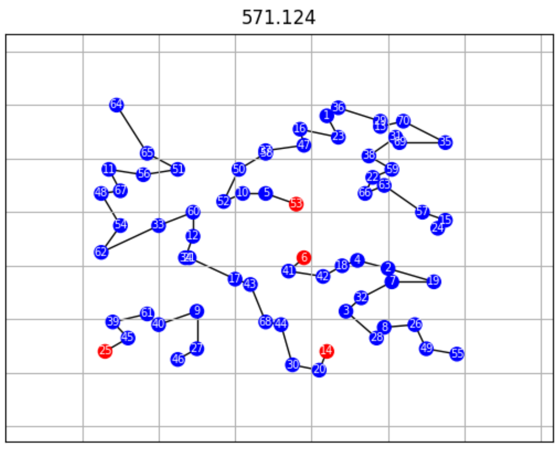


Figure A.11: Output from the 11th line of Table 5.10

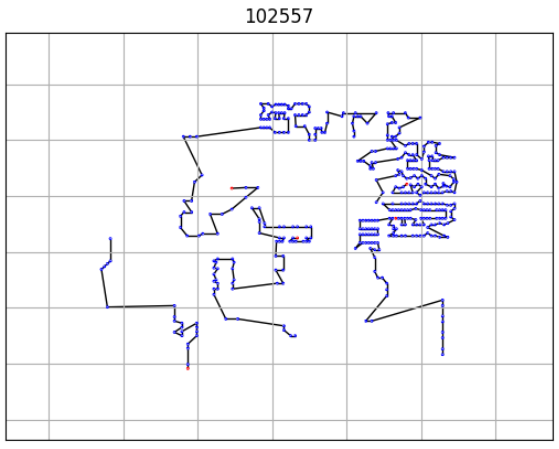


Figure A.12: Output from the 12th line of Table 5.10

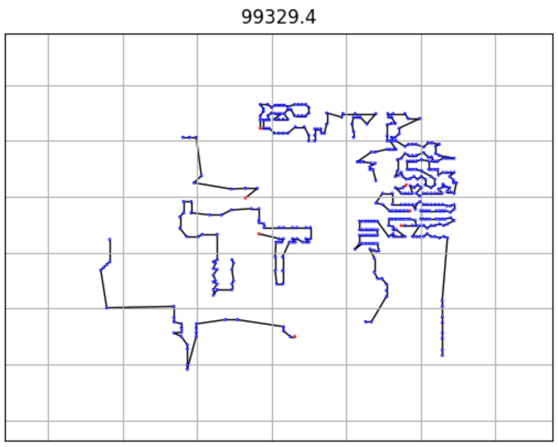


Figure A.13: Output from the 13th line of Table 5.10

APPENDIX B

APPENDIX FOR LARGER INSTANCES

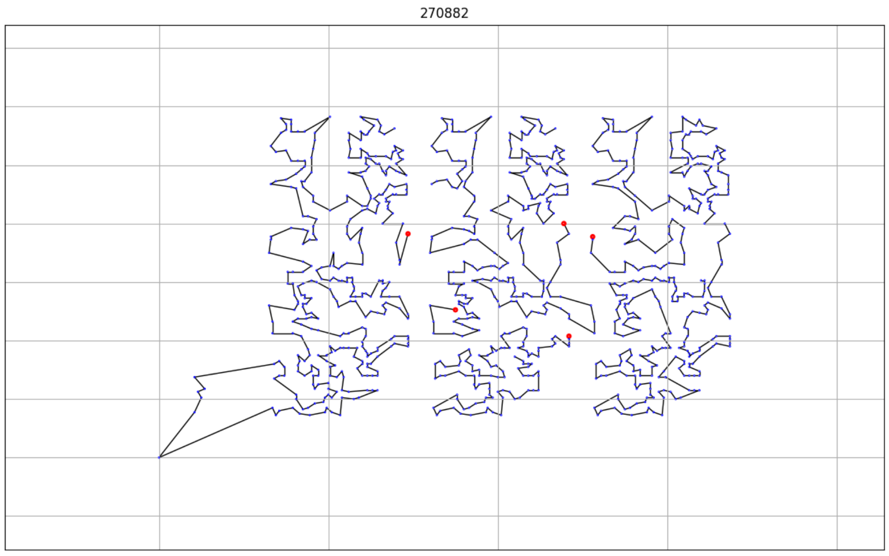


Figure B.1: RDA's solution to the pr1002 instance with 5 salesmen found in 30 secs.

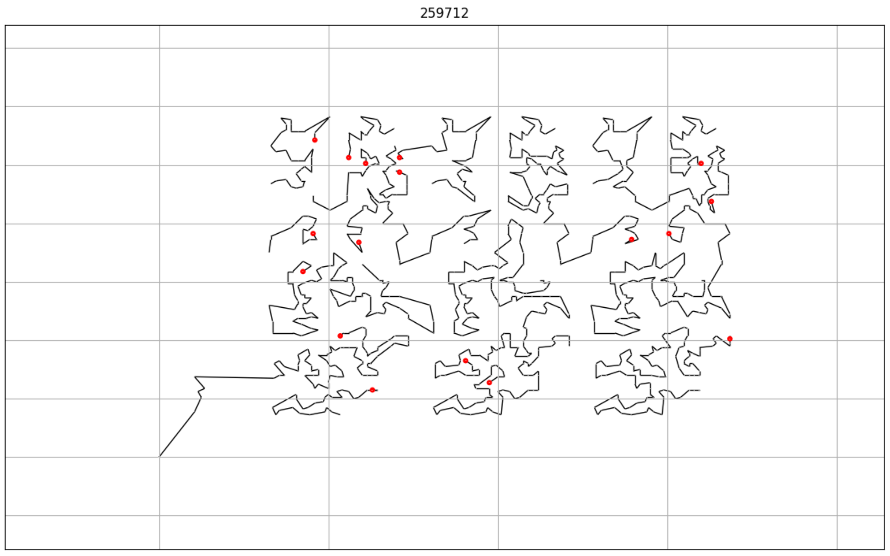


Figure B.2: RDA's solution to the pr1002 instance with 17 salesmen found in 16 secs.

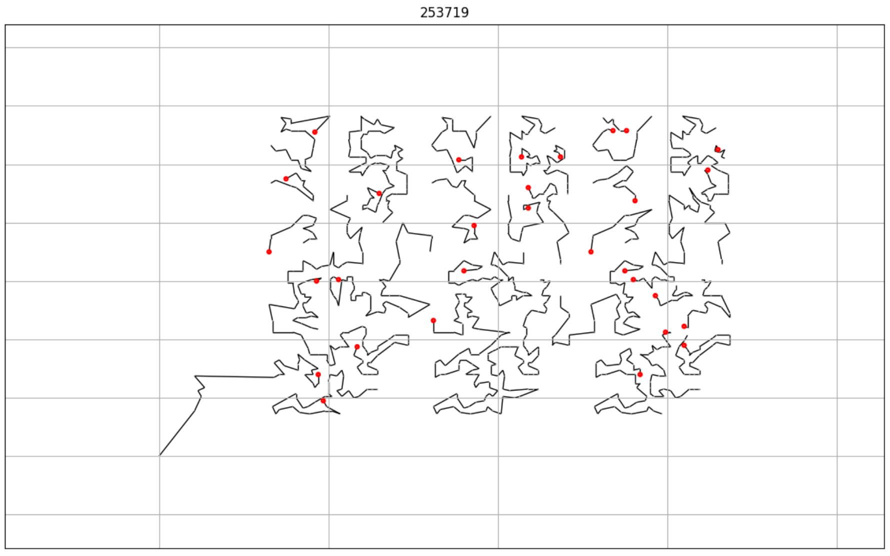


Figure B.3: RDA's solution to the pr1002 instance with 30 salesmen found in 9 secs.

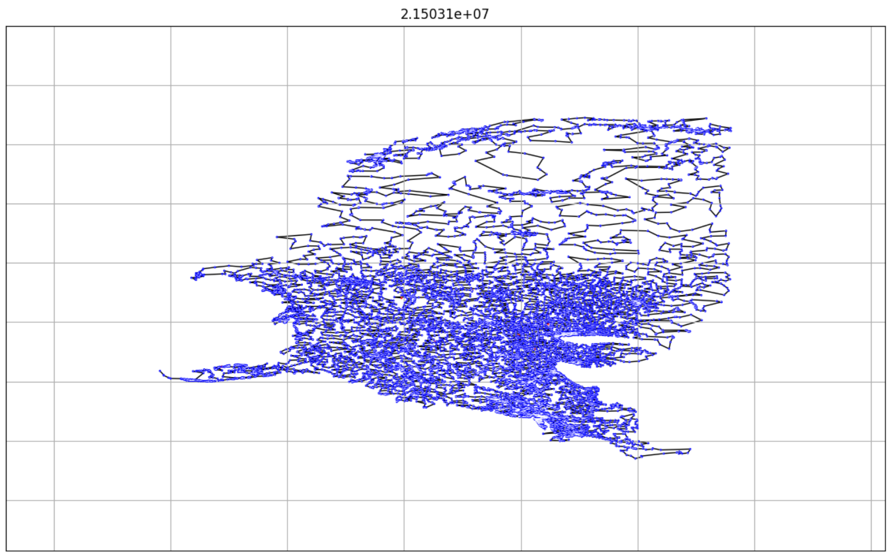


Figure B.4: RDA's solution to the usa13509 instance with 10 salesmen found in 1214 secs.

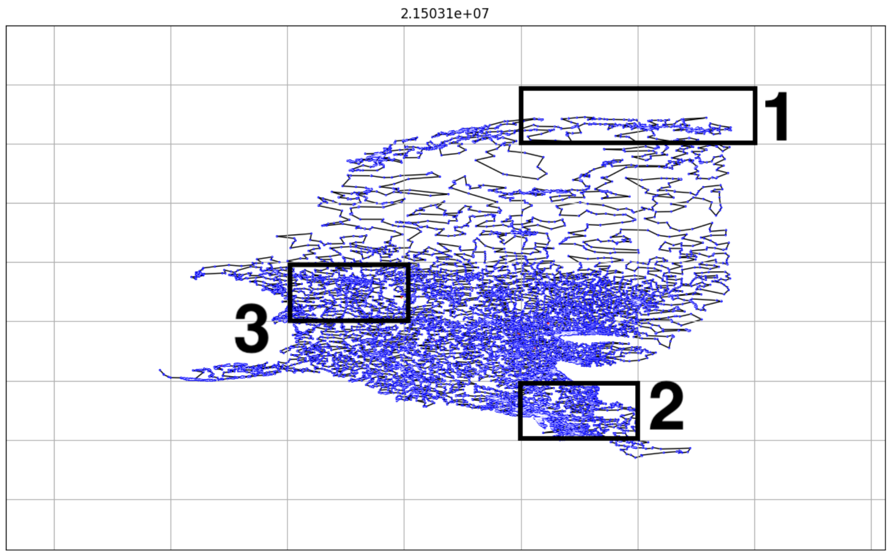


Figure B.5: From the visual, it can be misunderstood that there are intersections in some regions

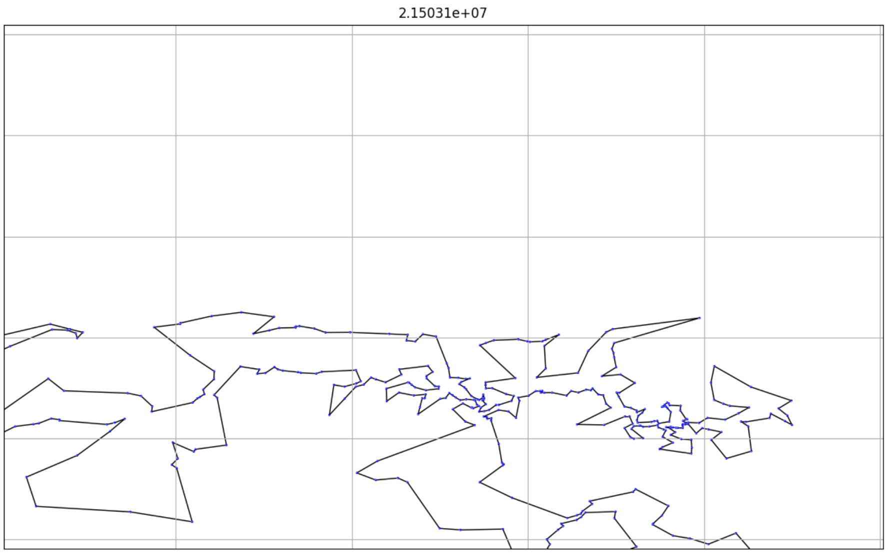


Figure B.6: There is no intersection in the first rectangle

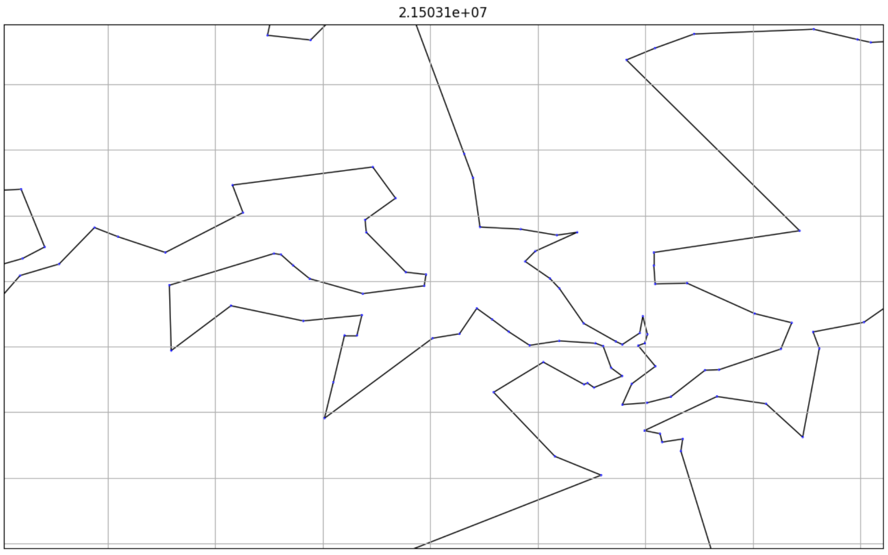


Figure B.7: Zoomed in more to the region 1

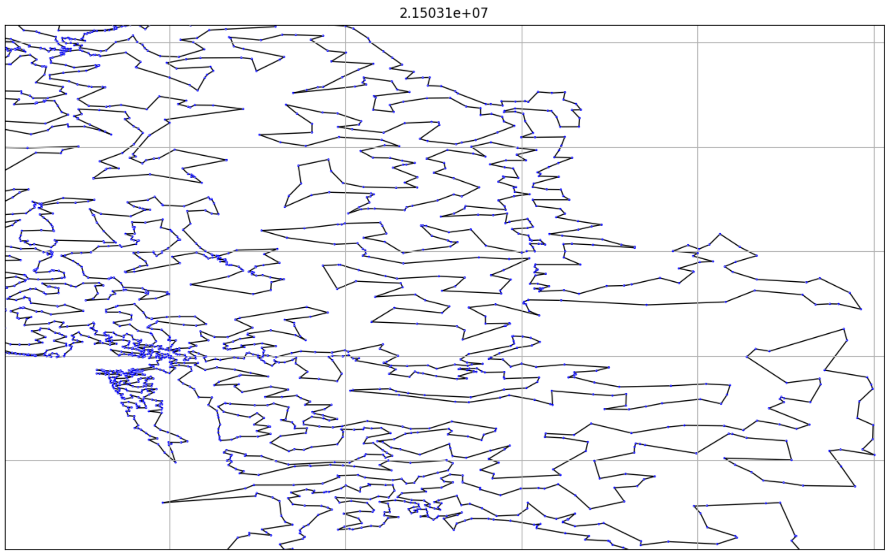


Figure B.8: There is no intersection in the second rectangle

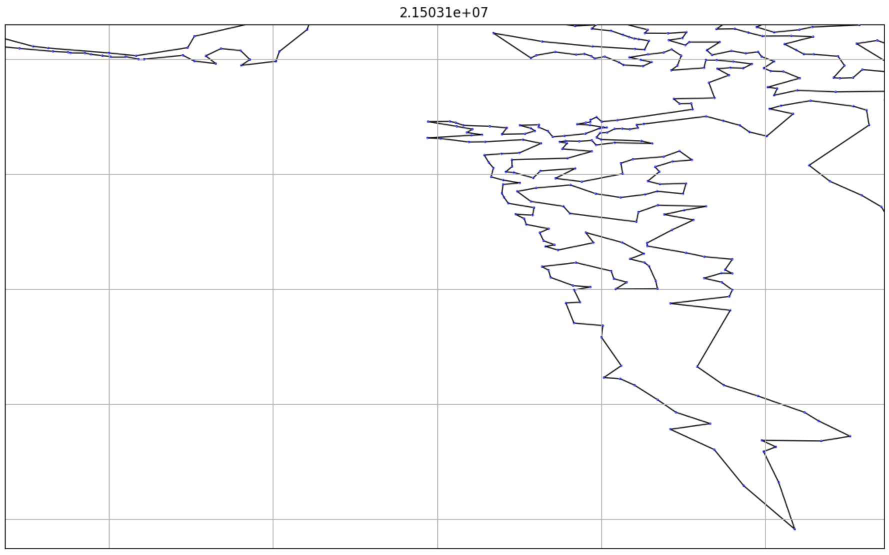


Figure B.9: Zoomed in more to the region 2

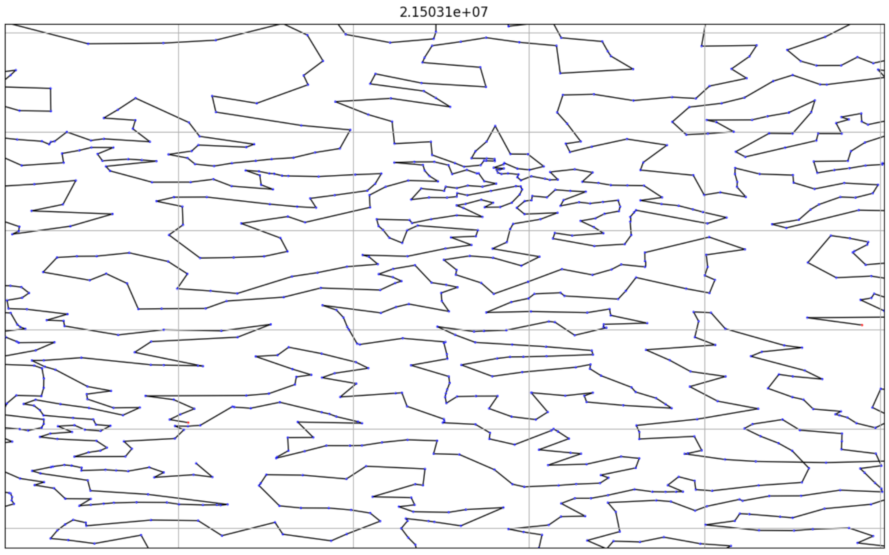


Figure B.10: There is no intersection in the third rectangle