T.R.
SAKARYA UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

ANT COLONY OPTIMIZATION AND GREEDY ALGORITHM PERFORMANCE COMPARISON IN TRAVELLING SALESMAN PROBLEM

MSc THESIS

Merve Ece GÖRGÜN

Industrial Engineering Department

Industrial Engineering Program

MAY 2024

T.R.
SAKARYA UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

ANT COLONY OPTIMIZATION AND GREEDY ALGORITHM PERFORMANCE COMPARISON IN TRAVELLING SALESMAN PROBLEM

MSc THESIS

Merve Ece GÖRGÜN

Industrial Engineering Department

Thesis Advisor: Associate Prof. BERRİN DENİZHAN

MAY 2024

The thesis work titled "Ant Colony Optimization and Greedy Algorithm Performance Comparison in Travelling Salesman Problem" prepared by Merve Ece Görgün was accepted by the following jury on 16/05/2024 by unanimously/majority of votes as a MSc THESIS in Sakarya University Graduate School of Natural and Applied Sciences, Industrial Engineering department.

Thesis Jury

Head of Jury : **Prof.Dr. Harun Reşit Yazgan**
Sakarya University

Jury Member : **Doç.Dr.Berrin Denizhan (Advisor)**
Sakarya University

Jury Member : **Prof.Dr. Alpaslan Fığlalı**
Kocaeli University

STATEMENT OF COMPLIANCE WITH THE ETHICAL PRINCIPLES AND RULES

I declare that the thesis work titled " ANT COLONY OPTIMIZATION AND GREEDY ALGORITHM PERFORMANCE COMPARISON IN TRAVELLING SALESMAN PROBLEM ", which I have prepared in accordance with Sakarya University Graduate School of Natural and Applied Sciences regulations and Higher Education Institutions Scientific Research and Publication Ethics Directive, belongs to me, is an original work, I have acted in accordance with the regulations and directives mentioned above at all stages of my study, I did not get the innovations and results contained in the thesis from anywhere else, I duly cited the references for the works I used in my thesis, I did not submit this thesis to another scientific committee for academic purposes and to obtain a title, in accordance with the articles 9/2 and 22/2 of the Sakarya University Graduate Education and Training Regulation published in the Official Gazette dated 20.04.2016, a report was received in accordance with the criteria determined by the graduate school using the plagiarism software program to which Sakarya University is a subscriber, I accept all kinds of legal responsibility that may arise in case of a situation contrary to this statement.

(16/05/2024)

Merve Ece Görgün

To my dearest family.

ACKNOWLEDGMENTS

In profound gratitude, I extend my appreciation to Dr. Berrin Denizhan, my dedicated and insightful advisor. Her guidance, support, and expertise have been indispensable throughout this journey. Dr. Denizhan's unwavering commitment to excellence and her constructive feedback have played a pivotal role in shaping the outcome of this work. Many thanks for her encouragement, patience, and mentorship, which have been instrumental in successfully completing this thesis. Her support has not only enriched my academic experience but has also inspired me to strive for excellence in my future endeavors.

I would also like to acknowledge Mr. Abdullah Hulusi Kökçam, my Matlab mentor during my master's studies, for his proficiency in the subject and willingness to share knowledge, which significantly contributed to my understanding of Matlab and its applications.

Additionally, sincere gratitude goes to the Industrial Engineering Department at Sakarya University for providing a conducive academic environment and resources that facilitated the realization of this project.

Merve Ece Görgün

TABLE OF CONTENTS

Page

ABBREVIATIONS

ACO	: Ant Colony Optimization
GA	: Genetic Algorithm
MATLAB	: Matrix Laboratory
MST	: Minimum Spanning Tree
SP	: Shortest Path
RNG	: Random Number Generator
TSP	: Travelling Salesman Problem

SYMBOLS

N	: Number of Seeds
P	: Path
$\boldsymbol{p}$	: Parameter of Statistical Significance
t	: The size of the difference between the groups' means and how different the groups are from each other.
$\bar{\mathbf{d}}$	: The average value of the difference between two conditions for the specified sample
$\boldsymbol{s}_{\mathbf{d}}$	: Standard deviation of differences
n	: Sample Size
$\boldsymbol{N_{ij}}$	: Probability value of ACO
$\boldsymbol{d_{ij}}$	: Distance parameter of ACO
η (eta)	: Heuristic information (reverse of distance)
$\boldsymbol{\tau}$ **(tau)**	: pheromone level between two nodes
$\boldsymbol{\alpha}$ **(alpha)**	: the effect of pheromone level
β (beta)	: sezgisel bilginin etkisi

LIST OF TABLES

Page

LIST OF FIGURES

Page

ACO AND GREEDY ALGORITHM PERFORMANCE COMPARISON IN TRAVELLING SALESMAN PROBLEM

SUMMARY

Key Words: Ant Colony Optimization (ACO), Traveling Salesman Problem (TSP), Greedy Algorithm, Network Theory, Combinatorial Optimization.

Nestled within the intricate realm of combinatorial optimization, the TSP emerges as a focal point, not merely entangled in theoretical complexities but also exerting a substantial influence on the practical terrains of logistics and supply chain management. This thesis embarks on a comparative journey, intricately exploring the efficacies of two prominent optimization methodologies—ACO and the Greedy Algorithm—in unraveling the subtleties of the TSP. Recognizing the pivotal role of streamlined route planning in modern supply chains, the study meticulously examines the theoretical underpinnings and practical ramifications of these algorithms.

The methodological framework for this study encompasses the execution and assessment of ACO and the Greedy Algorithm through Matlab programming. The algorithms are meticulously executed within the Matlab environment, allowing for a precise examination of their performance.

The comparative tests encompass a set of meticulously crafted scenarios for evaluating the algorithms' efficiency in addressing the TSP. The Matlab platform serves as the computational engine, enabling the execution of algorithmic processes and facilitating an in-depth analysis of key metrics, including time and cost. The choice of Matlab as the programming environment ensures a standardized and rigorous comparison, providing a robust foundation for deriving meaningful insights into the capabilities and limitations of both ACO and the Greedy Algorithm.

The study yields illuminating results by closely examining pivotal metrics, such as time and cost, providing a well-rounded comprehension of the performance dynamics of ACO and the Greedy Algorithm across a spectrum of TSP scenarios. ACO's versatility shines through in its adept navigation of diverse TSP instances, while the Greedy Algorithm's efficiency becomes apparent in optimizing more extended and intricate routes without significant drawbacks.

This comparative exploration introduces and clarifies methodologies and outcomes, contributing significantly to the discourse on optimization strategies for real-world logistical challenges. Considering the intricate complexity of global supply chains, the selection between ACO and the Greedy Algorithm is crucial for streamlined solutions for the TSP. The thesis aims to offer actionable insights for practitioners and researchers seeking optimal algorithmic approaches in the expansive field of logistics and supply chain optimization.

GEZGİN SATICI PROBLEMİNDE KARINCA KOLONİSİ OPTİMİZASYONU VE GREEDY ALGORİTMASI PERFORMANS KARŞILAŞTIRMASI

ÖZET

Gezgin Satıcı Problemi (TSP), lojistik ve tedarik zinciri yönetiminde önemli etkileri olan, kombinatoryal optimizasyonun temel zorluklarından biridir.

Bu çalışma, Gezgin Satıcı Probleminin teorik temelleri, pratik çıkarımları ve gerçek dünya uygulamalarını derinlemesine incelemektedir. Lojistik ve tedarik zinciri optimizasyonunun önemli bir parçası olan TSP probleminin çözümü için optimal algoritmik yaklaşımlar arayan uygulayıcılar ve araştırmacılar için değerli bilgiler sunmaktadır. Bu sebeple bu çalışmada Ant Colony Optimization (ACO) ve Greedy Algoritma olmak üzere iki ana optimizasyon stratejisi karşılaştırmalı analiz şeklinde sunulmaktadır.

Matematik ve bilgisayar bilimleri alanında kombinatoryal optimizasyon, TSP modeli gibi dinamik sistemlerdeki ilişkilerin kavranmasında önemli bir etki sağlar. TSP, bir dizi şehri birer kez ziyaret ederek başlangıç noktasına geri dönüş için en iyi rotayı belirlemeyi içerir. Bu çalışma, TSP'yi daha geniş bir ağ teorisi çerçevesinde ele alarak, karmaşık sistemler içindeki ilişkileri anlamadaki temel öneminin altını çizmektedir.

Temel bir teorik dayanak olan ağ teorisi, modern tedarik zinciri yönetiminde giderek daha fazla önem kazanmaktadır. Küresel tedarik zincirleri daha karmaşık hale geldikçe, verimli rota planlaması operasyonel başarı için önemli bir etmen haline gelmektedir. Bu çalışma, ağ teorisinin TSP'nin karmaşıklıklarıyla başa çıkmak için teorik bir temel sağladığını ve yenilikçi optimizasyon stratejilerinin önünü açtığını savunmaktadır.

Bu çalışmada ele alınan literatür taraması, TSP'nin tarihsel gelişimini ve teorik temellerini aydınlatarak zemin hazırlamaktadır. Literatürde TSP problemi, simetrik ve asimetrik varyantlar olarak sınıflandırılmış ve bu araştırmada iki algoritmanın karşılaştırılabilmesi için, simetrik TSP'ye odaklanılmıştır.

Literatürde teorik değerlendirmeler, düğüm sayısı arttıkça çözüm olasılıklarındaki üstel büyümenin altını çizmekte ve TSP'nin NP-zor bir problem olarak sınıflandırılmasına yol açmaktadır. ACO, Genetik Optimizasyon, Parçacık Çekirdeği, Kesirli Denge ve Dinamik Programlama gibi sezgisel algoritmalar, NP zor problemleri çözmenin pratik yolları olarak tanıtılmaktadır. Bu sebeple bu çalışmada, büyük ölçekli örnekler için doğrusal matematiksel yöntemlerin pratik olmadığı göz önüne alındığında, sezgisel algoritmaların hızını, esnekliğini ve önemini vurgulamaktadır.

ACO ve Greedy Algoritmasının deneysel olarak incelenmesi için sağlam bir metodolojik çerçeve çok önemlidir. Bu çalışmada, çok yönlülüğü ve analitik yetenekleriyle bilinen ve geniş çapta kullanılan bir programlama ortamı olan Matlab kullanılarak titiz bir uygulama ve değerlendirme süreci tercih edilmiştir. Matlab,

algoritmaların performansının kesin bir şekilde değerlendirilmesini kolaylaştırarak standartlaştırılmış ve titiz bir karşılaştırma sağlamaktadır.

Bu tez çalışması, kapsamını teorik tartışmaların ötesine taşıyarak çeşitli senaryolarda algoritmaların verimliliğini değerlendirmek için pratik deneylere yer vermektedir. Matlab'ın hesaplama becerisinden yararlanan çalışma, zaman ve maliyet gibi önemli ölçütleri analiz ederek bu algoritmaların farklı koşullardaki performansına ilişkin kapsamlı bir içgörü sağlamaktadır.

Hem ACO hem de Greedy Algoritmanın temel ilkeleri, mekanizmalarına daha derinlemesine bir perspektif sunmak amacıyla dikkatlice incelenmiştir. Karıncaların yiyecek arama davranışından ilham alan ACO, rotaları optimize etmek için feromon izlerini kullanır. Algoritma, ziyaret edilen düğümler, şehirler arasındaki mesafeler ve feromon seviyeleri gibi faktörleri bir araya getirerek karıncaların bilinçli kararlar almasını sağlar ve algoritmanın genel verimliliğine katkıda bulunur. Öte yandan, basitliği ile tanınan Greedy Algoritması, her adımda en yakın komşuyu seçerek yerel optimizasyona odaklanır. ACO'nun uyarlanabilirliği ve zaman içinde optimize etme kabiliyeti ile Greedy Algoritmasının yerel düzeydeki basitliği ve verimliliği arasında karşılaştırmalı bir analiz, bu algoritmaların farklı avantajlarını ortaya koyar.

Ağ teorisi çerçevesinde, düğümlerin önemi vurgulanır. Bu çalışma, ağ teorisinin pratik uygulamalarını, özellikle en kısa yol (SP) problemi ve Minimum Yayılan Ağaçlar (MST) üzerinden araştırarak inceler. SP problemleri, navigasyon sistemlerinde önemli bir rol oynar. MST, altyapı projelerinde maliyetleri azaltmayı amaçlar.

Bu ağ teorisi çözümleri, gerçek dünya senaryolarının optimize edilmesinde somut uygulamalar bulur. SP problemleri, mesafe veya yakıt tüketimi gibi faktörlere bağlı olarak verimliliği artıran uyarlanabilir çözümlerle günlük hayatı etkiler. MST'nin altyapı projelerindeki rolü, operasyonel mükemmelliğin hedefine uygun olarak maliyetleri en aza indirir.

Lojistikte ağ teorisi çözümlerinin ekonomik önemi vurgulanmaktadır. Bu ilkelerden esinlenen verimli rota planlaması, maliyetleri azaltır ve operasyonel etkinliği artırır. Çalışma, ağ teorisinin teorik temellerinin ve pratik sonuçlarının anlaşılmasına katkıda bulunur, sistemlerin optimizasyonunda ve lojistik zorlukların ele alınmasındaki rolünü ortaya koyar.

Gezgin Satıcı Problemi'nin inceliklerini araştıran çalışma, 1930'larda ortaya çıkan bu problemi ağ teorisi içinde önemli bir zorluk olarak sunmaktadır. Bu problem, sabit bir düğümden başlayarak ağdaki tüm düğümleri dolaşan ve başlangıç noktasına geri dönen en verimli rotayı bulma etrafında dönmektedir. Algoritma biliminde "Big O Notasyonu "nun kullanılmaya başlanması TSP'nin çözümünü O(N!) tipi bir problem olarak kategorize etmektedir.

Çalışma içerisinde artan N değerleri ile çözüm olasılıklarındaki üstel artışı canlı bir şekilde gösterilmekte ve TSP'nin doğasında var olan önemli karmaşıklığın altı çizilmektedir. Düğüm sayısı arttıkça artan hesaplama karmaşıklıklarını vurgulayarak çeşitli "O Notasyonu" çözümlerini sistematik olarak karşılaştırmaktadır. Kapsamlı TSP örneklerini çözmek için doğrusal matematiksel yöntemler kullanmanın pratik olmadığı vurgulanarak, bu NP-zor problemleri ele almak için sezgisel algoritmaların benimsenmesi gerekliliği ortaya konmuştur.

Çeşitli algoritmik yaklaşımları inceleyen çalışma, NP-zor problemlerin çözümünde sezgisel yöntemlerin etkinliğini vurgular. ACO ve Greedy algoritmalara genel bir bakış sunar. Bu algoritmaların sezgisel doğasını vurgular, optimuma yakın çözümler üretme yeteneklerini vurgular ve teorik kavramlar ile pratik uygulamalar arasında bir bağlantı kurar.

Bu algoritmaların sezgisel doğasını vurgulayan çalışma, makul bir zaman dilimi içinde optimuma yakın çözümler üretme yeteneklerinin altını çizmektedir. TSP gibi karmaşık optimizasyon problemlerinin ele alınmasında sezgisel yaklaşımların uyarlanabilirliğini göstererek teorik kavramlar ve pratik uygulamalar arasında bir bağlantı kurmaktadır.

ACO ve Greedy Algoritmanın matematiksel formülasyonları sunulmakta ve teorik temelleri açıklanmaktadır. ACO, karıncaların karar verme sürecine rehberlik eden feromon seviyeleri ile olasılıksal bir yaklaşım kullanır. Matematiksel ifadeler, feromon izlerinin yol seçimini nasıl etkilediğini ve zaman içinde rota optimizasyonunu nasıl teşvik ettiğini göstermektedir.

Buna karşılık, Greedy Algoritmasının matematiksel formülasyonu, basitliğini ve yerel optimizasyon stratejisini vurgular. Algoritma her adımda en yakın komşuyu seçerek kademeli olarak bir çözüm oluşturur. Her iki formülasyonun matematiksel netliği, algoritmaların davranışlarını ve performanslarını anlamak için bir temel sağlar.

Çalışmada ACO ve Greedy Algoritmasının uygulanması ve değerlendirilmesi için programlama ortamı olarak Matlab kullanılmıştır. Matlab'ın seçimi çok yönlülüğü, kullanıcı dostu arayüzü ve zengin analiz araçlarına dayanmaktadır.

Matlab'ın karmaşık matematiksel işlemleri gerçekleştirmedeki verimliliğinin altı çizilerek, optimizasyon problemleri için algoritmik çözümlerin yürütülmesindeki rolü vurgulanmaktadır. Çalışma, seçilen programlama platformunun şeffaflığını ve güvenilirliğini sergileyerek tekrarlanabilirliğin önemini vurgulamaktadır. Algoritmik uygulama dünyasına pratik bir bakış sunmakta ve seçilen programlama ortamının sağlam ve tekrarlanabilir sonuçlar sağlamadaki önemini vurgulamaktadır.

Araştırma, çeşitli TSP senaryolarında ACO ve Greedy Algoritmasının performans dinamiklerine ışık tutan karşılaştırmalı analizin sonuçlarını sunmaktadır. Zaman ve maliyet dahil olmak üzere temel ölçütler titizlikle değerlendirilerek algoritmaların güçlü yönleri ve sınırlamaları hakkında incelikli bir anlayış sağlanmaktadır.

ACO, çok yönlü bir algoritma olarak ortaya çıkmakta, çeşitli TSP örneklerinde gezinme ve daha optimum rotalar elde etme konusunda uyarlanabilirlik sergilemektedir. Algoritmanın zaman içinde evrilme ve optimize etme kabiliyeti, karmaşık lojistik ve tedarik zinciri senaryolarını ele almadaki etkinliğine katkıda bulunmaktadır. Çalışmada ayrıca optimizasyon algoritmalarını seçerken hesaplama kaynaklarını ve problem gereksinimlerini göz önünde bulundurmanın önemi de vurgulanmaktadır.

Buna karşılık, Greedy Algoritması zaman verimliliği açısından kayda değer avantajlar göstermekte ve hızlı karar vermenin büyük bir öneme sahip olduğu olduğu senaryolar için pratik bir seçim haline gelmektedir. Çalışma, ACO gibi karmaşık algoritmaların her zaman daha üstün olduğu varsayımına meydan okuyarak, her algoritmanın belirli bağlamlardaki nüanslı faydalarını vurgulamaktadır.

Hem algoritmik gelişmişliği hem de pratik kısıtlamaları göz önünde bulundurarak dengeli bir yaklaşıma duyulan ihtiyacı vurgulamaktadır. Algoritmik seçimleri belirli problem örnekleriyle uyumlu hale getirmenin öneminin altını çizerek, herkese uyan tek bir zihniyetin optimizasyon alanında nasıl optimal olmayabileceğini göstermektedir.

Sonuç olarak bu tez çalışması, lojistik ve tedarik zinciri yönetimi kapsamında TSP'nin çözümü için ACO ve Greedy Algoritmanın kapsamlı bir karşılaştırmalı analizini sunmaktadır. Araştırma, kombinatoryal optimizasyonun karmaşık ortamında gezinerek, uygulayıcılara ve araştırmacılara belirli kısıtlamalara ve hedeflere göre uyarlanmış bilinçli kararlar vermeleri için uygulanabilir bilgiler sunmaktadır.

Çalışmanın gerçek dünya uygulamalarına, teorik temellere ve pratik çıkarımlara odaklanması ile hem literature hem de uygulayıcılara katkı sağlaması beklenmektedir. Bu tez çalışması, ACO ve Greedy Algoritmasının güçlü yönlerini ve sınırlamalarını vurgulayarak, karar vericilerin lojistik ve tedarik zinciri senaryolarının benzersiz gereksinimlerine göre en uygun yaklaşımları seçmelerini sağlayabilecektir. Aynı zamanda, ağ teorisi unsurlarını, algoritmik yaklaşımları ve pratik hususları bir araya getirerek TSP'nin bütüncül bir şekilde anlaşılmasını teşvik etmektedir. Lojistik ve tedarik zinciri zorlukları gelişmeye devam ederken, bu tez çalışmasının bulguları gelecekteki araştırma çabaları ve modern iş operasyonlarının dinamik ortamında yenilikçi optimizasyon stratejilerinin geliştirilmesi için bir temel oluşturmaktadır.

Teorik temellerin, algoritmik inceliklerin ve ampirik değerlendirmelerin kapsamlı bir şekilde araştırılması, TSP için optimizasyon stratejilerinin incelikli bir şekilde anlaşılmasına katkıda bulunmaktadır. Çalışmanın bilimsel titizliğe bağlılığı, pratik içgörüleriyle birleştiğinde,hem akademi hem de endüstri için değerli bir kaynak olması beklenmekte ve kombinatoryal optimizasyon ve tedarik zinciri yönetimi alanındaki problemlere uygulanmasını teşvik etmektedir.

1. INTRODUCTION

In the complex realm of combinatorial optimization, the Traveling Salesman Problem (TSP) stands as a pivotal challenge, its ramifications extending beyond theoretical intricacies to the practical landscapes of logistics and supply chain management (Ye, 2014). At the heart of this challenge lies the profound importance of network theory, a foundational pillar shaping our understanding of intricate connections within dynamic systems.

At the crossroads of mathematics, computer science, and operations research, solving the TSP is not merely an academic pursuit; it holds profound significance in real-world applications. Network theory, a cornerstone of modern logistics and supply chain management, provides a theoretical framework for understanding the intricate relationships and connections inherent in complex systems. As businesses strive for operational excellence, network theory proves indispensable, with the TSP serving as a litmus test for algorithmic efficacy in optimizing routes and minimizing resource utilization.

In the complex realm of combinatorial optimization, the Traveling Salesman Problem (TSP) emerges as an overarching challenge, transcending theoretical intricacies to permeate the practical landscapes of logistics and supply chain management. The profundity of this challenge underscores the pivotal role played by network theory, a foundational pillar that not only delineates the theoretical underpinnings but also shapes our comprehensive understanding of intricate connections within dynamic systems.

Situated at the confluence of mathematics, computer science, and operations research, the resolution of the TSP surpasses the realms of mere academic pursuit; its ramifications reverberate significantly in real-world applications. Network theory, constituting a cornerstone of modern logistics and supply chain management, bestows upon us a theoretical framework essential for comprehending the intricate relationships and connections inherent in complex systems. This theoretical foundation, rooted in rigorous academic principles, is not only a scholarly pursuit but a pragmatic necessity as businesses ardently strive for operational excellence.

In the exigent landscape of modern commerce, network theory unfolds as an indispensable tool, with the TSP serving as a litmus test for algorithmic efficacy in the optimization of routes and the judicious minimization of resource utilization. This academic exploration, steeped in theoretical rigor, accentuates the profound significance of network theory in shaping operational strategies that are not only theoretically grounded but also pragmatically efficacious in addressing the complex challenges posed by the TSP. In navigating this intricate intersection of theoretical depth and practical application, the study contributes to the nuanced understanding of algorithmic solutions within the dynamic and multifaceted domains of logistics and supply chain management.

In recent years, there has been a notable upswing in scholarly focus directed toward network theory, underscoring its pivotal and strategic significance in informing decision-making processes within the scope of supply chain management (Melo, 2008). As we immerse ourselves in the intricacies of the TSP within the contextual framework of network theory, this article undertakes a scholarly endeavor through a comparative analysis of two prominent optimization algorithms—Ant Colony Optimization (ACO) and the Greedy Algorithm. Acknowledged as stalwarts in the optimization toolkit, these algorithms proffer distinctive approaches to addressing and navigating the challenges inherent in the TSP.

In the academic exploration of these methodologies, the study aims to contribute to the evolving discourse surrounding optimization strategies, particularly within the framework of supply chain management. By critically examining and comparing the efficacies of ACO and the Greedy Algorithm, this research endeavors to elucidate nuanced insights that can inform strategic decision-making processes in the dynamic landscape of contemporary supply chains.

This scholarly pursuit is framed within the broader context of contemporary research trends, responding to the escalating interest in the intersection of network theory and optimization algorithms. By conducting a thorough and analytical investigation of these well-established optimization tools, the study seeks to enhance our understanding of their applicability and effectiveness in tackling the complex challenges presented by the TSP. In doing so, it aims to contribute substantively to the academic canon, fostering advancements in both theoretical and applied realms of optimization research within the complex milieu of supply chain management.

In the relentless quest to optimize supply chain logistics, the Traveling Salesman Problem (TSP) assumes a pivotal role, emerging as a strategic battleground. The discernment of the most efficient routes for delivery trucks or sales representatives transcends mere cost reduction; it becomes a linchpin for elevating overall operational effectiveness. The gravity of this challenge is accentuated by the ever-expanding integration of global supply chains, where adept route planning stands as the differentiator between triumph and stagnation.

In the confluence of network theory and optimization algorithms, the pressing query of which methodology—whether the nuanced Ant Colony Optimization (ACO) or the heuristic Greedy Algorithm—holds the proverbial key to unlocking the complete potential of TSP looms large. This academic exploration, situated within the dynamic landscape of contemporary supply chain management, seeks to explore the intricacies of these optimization methodologies. By scrutinizing their respective efficacies, the study aspires to contribute substantively to the scholarly discourse, offering nuanced insights that resonate with the multifaceted challenges inherent in the optimization of supply chain logistics.

The overarching significance of this inquiry is emphasized by the transformative impact that effective route planning can exert on the efficiency and competitiveness of supply chain operations. As global supply chains continue to evolve and intertwine, the strategic deployment of optimization methodologies assumes paramount importance, demanding a meticulous evaluation of ACO and the Greedy Algorithm within the context of the TSP. Through a rigorous academic lens, this investigation endeavors to unravel the intricacies, complexities, and potentialities encapsulated within these optimization tools, offering a scholarly contribution to the ongoing dialogue on supply chain logistics optimization.

In the forthcoming sections, a meticulous dissection of the foundational principles that underlie both the Ant Colony Optimization (ACO) and the Greedy Algorithm is embarked. This endeavor seeks to conduct a rigorous and scholarly comparison, unveiling the intricacies of their respective strengths and limitations. Progressing beyond theoretical considerations, our exploration will delve into practical implications through a series of meticulously designed experiments and comprehensive performance evaluations.

This academic pursuit transcends the realm of abstract theorization, aiming to bridge the gap between theoretical frameworks and tangible applications. By synthesizing critical analyses with empirical evidence, the overarching objective is to furnish practitioners and researchers within the field of supply chain management with actionable insights. These insights, derived from the nuanced comparison of ACO and the Greedy Algorithm, are intended to serve as a guiding beacon, facilitating informed decision-making processes regarding algorithmic approaches. This tailored decision-making, attuned to the specific constraints and objectives inherent in diverse supply chain contexts, stands as the hallmark of this scholarly undertaking.

In essence, this academic exploration aspires to contribute to the evolving discourse on optimization strategies within the domain of supply chain logistics. By traversing the continuum from theoretical foundations to practical implications, the study endeavors to enrich the scholarly canon, offering a nuanced perspective that aligns theoretical principles with pragmatic considerations. Through such alignment, the aim is to empower decision-makers with the knowledge requisite for judicious algorithmic selection, thereby enhancing the efficacy and adaptability of supply chain operations in the dynamic and complex landscapes of contemporary commerce.

1.1. Literature Review

The vehicle scheduling problem involves finding optimal schedules for a fleet of vehicles originating and terminating at a central depot. These vehicles have the specific capacity and travel time constraints while servicing a set of nodes. The assignment of nodes to vehicles must adhere to the constraint that each node is serviced exactly once, ensuring that each vehicle complies with the given constraints (Park, 2001). The Traveling Salesman Problem (TSP) is a key challenge in combinatorial optimization, involving finding the most efficient route to visit a set of cities exactly once and return to the starting city. The Traveling Salesman Problem (TSP) has been documented in the annals of combinatorial optimization since 1930. Merrill M. Flood initially formulated the problem mathematically when he encountered it while addressing a school bus routing dilemma (Flood, 1948). TSP comes in two forms based on node distances. Asymmetric TSP has varying distances between cities depending on the travel direction, such as navigating one-way streets or encountering different fuel costs. Symmetric TSP, in contrast, maintains uniform distances between nodes, regardless of travel direction (Rachmat & Mandala, 2022). In this study, the focus is on Symmetric TSP. The goal is to examine the impact of the chosen methodology without introducing unnecessary complexity. Symmetric TSP has been selected for its simplicity in this context.

The Ant Colony Optimization (ACO) algorithm, drawing inspiration from how ants forage for food, proves highly effective in addressing the Traveling Salesman Problem (TSP). By emulating the intelligent behavior of ants, artificial ants consider factors such as visited nodes, distances between cities, and real-time pheromone levels to make informed decisions. ACO excels in optimizing routes by leveraging these characteristics, providing a smart and efficient solution to the TSP (Donbosco & Chakraborty, 2021). The simplicity of the Greedy Algorithm has been explained by Gil-Gala, Durasević, Sierra, and Varela (2023) as follows:

When it comes to solving large instances in real time, greedy algorithms guided by priority rules represent the most common approach, being the nearest neighbor (NN) heuristic as one of the most popular rules. NN is quite general but it is too simple and so it may not be the best choice in some cases. Alternatively, we may design more sophisticated heuristics considering the particular features of families of instances (Gil-Gala, Durasević, Sierra, & Varela, 2023).

It's noted that simple algorithms, like the nearest neighbor (NN) heuristic, may not always produce the optimal results for large instances in real-time scenarios. In contrast, more sophisticated heuristic algorithms, designed to consider specific features of instance families, have the potential to yield better results.

Nevertheless, it's crucial to acknowledge that complex algorithms typically demand increased time, effort, and advanced technical resources. Achieving superior results, characterized by a shorter and more cost-effective path, may entail substantial investments in both financial resources and time during the algorithm development phase. This could include allocating funds for server costs and optimizing algorithm runtime to enhance overall efficiency.

1.2. Network Theory

In the realm of network theory, a pivotal concept that demands nuanced exploration is centrality—a metric that illuminates the significance of a node within a network. This multifaceted notion is characterized by diverse centrality metrics, each providing distinct insights into the role and influence of nodes within the network architecture (Gómez, 2019). Among these metrics, degree centrality emerges as a foundational measure, assessing the number of connections attributed to a particular node. In parallel, betweenness centrality scrutinizes the node's position in facilitating the shortest paths, while eigenvector centrality evaluates the centrality of a node in consideration of the centrality of its connected neighbors (Borgatti, 2011). This intricate web of centrality metrics constitutes an indispensable facet of network theory, offering a nuanced lens through which the structural importance of nodes within a network can be comprehensively gauged and deciphered (Şeker, 2015; Bollobás, 1998).

The practical utility of network theory extends across diverse domains, serving as a versatile analytical framework with far-reaching applications. One notable application resides in the analysis of social networks, where network theory becomes instrumental in elucidating the intricate dynamics of interpersonal connections and discerning the pathways through which information propagates within these networks. Furthermore, its applicability extends to the realm of epidemiology, where network theory proves invaluable in comprehending the transmission and spread of diseases within populations (Danon, 2011). By modeling and scrutinizing the interconnections among individuals, it facilitates a nuanced understanding of the factors influencing disease propagation.

Additionally, network theory finds resonance in the analysis of the internet's structural architecture. This application involves the examination of connectivity patterns, node relationships, and information flow dynamics within the vast and complex network that constitutes the Internet. The theoretical underpinnings of network theory, when applied to internet structures, offer insights into the resilience, efficiency, and vulnerability of the digital infrastructure. Through such analyses, scholars and practitioners gain a more profound comprehension of the internet's organization and functioning, contributing to advancements in network design, cybersecurity, and overall system optimization.

In essence, the multifaceted applications of network theory, ranging from social networks to epidemiology and internet architecture, underscore its significance as a foundational framework for understanding complex systems (Danon, 2011). As scholars and researchers delve into these applications, they contribute not only to the refinement of theoretical models but also to the development of practical solutions addressing real-world challenges across diverse disciplines.

Network theory stands as a captivating and pivotal field of scholarly inquiry, marked by a plethora of practical applications that underscore its significance. Positioned at the intersection of theoretical exploration and practical utility, network theory furnishes a robust framework for comprehending complex systems by directing attention to their inherent structure and dynamics. This analytical approach enables a meticulous dissection of relationships among entities, unraveling the intricate web of connections that permeate these systems. Through this nuanced lens, network theory facilitates the discernment of emergent properties, shedding light on phenomena that manifest as a consequence of these interconnections.

The potency of network theory lies in its capacity to systematically analyze the relational intricacies within complex systems, transcending disciplinary boundaries. By affording scholars the tools to probe into the networked nature of entities, it opens the door to a more profound understanding of the synergies and dependencies that characterize diverse domains. Moreover, as technological advancements burgeon and the reservoir of available data expands, the application of network theory assumes an increasingly pivotal role. This methodological framework, fortified by the burgeoning troves of data, stands poised to yield continued and profound insights across a spectrum of academic disciplines.

In the academic milieu, the trajectory of network theory unfolds as an enduring narrative, propelled by an incessant quest to decipher the intricacies of interconnected systems. Its contribution to the elucidation of emergent properties, coupled with its adaptability to evolving technological landscapes, positions network theory as an enduring cornerstone in the pursuit of knowledge across various disciplines. As scholars delve further into the rich tapestry of complex systems, the tenets of network theory are set to play an increasingly significant role in unraveling the profound interplay of entities and relationships that define our interconnected world.

Networks are divided into different classifications due to their properties. To give an example of these; Although they are the most known types such as one-way (directed) or two-way (undirected), weighted or unweighted, there are more network types with different properties (Şeker, 2015; Bollobás, 1998). Appropriate analyses differ according to network types. To analyze any kind of network, it should be suitable to be expressed mathematically via matrices of special tables. To give an example of these matrices over a simple network shown in Figure 1.1.; edge-node matrix (Table 1.1.), neighborhood matrix (Table 1.2.), distance matrix, etc. (Table 1.3) may be representations.

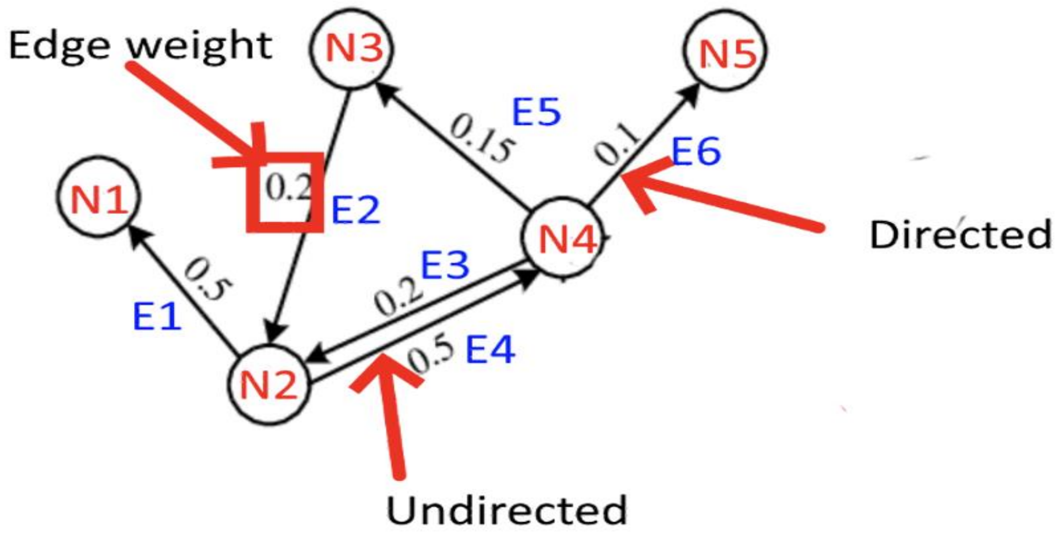

Figure 1.1. Representation of different parametric properties of a simple network, visualization of a directional and weighted network.

Table 1.1. Edge-node Matrix of Figure 1.

	E1	E2	E3	E4	E5	E6
N1	1	0	0	0	0	0
N2	1	1	1	1	0	0
N3	0	1	0	0	1	0
N4	0	0	1	1	1	1
N5	0	0	0	0	0	1

Table 1.2. The Neighborhood Matrix of Figure 1.

	N1	N2	N3	N4	N5
N1	0	1	0	0	0
N2	1	0	1	1	0
N3	0	1	0	1	0
N4	0	1	1	0	1
N5	0	0	0	1	0

Table 1.3. Distance Matrix of Figure 1.

	N1	N2	N3	N4	N5
N1	0	0	0	0	0
N2	0,5	0	0	1	0
N3	0	0,2	0	0	0
N4	0	0,2	0,15	0	0
N5	0	0	0	0	0

Solutions using network theory have uses in daily and business life. One of the usages is the shortest path (SP) problem between two locations, which is used in navigation systems in daily life. The example is shown in Figure 1.3. shows the shortest path over the network in Figure 1.2. The shortest path method is mainly used in networks. Different SPs can be obtained by using more than one weight parameter of the edges (distance, fuel consumption per km, etc.). For example, calculating the SP route to the destination in the shortest time, considering the SP and traffic density values as distance. As another important network theory solution, Minimum scan trees (MST) can be used for electricity, telephone, etc. pole placement. The objective here is to ascertain the minimum number of edge lines that can connect with all nodes in a network cluster, as shown in Figure 1.4. It has great importance in minimizing project costs by using it actively in infrastructure works. The traveling salesman problem, which is also a focal point in this study, is a problem that should start from a starting node and traverse all the nodes in the network and return to the initial node as shown in Figure 1.5. The aim here is that this determined route should have a minimum path length. As an example of the economic contribution of this, it enables more work to be done in unit time by saving both fuel and time with less travel cost in any cargo distribution business (Şeker, 2015; Jünger, Reinelt, & Rinaldi, 1995; Bollobás, 1998; Bellmore & Nemhauser, 1968).

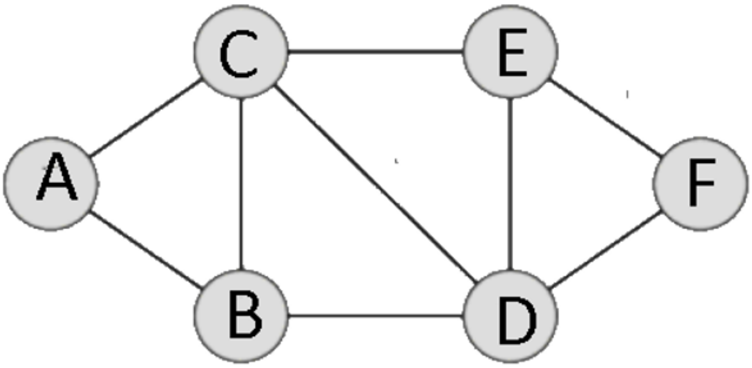

Figure.1.2. Example Network

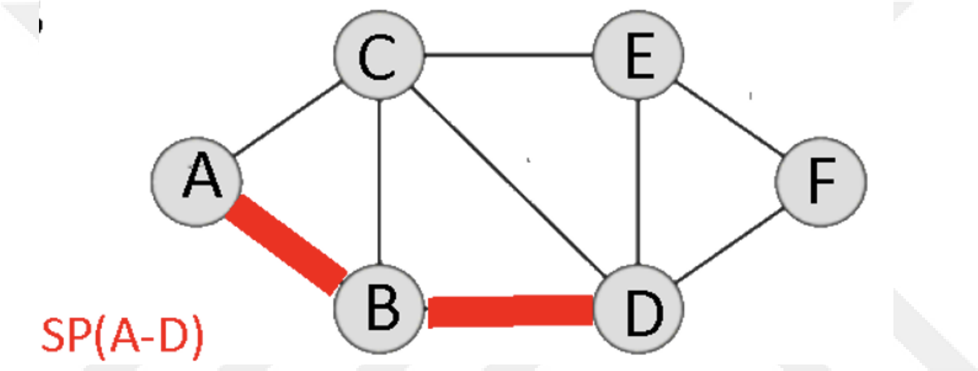

Figure 1.3. Shortest path.

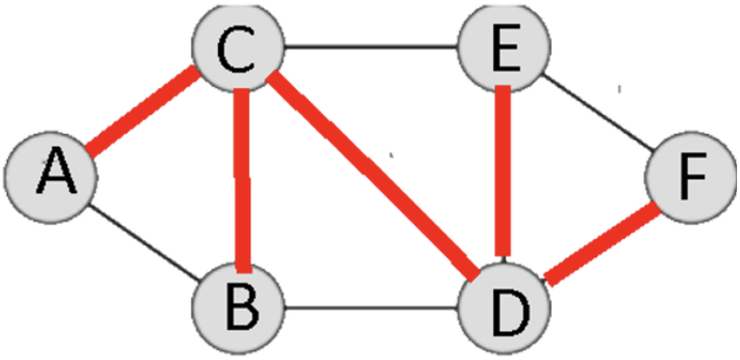

Figure 1.4. Minimum spanning tree.

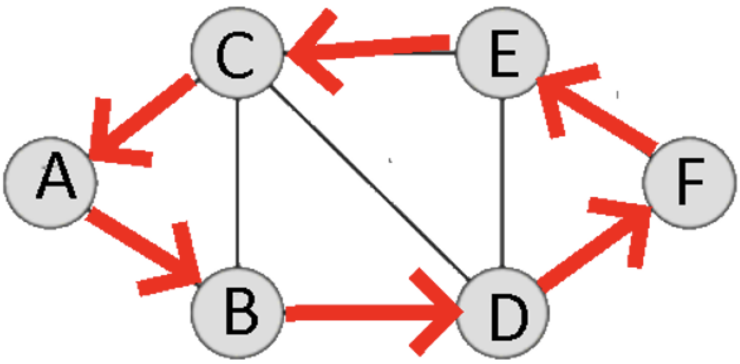

Figure 1.5. Traveling Salesman Problem

1.3. The Travelling Salesman Problem

This problem holds paramount significance in network theories (also mathematical problems in scope) problems formulated in the 1930s. Being one of the most renowned usage areas, it can be defined as starting at a certain start point and reaching the starting point again by passing through all the necessary nodes in sectors such as cargo transportation. As the simplest solution (Davis, 2010; Bellmore & Nemhauser, 1968; Jünger, Reinelt, & Rinaldi, 1995);

• Determination of starting point from N nodes. If this point is the cargo center, which is fixed for each solution, the problem will be formulated for the N-1 location.

• There is a choice for one of the N-1 positions after the starting position.

• He then has a choice for one of the N-2 positions.

• …

• It continues with 1 choice for the last point.

When considering a fixed initial node in the context of the discussed problem, a combinatorial explosion of solutions emerges, yielding (n-1)! distinct possibilities. However, a notable symmetry presents itself when traversing in the reverse direction from the established starting point, resulting in solutions of equivalent value. This symmetry leads to a reduction in the total count, resulting in (n-1)!/2 unique solutions. The computational complexity of this scenario manifests prominently, exemplified by a factorial growth rate, with a complexity of 12 for instances with 5 values of N, escalating to 181,440 for N equal to 10, and further surging to an astronomical 43,589,145,600 for the case of 15 values of N.

This exponential growth underscores the considerable challenges presented by the sheer combinatorial explosion inherent in the problem domain. The factorial growth rate, a testament to the increasing intricacy as the number of values of N escalates, accentuates the computational demands associated with exploring the solution space. Such complexities underscore the need for efficient algorithmic approaches, particularly within optimization contexts, where exhaustive enumeration becomes impractical. As the intricacies of the problem domain unfold and expand, the demand for sophisticated methodologies that navigate this labyrinth of possibilities becomes increasingly pronounced. In navigating these complexities, the academic exploration of algorithmic solutions assumes paramount importance, seeking avenues to tackle the significant challenges presented by the vast solution space. According to the concept expressed as “Big O Notation” in algorithm science, the rough solution of the TSP problem expresses an O(N!) type of problem. Figure 6 shows that the intricacy of the solution approach escalates as the N value increases compared to other types. Figure 1.7 shows the complexity level of these different "O Notation" solutions (Dean-Leon, Nair, & Knoll, 2012) depending on the N value (Chivers & Sleightholme, 2015; Rubinstein-Salzedo, 2018; Jünger, Reinelt, & Rinaldi, 1995; Bellmore & Nemhauser, 1968). In Figure 1.6., a representation of the time complexity of different kinds of algorithmic solutions according to Big O Notation has been shown.

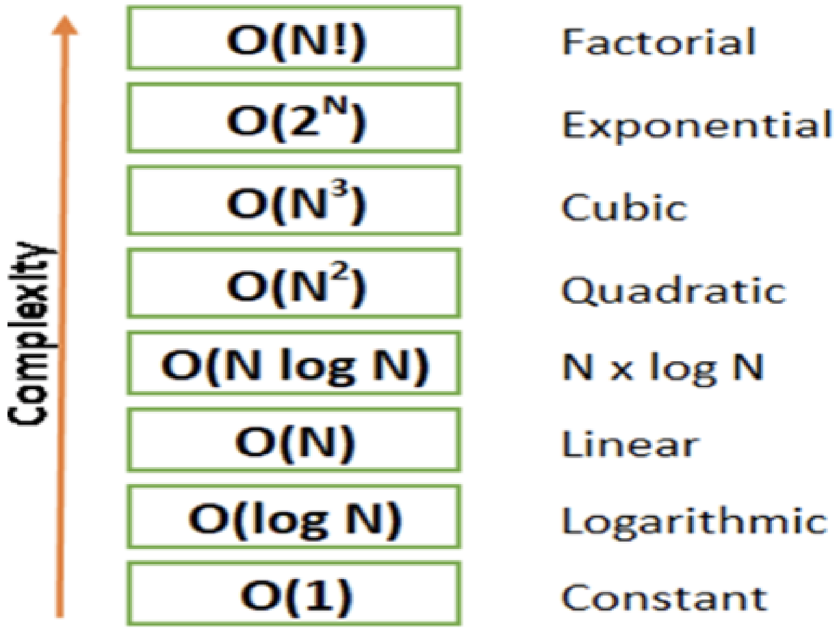

Figure 1.6. Display of complexity level.

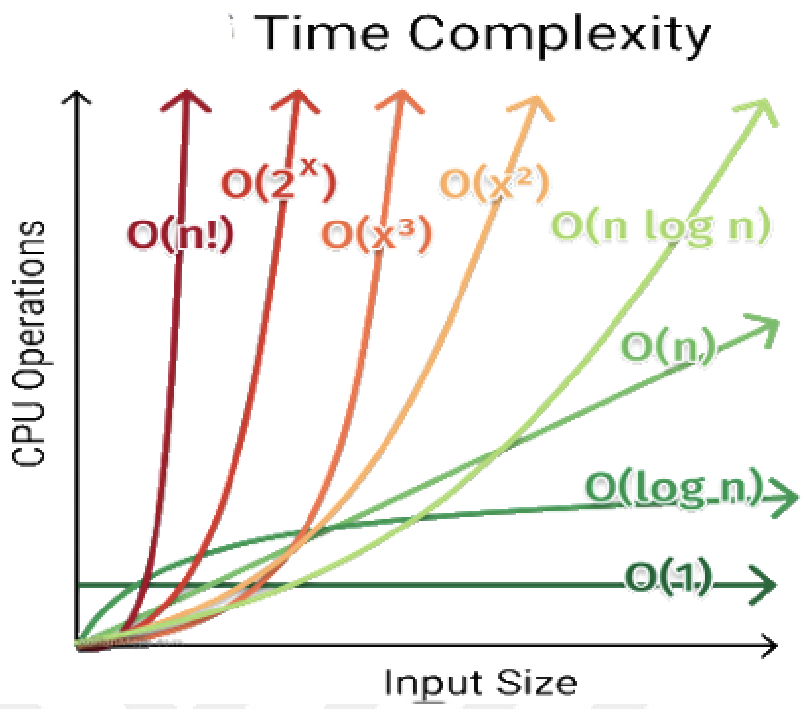

Figure 1.7. The time complexity of O Notation solutions.

In the domain of the Traveling Salesman Problem (TSP) and its variants, a pragmatic approach often favors algorithms that exhibit expeditious computational efficiency while yielding solutions closely approximating the optimal. This preference arises from a practical acknowledgment of the escalating computational intricacies accompanying the growth of the problem size, denoted by the variable N. The burgeoning complexity of TSP instances, particularly as N attains larger values, underscores a critical operational reality—the imperative to balance computational expediency with solution quality.

The rationale behind prioritizing algorithms that optimize for computational speed over attaining an absolute optimum resides in the inherent difficulties presented by the expanding problem dimensions. With each incremental increase in the N value, the combinatorial explosion of potential solutions exponentially augments the computational demands (Li W. , 2022). This surge in complexity not only affects the runtime duration but also amplifies the requisite computational resources—measured in terms of processing power and RAM—rendering the pursuit of the absolute optimal solution a computationally prohibitive endeavor.

In this academic discourse, we delve into the nuanced considerations surrounding algorithmic choices in TSP-type problems, emphasizing the pragmatic necessity of favoring computational efficiency without sacrificing the precision of the obtained solutions. By navigating the delicate balance between time complexity and solution quality, researchers and practitioners can formulate algorithmic strategies that are not only operationally feasible but also attuned to the practical constraints imposed by the increasing complexities inherent in larger-scale TSP instances. Furthermore, the inherent dynamism observed in sectors such as cargo distribution introduces an extra layer of intricacy to optimization problems, specifically within the context of the Traveling Salesman Problem (TSP). Unlike static scenarios where the number of location points, denoted as N, and the distances between these points remain constant, real-world applications frequently exhibit variability. In instances of cargo distribution, for instance, the quantity of cargo deliveries may fluctuate, ranging from 50 to 60 deliveries on different days, introducing an element of variability in the problem dimensions. Additionally, the spatial relationships between locations are subject to change, exemplified by variations in road distances arising from factors such as ongoing road maintenance.

The dynamic nature inherent in parameters central to the Traveling Salesman Problem (TSP) introduces a compelling imperative: the perpetual reevaluation of the optimal distribution route for each operational cycle. This dynamism renders the TSP, in the context of cargo distribution, an inherently dynamic problem, diverging markedly from static instances characterized by fixed parameters. Unlike scenarios where location points (expressed as N) and distances remain constant, the real-world landscape of cargo distribution is rife with variability, manifested through fluctuating cargo quantities—oscillating between, for instance, 50 and 60 deliveries on distinct days—and mutable spatial relationships arising from factors like road maintenance-induced alterations in standard road distances.

This fluidity in TSP parameters engenders an environment of perpetual uncertainty and variability, compelling a departure from conventional optimization approaches designed for static problem formulations. The exigencies of dynamic cargo distribution scenarios demand the deployment of adaptive optimization strategies capable of promptly recalibrating distribution routes in reaction to changing circumstances. In this scholarly exploration, we commence a thorough analysis of the intricate complexities stemming from the variable nature of location points and distances in practical TSP applications. This academic discourse underscores the pressing need for algorithmic frameworks adept at accommodating such fluctuations, thereby dynamically adapting distribution routes to align with the ever-changing conditions characteristic of real-world cargo distribution scenarios.

The intrinsic complexity inherent in the Traveling Salesman Problem (TSP) precludes the efficacy of linear mathematical methods as optimal solutions for its resolution. The intricacies embedded in TSP, marked by the exponential expansion of potential solutions with each increment in the problem's size, necessitate a departure from linear methodologies. Linear approaches, which presuppose a constant relationship between variables, confront substantial challenges in accommodating the dynamic and changing characteristics of TSP scenarios.

In the context of TSP, the inadequacy of linear mathematical methods is further underscored by the imperative for iterative modifications to the problem model. The dynamic nature of scenarios such as cargo distribution, where factors like varying delivery quantities and shifting distances between locations introduce continual fluctuations, mandates a persistent revision of the problem model. This iterative refinement process not only amplifies the intricacy of the TSP but also exacerbates the challenges faced by linear methods in maintaining relevance throughout the evolving problem landscape.

In this academic discourse, we delve into the limitations posed by the TSP's complexity on the applicability of linear mathematical methods. Emphasizing the need for adaptive and non-linear approaches, this exploration underscores the infeasibility of relying solely on linear methodologies, particularly in the face of the dynamic and mutable nature of TSP scenarios. In the literature, heuristic algorithmic solutions related to TSP are used for NP-hard problems. Some of these are algorithms such as heuristic Genetic Algorithm (GA), Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO), Greedy algorithms, and Dynamic Programming (Sathya & Muthukumaravel, 2015; Chauhan, Gupta, & Pathak, 2012).

1.4. Problem Definition

The conjecture propounded in this scholarly investigation posits that, within the context of daily Traveling Salesman Problem (TSP) instances encountered in routine logistical operations, the Greedy Algorithm may present itself as a sufficiently efficacious solution, obviating the imperative for the application of Ant Colony Optimization (ACO). The anticipation rests on the Greedy Algorithm's recognized attributes of simplicity and expeditiousness, characterized by its adeptness in selecting the nearest neighbor at each computational step. It is envisaged that the Greedy Algorithm will demonstrate notable efficacy in route optimization, particularly when applied to TSP scenarios characterized by brevity and simplicity, which are frequently encountered in the day-to-day logistical milieu.

The underlying hypothesis contends that the streamlined and straightforward nature of the Greedy Algorithm may obviate the need for the additional computational complexity associated with ACO, particularly in scenarios typified by routine TSP challenges. This study endeavors to empirically substantiate this hypothesis through a meticulous and thorough comparative analysis of the effectiveness of ACO and the Greedy Algorithm. The principal aim of this study is to furnish nuanced insights into the optimal algorithmic approach for addressing daily instances of the Traveling Salesman Problem (TSP) within the distinct realm of logistics and supply chain management. This contribution aspires to enhance the ongoing discourse on algorithmic selection in practical operational contexts. In the second part, materials and methods are explained. In the third part, the formulation of the problem and function sets are explained. The forth part explains the analysis and results.

2. MATERIALS AND METHOD

2.1. Matlab

MATLAB, an acronym for MATrix LAboratory, was devised by Cleve Moler in the 1960s and has since evolved into a robust computational system (Wallisch, 2014). As of 2020, with a user base of approximately 4 million legal users globally, MATLAB stands as a pervasive and actively utilized platform across various sectors. Its prevalence extends to crucial domains such as engineering, science, and economics, indicative of its indispensability in facilitating diverse analytical tasks.

The widespread adoption of MATLAB in these fields can be attributed to its comprehensive suite of analysis tools, encompassing applications in image processing, bioinformatics, artificial intelligence, statistics-data analysis, and simulations. This versatility positions MATLAB as a versatile and all-encompassing tool for researchers and practitioners across a spectrum of scientific and engineering disciplines. In this field, MATLAB has been used with license number 41016490, 2023.

Another pivotal factor contributing to MATLAB's prominence is its user-friendly interface, coupled with an extensive repository of resources to support its user community. This feature significantly reduces the learning curve for new users, fostering a conducive environment for rapid skill development without undue concern about resource availability. In essence, MATLAB's accessibility and rich content empower users to advance in their work seamlessly, establishing it as an instrumental platform for scientific and computational endeavors (Davis, 2010; Dean-Leon, Nair, & Knoll, 2012).

2.2. Ant Colony Optimization

Ant Colony Optimization (ACO) emerges as a heuristic problem-solving algorithm, drawing inspiration from the foraging behavior of ants in their quest for resources. This algorithm is rooted in the sophisticated communication and organizational strategies employed by ant colonies, where chemical molecules known as pheromones play a pivotal role (Dorigo M. B., 2006).

In the intricate world of ant colonies, these tiny creatures utilize pheromones as a means of communication and coordination. Pheromones serve as chemical markers, allowing ants to convey information and orchestrate collective activities, particularly in endeavors such as foraging for food. The foraging pathways, akin to edges in a graph representation, are intricately mapped and navigated through the deposition and detection of pheromones (Jaiswal, 2011).

In this context, each edge within the algorithmic framework corresponds to a distinct pathway, mirroring the trails of pheromones left by ants during their exploration. The utilization of pheromone-inspired mechanisms in ACO encapsulates the essence of decentralized coordination and information sharing observed in ant colonies, offering a bio-inspired approach to solving complex problems through collective and adaptive behaviors.

In the intricate dynamics of the Ant Colony Optimization (ACO) method, the utilization of marked paths becomes a fundamental aspect of the algorithmic process. As ants traverse these designated routes, they release pheromones, effectively marking the paths and communicating their viability to other ants within the colony. This communal signaling mechanism contributes to the preference for specific routes, as the presence of pheromones renders those pathways more attractive to subsequent ants.

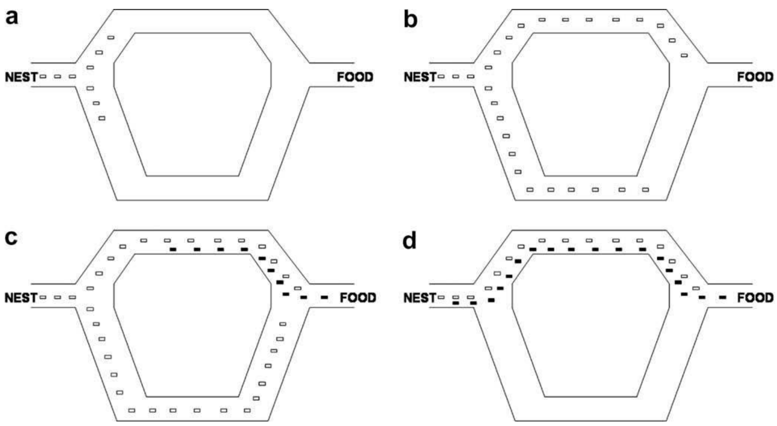

Figure 2.1. An illustration of the shortest-path experiment

The illustration shown in Figure 2.1 (Cordon, 2002) depicts two routes connecting the nest and the food source. The upper path is shorter, while the lower path is longer

(Çalişkan, 2011). In the experiment, when ants were released to forage for food, half of them favored the shorter upper path, while the other half preferred the longer lower path. Since the upper path is shorter, ants reach the food source more quickly. Upon reaching the food source, ants return to the nest by depositing pheromones along the path, guiding others to follow these pheromone trails to the food.

As a result, a higher concentration of pheromones accumulates on the upper path, increasing the likelihood of ants choosing this route. Ultimately, this process leads to a convergence towards the upper path as the preferred route due to its shorter distance.

The algorithm incorporates a temporal dimension through the natural evaporation of pheromones over time. As marked paths are traversed repeatedly, the amount of pheromone gradually diminishes through evaporation (Mavrovouniotis, 2014). This temporal decay mechanism serves to dynamically regulate the attractiveness of paths, fostering adaptability within the ant colony. Consequently, the sustained release of pheromones on common routes enhances their desirability among different ants, contributing to the emergence of optimal solutions.

Furthermore, the ACO method introduces parameters that are pivotal in shaping the algorithm's effectiveness. These parameters encompass the quantity of pheromone release, the coefficient dictating the rate of pheromone decrease over time, the number of participating ants, and the frequency of algorithmic repetitions. The strategic manipulation of these parameters allows for the fine-tuning of the ACO method, enabling the identification of optimal solutions through the collaborative and dynamic exploration facilitated by the exchange of pheromones. In essence, the ACO method harnesses the principles of decentralized cooperation and adaptive behavior observed in ant colonies, offering a bio-inspired paradigm for solving complex problems. The ACO method used for TSP proceeds as outlined in the steps below.

1. X number of ants are randomly placed in a network of N nodes

2. Each ant creates a route by wandering all nodes according to the determined alpha and beta values.

3. The route value of each ant is calculated.

4. According to the route distances of the ants, the pheromone values on each edge are increased.

5. According to the pheromone evaporation value, the pheromone values on each edge are decreased.

6. Repeat steps 2-5 until the initial set number of repetitions.

The ACO method is population-based and tends to find more optimal results with replications. Depending on these repetitions, more accurate routes are determined, and the total processing time increases. In addition, parameter selection is critical in such methods. For example, a high or low evaporation coefficient makes it difficult to determine shortcuts in route selection (Blum, 2005; Dorigo & Blum, 2005; Bellmore & Nemhauser, 1968).

The symbol N_{ij}, which expresses the probability value, is inversely proportional to the distance between node i and node j shown in equation 2.1 (Dorigo M. M., 1991).

$$N_{ij} = \frac{1}{d_{ij}} \tag{2.1}$$

$$\Delta\tau_{i,j}^{k} = \begin{cases} \frac{1}{L_k} & \text{The cost of the kth ant to travel to points i and j} \\ 0 & \text{Other Situations} \end{cases} \tag{2.2}$$

As shown in equation 2.2 (Dorigo M. M., 1991). $\Delta\tau$ represents the amount of pheromone each ant releases onto the road. Since the ant moves from one point to another, it represents the distance between i and j points in the model. As seen in the equation, if the ant does not move, the level of pheromone secreted is 0. $1/L_k$ represents the distance between two points. Here, 1 / the distance between two points is because the algorithm searches for the shortest path. Thanks to this formula, more pheromone levels will be secreted for a shorter route.

$$\Delta\tau_{i,j}^{k} = \sum_{k=1}^{m} \Delta\tau_{i,j}^{k} \qquad \text{without evaporation} \tag{2.3}$$

Above equation 2.3 (Dorigo M. M., 1991) m represents the total number of ants. This equation is calculated without an evaporation rate.

$$\Delta\tau_{i,j}^{k} = (1-\rho)\tau_{i,j}^{k} \sum_{k=1}^{m} \Delta\tau_{i,j}^{k} \qquad \text{with evaporation} \tag{2.4}$$

In the mathematical model above 2.4, the first part represents the current pheromone level and the second part represents the pheromone level secreted by all other ants. As seen in the model, Rho is a constant number and represents the evaporation rate.

When Rho = 0, there is no evaporation. In this case, the equation gives the same result as the equation without evaporation. If Rho = 1, all the pheromones left by the ant will evaporate.

$$P_{i,j} = \frac{\left(\tau_{i,j}\right)^{a}\left(\eta_{i,j}\right)^{\beta}}{\sum(a+b)} \quad , \quad \eta_{i,j} = \frac{1}{L_{i,j}} \tag{2.5}$$

Above equation 2.5. (Dorigo M. M., 1991) eta value (η) represents heuristic information (reverse of distance), tau (τ) represents pheromone level between two nodes, alpha (a) represents the effect of pheromone level, beta (β) represents the effect of heuristic information. The numerator of the equation calculates the probability of the specific condition to be calculated, denominator calculates all probabilities. P value represents the probability ratio.

2.3. Greedy Algorithm

Greedy Algorithms are heuristic and easy-to-implemented algorithms used in solving a particular problem (Curtis, 2003). In contrast to the conventional approach exemplified by Ant Colony Optimization (ACO), a distinctive design technique gains recognition for its simplicity of application and widespread applicability across diverse problem domains. This design methodology is characterized by its adaptability and ease of implementation, positioning it as a preferred choice in algorithmic problem-solving. Notably, its prevalence is prominently evident in the field of network theory, where it finds exemplary application through well-known algorithms such as Kruskal's and Prim's Algorithms in Minimum Spanning Tree (MST) problems, as well as Dijkstra's Algorithm in Shortest Path (SP) problems (de Aragao, 2002).

The adoption of this design technique is underscored by its inherent simplicity, making it an accessible and efficient choice for addressing complex optimization challenges. Its versatility is particularly evident in network-related problem instances, where the algorithms inspired by this design approach play a seminal role in elucidating optimal solutions. The prominence of Kruskal's and Prim's Algorithms in MST scenarios, coupled with Dijkstra's Algorithm in SP contexts, attests to the enduring relevance and effectiveness of this design technique in the realm of algorithmic optimization. Essentially, the pragmatic application of this design methodology emerges as a hallmark in algorithmic solutions, offering a pragmatic and accessible approach to problem-solving. However, the drawback of greedy algorithms lies in their tendency to prioritize short-term optimization by selecting the best available option at each step, without guaranteeing the globally optimal result. Consequently, in certain problem instances, greedy algorithms may either be unsuitable or fail to produce satisfactory outcomes (Olivos-Castillo, 2019). Within the confines of this investigation, the operational principle of the Preferred Nearest Neighbor algorithm is:

1. Consider every N node as the starting point.
2. Determine the nearest unused node to the current node.
3. Set the determined node as an active location.
4. Repeat steps 2-3 until you get back to the starting point.

According to the above steps, N different routes are generated for N nodes. The solution set has the shortest distance from these routes. Although it has a simple design, generally less processing time and complexity, less successful results are produced than most other methods. But it gives the best results in terms of “price/performance” due to its adaptability, low resource consumption, and short time resolution (Kizilateş & Nuriyeva, 2013; Abdulkarim & Alshammari, 2015).

Mathematically, this selection process is generally expressed with the following formulas:

$$x_{t+1} = arg\, min_{c \in C}\{d(s_t, c)\} \tag{2.6}$$

xt+1 represents the chosen closest city at step t+1. St denotes the state at step t. C,t is the set of available options at step t. $d(s_t, c)$ represents the objective function that needs to be optimized when transitioning from the state St to the next step with option c. This formula facilitates the determination of the best choice at each step by evaluating the current state and options on the objective function (Saraei, 2015).

2.4. Held-Karp Algorithm

The Held-Karp algorithm is a dynamic programming technique that ensures the exact optimal solution to the Traveling Salesman Problem (TSP). It achieves this by decomposing the TSP into smaller subproblems and progressively building up the solution through dynamic programming (Righini, 2021). To understand the results of ACO and Greedy and define the difference between absolute results, smaller N values are tested with Held-Karp. By doing that,

The algorithm examines all possible subsets of cities and all routes to reach each city within these subsets. This comprehensive search guarantees that no potential optimal route is missed. The TSP has an optimal substructure, meaning the best solution to the problem can be formed from the best solutions to its smaller subproblems. The Held-Karp algorithm utilizes this property by incrementally building the solution. By storing intermediate results in a table, known as memoization, the algorithm avoids redundant calculations, making it more efficient than a naive brute-force approach while still considering all possibilities. The recursive formula employed by Held-Karp accurately calculates the minimum cost for each subset of cities ending at each possible city. When these results are combined, they yield the overall minimum tour cost (Parjadis, 2023) (S. Dasgupta). The steps of the Held-Karp algorithm are shown below.

$C(S,j)$ represents the minimum cost to visit all cities in subset S and ends at city j.

For the starting city (city 0), the cost to visit only the city 0 and end in city 0 is zero:

$$C(\{0\}, 0) = 0 \tag{2.7}$$

For each subset of cities S that includes the starting city and another city j in S:

$$C(S, j) = \min_{i \in s, i \neq j} [C(S - \{j\}, i) + d(i, j)] \tag{2.8}$$

$d(i,j)$ is the distance between city i and city j. This formula finds the minimum cost to reach city j from any city i in subset S excluding j.

The tour must end in the starting city. The overall minimum tour is shown accordingly.

$$\min_{j \neq 0} [C\{O, 1,2, \dots, n-1\} + d(j, 0)] \tag{2.9}$$

2.5. Statistical Test

Statistical tests are tools used for analyzing and interpreting statistical data. They are utilized to evaluate the characteristics or relationships of one or more samples. Statistical tests are typically employed to test one or more hypotheses. A statistical hypothesis is a claim about a sample or sample distribution. By testing these hypotheses, statistical tests determine whether a sample, or samples, based on observed data, are representative of the general population (Limentani, 2005).

T-test, which is a commonly used statistical test method, is chosen to understand the results of this study. It is used to determine whether there's a significant difference between the averages of two groups. It's commonly used to compare data from two groups, assuming they follow a normal distribution. There are two main types; the independent samples t-test and the paired samples t-test. The Independent Samples T-Test checks whether the difference between the averages of two separate groups is meaningful. The Paired Samples T-Test compares samples taken from the same individuals or items under two different conditions (Kim T. K., 2015).

In this study, since the same nodes are examined in two different algorithms, paired samples T-Test has been used to test whether the difference between results is meaningful (Osaba, 2016) (Manfei, 2017).

One of the t-test outcomes is the "t statistic," which signifies the magnitude of the difference between group averages and the extent of disparity between the groups. This statistic is then assessed by computing the *p*-value within a specified confidence interval. The p-value indicates whether the observed difference is likely attributable to chance. A low *p*-value (typically below 0.05) suggests that the disparity between groups is statistically significant. The paired T-Test formula has been shown in the equation 2.10 (Kim, Park, & Wang, 2018).

$$t = \frac{\bar{d}}{\frac{s_d}{\sqrt{n}}} \tag{2.10}$$

In this equation, $\bar{d}$ is the average value of the difference between two conditions, s_d is the standard deviation of the differences, n represents the size of the sample.

3. PROBLEM FORMULATION AND FUNCTION SETS

The study addresses the Traveling Salesman Problem (TSP), a well-known challenge in combinatorial optimization. The TSP involves determining the most efficient route for a traveling salesman to visit a specified set of cities and return to the initial city, with the primary objective of minimizing either the total distance traveled, or the overall cost incurred during the journey.

The representative is required to visit each city precisely once, and the sequence of city visits is critical. The computational complexity of the TSP increases significantly with the number of cities due to its NP-hard nature. In this context, the research focuses on comparing two heuristic algorithms, Ant Colony Optimization (ACO) and the Greedy Algorithm, as potential solutions to the TSP. These algorithms are designed to provide approximate solutions for combinatorial optimization problems.

The primary objective of this research is to evaluate and compare the effectiveness of ACO and the Greedy Algorithm in solving the TSP under various conditions. Specifically, the research aims to offer insights into the strengths, weaknesses, and efficiency of these algorithms in finding near-optimal solutions for the Traveling Salesman Problem. By conducting a comprehensive analysis, the study seeks to contribute valuable knowledge to the field of optimization algorithms and aid in determining the most suitable approach for addressing the TSP in different scenarios.

Constraints in the context of the Traveling Salesman Problem (TSP) typically revolve around the limitations and conditions that must be adhered to when finding a solution. Here are some common constraints associated with the TSP:

- Visiting Each City Exactly Once: The primary constraint is that the salesman must visit each city exactly once. Revisiting a city is not allowed.
- Returning to the Initial City: The salesman must return to the initial city after visiting all other cities. The TSP involves a closed loop.
- Symmetric vs. Asymmetric TSP: In symmetric TSP, the distance between any two cities remains consistent in both directions. In asymmetric TSP, the

distance from one city to another may differ from the distance in the opposite direction. In this study, symmetric TSP has been examined.

- Distance or Cost Matrix: The TSP is commonly characterized by a matrix detailing the distances or costs associated with traversing between every pair of cities. This study aims to minimize these costs, aligning with the overarching goal of enhancing efficiency and optimization within logistics operations.
- Operational Time: A critical factor in the evaluation is the computational efficiency of the algorithms in MATLAB, with a strong emphasis on swift execution. The algorithms are required to produce results promptly, prioritizing speed without being bound by a specific time frame.
- Evaluation Criteria: Speed of execution of algorithms in MATLAB, cost.

Instead of creating functions from scratch for the analyses to be made with the ACO method, the source codes in the literature suitable for the project scope were searched. Thus, each time was saved, and it was possible to create functions that the accuracy would be tested. As a result of the research, the study continued with the MATLAB files shared by Seyedali Mirjalili at https://seyedalimirjalili.com/aco (Ant Colony Optimization, 2023).

The parts in the original codes that do not fit for purpose and affect the working time have been corrected and made suitable for ACO and Greedy comparison. All functions used in this study and their functions are shown in Table 3.1, Table 3.2, and Table 3.3.

Table 3.1. Mutual functions and patterns used in the study.

Mutual Functions	Definition
model=CreateModel(N, RNG)	It provides a random network of N size and according to the RNG parameter.
drawGraph(model)	It provides the visualization of the randomly generated network.

Table 3.2. Functions and patterns used in the Greedy Method.

Functions of the Greedy method	Definition
p = path_greedy (model, start);	It generates N paths for N nodes.
cost = path_cost (model, p);	Calculate the total length for each path.
drawBestTour_G(model, p_best)	Visualizes the network for the shortest path

Table 3.3. Functions and patterns used in ACO Method.

Functions of the ACO method	Definition
colony = createColony(model, colony , antNo, tau, eta, alpha, beta)	It is the function that creates the ant colony and other necessary information used in the ACO analysis.
drawPhromone(tau , model)	It is the function that visualizes the Phromone density on the network.
drawBestTour(colony , model)	It is the function that visualizes the optimal path on the network.

These functions provide visualization of the results. As an example, the graph of the sample network produced according to N=90 and RNG=1 value is shown in Figure 3.1 and the best path solution of the greedy method for the same mesh is shown in Figure 3.2. For the same network, the Pheromone distribution and the best solution graphs of the ACO method are shown in Figure 3.3 and Figure 3.4.

The best path plot and the edges with the pheromone density are generally the same. In addition, it is seen that the solution in Figure 2.5 has improved depending on the repetitions. However, depending on the pheromone balance (evaporation-addition), the improvement in the solution decreases and stops over time as shown in Figure 3.5.

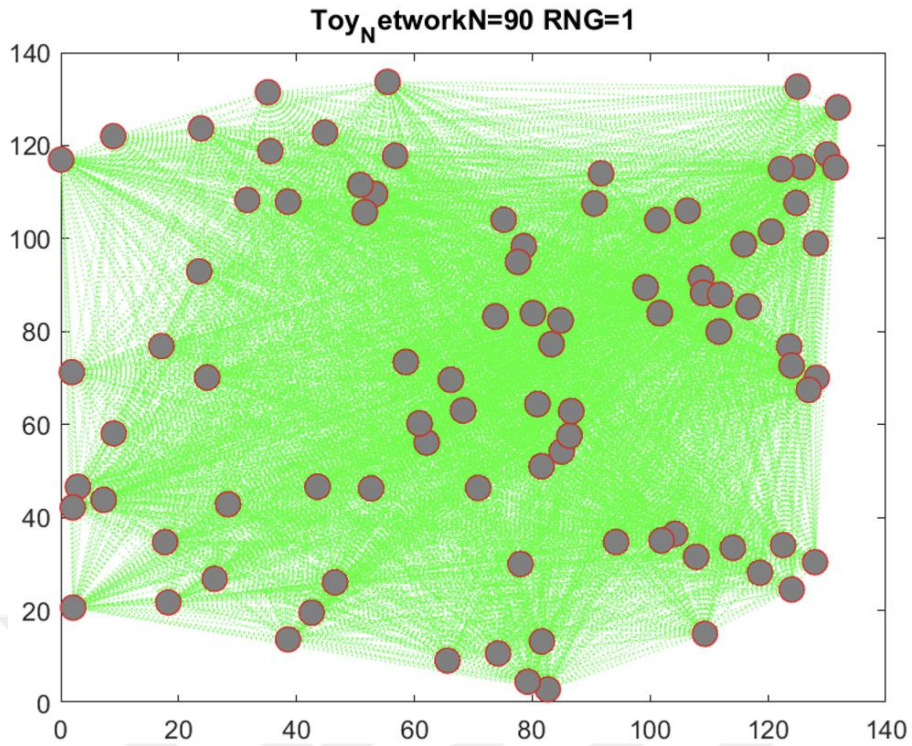

Figure 3.1. A randomly generated network for testing purposes.

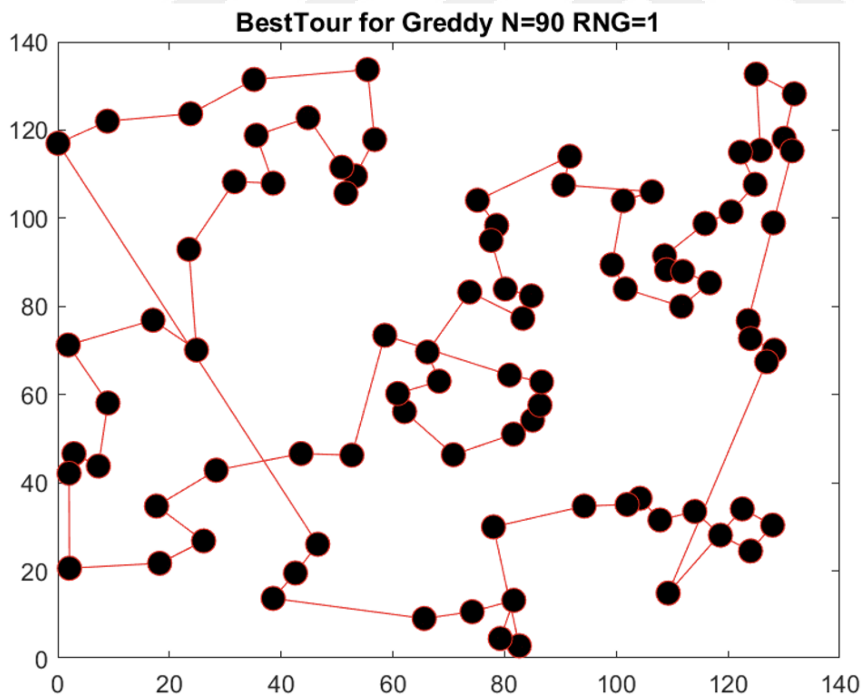

Figure 3.2. The best path is obtained as a result of the solution of the test network with the greedy method.

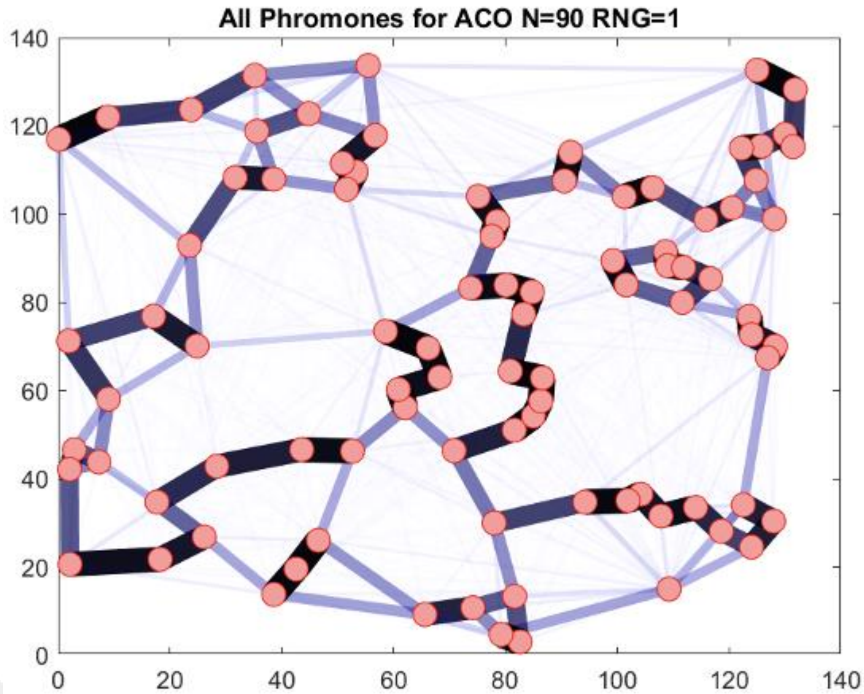

Figure 3.3. Visual results of the ACO method, pheromone distribution on the network.

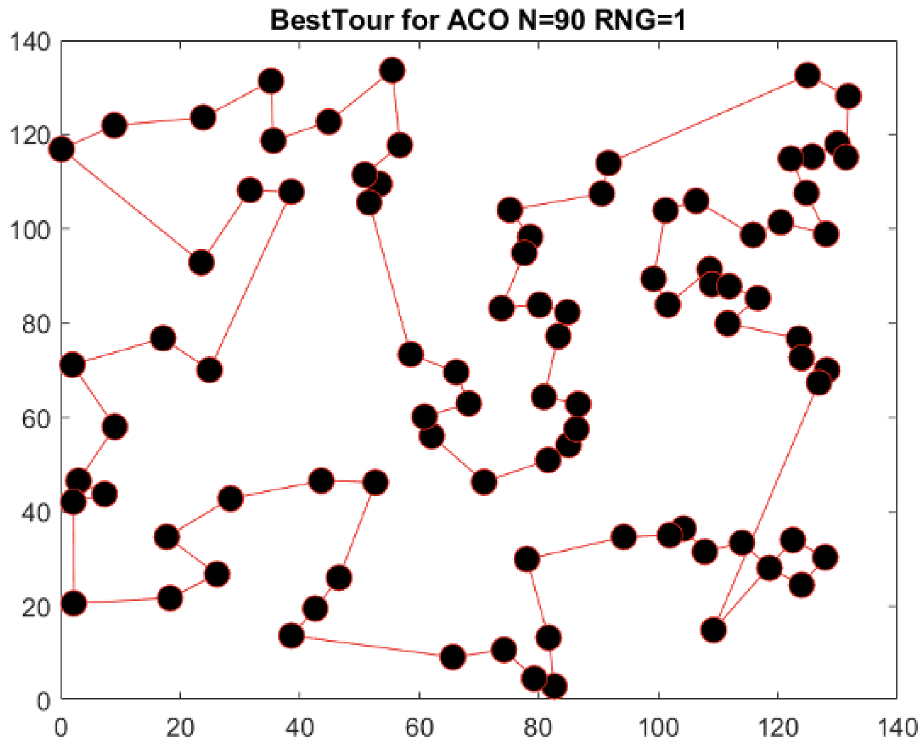

Figure 3.4. Visual results of the ACO method, best path.

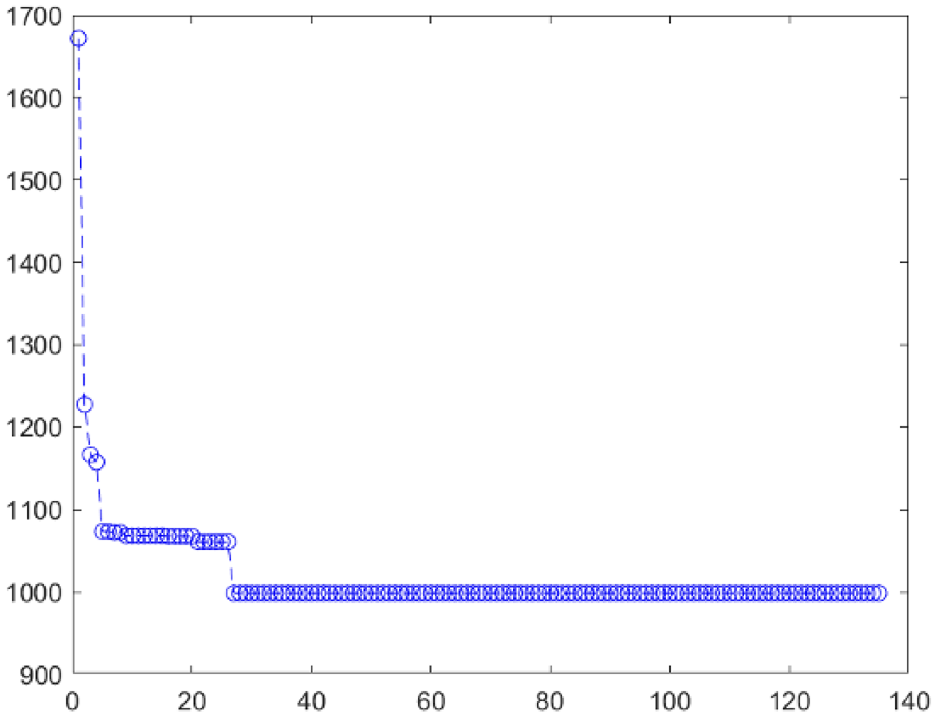

Figure 3.5. Visual results of the ACO method, Improvement in the best path due to iterations.

4. ANALYSIS AND RESULTS

4.1. Parameter Estimation

ACO is a multi-parameter searching method. Shorter paths can be found depending on the combination of parameters. For this purpose, a network was created with the create_model function (N=90 and RNG=1) and tests were conducted. It is seen that when the evaporation coefficient Rho value is 0.8, it gives better results than the others.

The reason for this is that there is excess pheromone accumulation on the roads traveled in the first stages with a low evaporation coefficient. Therefore, the possibility of choosing new routes decreases. Alpha and Beta parameters were tested as 1 and 2.5. The number of ants in the colony and the maximum number of repetitions were taken as 1.5 times and 2 times as values depending on the number of nodes. The high values not only increase the total working time but also enable us to find the best possible solutions. If smaller values are selected, the process may be finished before reaching equilibrium (the last step where the solution does not change) as shown in Figure 2.5. Therefore, the optimal result will not be obtained.

4.2. Performance Evaluation

The Greedy Algorithm and Ant Colony Optimization (ACO) are two distinct optimization algorithms, each equipped with unique strengths tailored to address specific problem characteristics. The Greedy Algorithm, known for its simplicity and efficiency, excels in scenarios requiring quick decision-making and minimal memory usage. It prioritizes local optimization without guaranteeing global optimality.

In contrast, ACO is celebrated for identifying global optima, making it effective in solving complex, multidimensional problems. Its adaptability and suitability for parallel processing enhance its overall efficacy.

The choice between these algorithms depends on the specific problem requirements. While the Greedy Algorithm offers quick solutions, ACO's global optimization capabilities make it advantageous in certain contexts.

In practical applications, choosing between these algorithms involves a thorough evaluation of problem requirements and computational resources. Challenges arise in comparative studies, where one algorithm's success in one test sample may not ensure consistent performance across all samples (Li W. X., 2022).

Hence, in this phase, a comprehensive set of 24 distinct networks was generated by employing unique values for N (N=10:120:5), and for every N number, the algorithm was run 5 times to find more accurate results. Tables 4.1., 4.2., 4.3., 4.4, 4.5 and 4.6. present the respective findings of both the ACO and Greedy methods, encompassing the time taken and the lengths of the optimal paths identified.

Table 4.1. Cost Matrix of ACO

N-Seed	1	2	3	4	5
10	5.756,12	6.026,83	5.892,85	5.756,01	5.723,20
15	6.726,41	7.267,94	7.413,38	6.570,48	6.189,79
20	7.361,94	8.387,11	8.362,75	6.826,53	7.144,36
25	8.369,66	8.607,37	8.880,31	7.263,96	8.433,40
30	8.822,57	9.086,48	10.382,07	7.875,05	8.786,13
35	10.098,27	10.586,75	11.059,57	8.651,60	9.645,90
40	10.756,80	10.899,29	11.353,42	8.860,87	10.374,06
45	11.137,30	12.207,01	11.792,60	9.232,52	11.261,38
50	11.563,87	12.949,08	11.846,65	10.412,83	11.937,94
55	13.198,19	13.264,81	12.281,49	11.011,13	12.505,04
60	13.318,65	13.595,05	13.368,77	11.842,32	13.183,51
65	13.501,81	14.498,24	14.084,49	12.083,60	13.720,31
70	13.807,82	15.316,47	14.431,24	12.666,36	14.224,24
75	14.690,24	15.508,39	15.074,71	12.691,09	14.274,04
80	14.516,25	15.380,53	15.382,23	13.342,59	14.486,49
85	15.393,10	15.742,17	15.532,39	14.247,72	15.222,28
90	15.563,65	16.648,68	15.838,21	14.391,72	15.460,89
95	16.506,51	17.052,15	16.508,02	15.072,41	16.198,89
100	17.401,22	17.543,95	16.847,97	16.546,14	16.397,84
105	17.759,76	18.064,18	17.018,46	16.477,12	16.604,07
110	17.992,71	18.000,69	17.341,83	16.493,89	17.187,10
115	18.752,62	18030,17	17.860,22	16.870,31	17.214,40
120	19.856,00	18.960,00	18.540,00	20.107,00	19.456,00

Table 4.2. Cost Matrix of Greedy

N-Seed	1	2	3	4	5
10	5861,83	6026,83	6127,90	5756,01	5723,20
15	6871,15	7292,57	7413,38	6747,62	6314,08
20	7677,16	8704,52	8633,04	6826,53	7268,65
25	9160,62	8633,22	9491,19	7286,27	9245,26
30	9355,58	9086,48	10456,79	8260,33	9328,22
35	10366,05	11049,54	12071,29	10071,14	10756,02
40	11041,21	11380,95	11994,45	10392,26	11579,04
45	12344,19	13091,74	12594,44	10974,96	12227,53
50	12255,66	14099,61	13805,56	11742,59	12693,82
55	14553,02	15313,77	14422,60	11301,36	13282,37
60	14789,91	16151,44	15311,26	12390,35	14350,72
65	14815,97	16840,63	15819,29	12435,59	14710,74
70	15621,74	17589,30	15527,65	13237,05	15309,51
75	15766,36	17500,83	15426,78	13529,62	15850,01
80	16088,69	17036,07	15867,24	14558,61	16033,72
85	16730,08	18233,07	16547,10	15090,84	16565,37
90	16908,15	17828,86	17394,83	15672,93	17091,53
95	17680,26	19096,40	17666,96	15741,33	17480,24
100	19606,09	19364,90	18109,61	17603,51	17736,75
105	19930,12	19370,67	18341,07	18424,71	18711,83
110	19437,25	19422,69	18619,48	18567,97	18359,13
115	19840,36	19709,63	19234,02	18182,29	17901,94
120	19857,88	20201,76	19990,27	18313,86	17987,22

Table 4.3. The cost difference between ACO and Greedy

N-Seed	1	2	3	4	5
10	105,71	0,00	235,05	-	-
15	144,74	24,63	0,00	177,14	124,29
20	315,22	317,40	270,30	-	124,29
25	790,96	25,85	610,88	22,31	811,86
30	533,01	-	74,71	385,28	542,09
35	267,78	462,79	1.011,72	1.419,53	1.110,12
40	284,40	481,66	641,04	1.531,39	1.204,98
45	1.206,89	884,74	801,84	1.742,44	966,16
50	691,79	1.150,53	1.958,91	1.329,77	755,89
55	1.354,83	2.048,96	2.141,11	290,24	777,34
60	1.471,27	2.556,39	1.942,49	548,02	1.167,22
65	1.314,17	2.342,39	1.734,80	351,99	990,43
70	1.813,92	2.272,83	1.096,41	570,69	1.085,28
75	1.076,12	1.992,44	352,07	838,52	1.575,97
80	1.572,44	1.655,53	485,01	1.216,02	1.547,22
85	1.336,97	2.490,90	1.014,70	843,12	1.343,09
90	1.344,49	1.180,18	1.556,62	1.281,21	1.630,64
95	1.173,75	2.044,25	1.158,94	668,92	1.281,34
100	2.204,87	1.820,95	1.261,64	1.057,37	1.338,91
105	2.170,36	1.306,50	1.322,60	1.947,58	2.107,76
110	1.444,54	1.422,00	1.277,66	2.074,08	1.172,03
115	1.087,74	1.679,63	1.373,80	1.311,98	687,53
120	1,88	1.241,76	1.450,27	- 1.793,14	- 1.468,78

Table 4.4. Time Matrix of ACO

N-Seed	1	2	3	4	5
10	0,56	0,52	0,52	0,52	0,52
15	1,66	1,66	1,66	1,67	1,69
20	3,89	3,87	3,93	4,42	4,50
25	8,69	8,70	8,75	8,77	8,08
30	12,72	12,70	12,87	12,76	12,87
35	20,05	20,10	20,36	20,36	20,12
40	30,49	30,03	29,90	30,18	29,92
45	43,11	43,13	42,86	42,59	42,61
50	58,51	58,43	58,63	58,95	59,02
55	78,38	77,90	78,29	78,34	77,99
60	103,77	101,75	102,86	101,87	103,95
65	129,67	128,86	129,47	130,60	129,55
70	164,17	161,95	163,18	163,95	163,85
75	202,46	201,31	201,33	201,29	200,42
80	285,18	316,31	316,39	299,12	293,04
85	369,77	350,66	358,28	387,46	378,79
90	442,62	446,77	360,56	356,55	382,24
95	484,99	425,31	511,86	486,10	486,66
100	608,07	570,70	552,39	527,78	618,27
105	703,90	710,73	668,35	665,43	673,72
110	755,75	761,30	727,34	747,88	789,84
115	903,39	875,00	858,19	814,00	893,48
120	1023,00	1065,00	1148,00	1103,00	1096,00

Table 4.5. Time Matrix of Greedy

N-Seed	1	2	3	4	5
10	0,01	0,00	0,00	0,00	0,00
15	0,01	0,01	0,01	0,01	0,01
20	0,01	0,01	0,01	0,01	0,01
25	0,02	0,02	0,02	0,02	0,02
30	0,03	0,03	0,03	0,03	0,03
35	0,04	0,04	0,04	0,04	0,04
40	0,05	0,05	0,05	0,05	0,05
45	0,06	0,06	0,06	0,06	0,06
50	0,07	0,07	0,07	0,07	0,07
55	0,08	0,08	0,09	0,09	0,08
60	0,10	0,10	0,10	0,10	0,10
65	0,12	0,12	0,12	0,12	0,12
70	0,13	0,14	0,14	0,14	0,14
75	0,16	0,16	0,16	0,16	0,15
80	0,18	0,20	0,20	0,20	0,21
85	0,22	0,21	0,22	0,26	0,27
90	0,24	0,26	0,22	0,22	0,22
95	0,26	0,32	0,27	0,33	0,25
100	0,36	0,36	0,30	0,37	0,30
105	0,34	0,40	0,33	0,33	0,39
110	0,43	0,43	0,43	0,44	0,44
115	0,48	0,52	0,35	0,47	0,41
120	0,51	0,41	0,41	0,39	0,43

Table 4.6. The time difference between ACO and Greedy

N-Seed	1	2	3	4	5
10	0,01	0,00	0,00	0,00	0,00
15	0,01	0,01	0,01	0,01	0,01
20	0,01	0,01	0,01	0,01	0,01
25	0,02	0,02	0,02	0,02	0,02
30	0,03	0,03	0,03	0,03	0,03
35	0,04	0,04	0,04	0,04	0,04
40	0,05	0,05	0,05	0,05	0,05
45	0,06	0,06	0,06	0,06	0,06
50	0,07	0,07	0,07	0,07	0,07
55	0,08	0,08	0,09	0,09	0,08
60	0,10	0,10	0,10	0,10	0,10
65	0,12	0,12	0,12	0,12	0,12
70	0,13	0,14	0,14	0,14	0,14
75	0,16	0,16	0,16	0,16	0,15
80	0,18	0,20	0,20	0,20	0,21
85	0,22	0,21	0,22	0,26	0,27
90	0,24	0,26	0,22	0,22	0,22
95	0,26	0,32	0,27	0,33	0,25
100	0,36	0,36	0,30	0,37	0,30
105	0,34	0,40	0,33	0,33	0,39
110	0,43	0,43	0,43	0,44	0,44
115	0,48	0,52	0,35	0,47	0,41
120	0,51	0,41	0,41	0,39	0,43

It is noteworthy that, ideally, as the parameter "n" changes, both the Greedy Algorithm and Ant Colony Optimization (ACO) persist in generating solutions without inherent restrictions. However, it becomes evident that ACO may experience prolonged computation times, particularly as "n" increases, presenting challenges in time-sensitive logistics and distribution problem-solving scenarios.

The Greedy Algorithm, while simple, has limitations. Unlike ACO, it may struggle with adaptability in dynamic environments and can converge to suboptimal solutions due to its myopic decision-making. Additionally, it's prone to getting stuck in local optima, limiting exploration of the broader solution space.

Moreover, it is important to highlight that although ACO yields cost-effective results, the computational costs, specifically in terms of time and memory usage, are generally higher. In practical scenarios, this becomes a critical consideration, as the cost advantage of the solution needs to be weighed against the resource-intensive demands of the optimization process.

In negotiating the intricacies of this delicate equilibrium, a discerning analysis prompts a directed consideration toward the nuanced advantages inherent in the application of the Greedy Algorithm within the specialized domain of logistics and the resolution of distribution-centric problem-solving. This reflective exploration underscores the importance of a judicious approach in selecting algorithmic methodologies, particularly emphasizing the intricate dynamics at play within the complex logistical landscape. The nuanced benefits attributed to the Greedy Algorithm emerge as a focal point, warranting scholarly attention and contributing to the scholarly discourse on optimal algorithmic choices in logistics optimization endeavors.

In this comprehensive analysis, the performance of two influential optimization algorithms, the greedy algorithm and ant colony optimization (ACO), applied to the traveling salesman problem was scrutinized. Across varying problem sizes, ranging from 10 to 120 nodes, a meticulous evaluation of both algorithms' efficacy was conducted in terms of solution quality, computational time, and statistical robustness. By examining not only the raw cost and time results but also the underlying statistical data, including minimum, average, maximum, and standard deviation, a nuanced understanding of each algorithm's capabilities and limitations was aimed to be provided. Through this rigorous comparison, decision-makers in supply chain management are empowered with valuable insights into the trade-offs between solution quality and computational efficiency, aiding in informed algorithm selection tailored to specific needs and constraints.

As indicated by the results section, the strengths and limitations of both methods align with findings from existing literature. Despite the Greedy method falling short of achieving results on par with ACO, it remains a viable and successful approach, producing outcomes that closely approximate optimal solutions.

4.3. Cost Results

The cost results from the greedy algorithm show that, on average, solutions tend to have higher costs compared to ACO, especially for larger problem instances.

The average cost increases steadily with problem size, indicating that the greedy algorithm struggles to find optimal or near-optimal solutions as the problem complexity increases. ACO is better able to find solutions with lower costs for larger and more complex problem instances compared to the greedy algorithm.

The minimum and maximum costs vary across different problem sizes, indicating variability in solution quality achieved by both ACO and Greedy.

The average costs obtained from the ACO generally have lower standard deviations compared to Greedy, suggesting less variability in solution quality.

Conversely, ACO demonstrates the ability to identify more optimal routes, achieving improvements between 3-10% as shown in Table 4.7 and Table 4.8. However, it is acknowledged that these results could be further enhanced with meticulous parameter tuning.

Table 4.7. Cost results of ACO.

N	Min.	Avg.	Max.	Std. Deviation
10	5.723,20	5.831,01	6.026,83	127,50
15	6.189,79	6.833,60	7.413,38	504,98
20	6.826,53	7.616,54	8.387,11	718,07
25	7.263,96	8.310,94	8.880,31	617,79
30	7.875,05	8.990,46	10.382,07	902,76
35	8.651,60	10.008,42	11.059,57	924,64
40	8.860,87	10.448,89	11.353,42	954,37
45	9.232,52	11.126,16	12.207,01	1142,00
50	10.412,83	11.742,07	12.949,08	908,96
55	11.011,13	12.452,13	13.264,81	911,79
60	11.842,32	13.061,66	13.595,05	697,58
65	12.083,60	13.577,69	14.498,24	916,88
70	12.666,36	14.089,22	15.316,47	967,61
75	12.691,09	14.447,70	15.508,39	1083,15
80	13.342,59	14.621,62	15.382,23	839,61
85	14.247,72	15.227,53	15.742,17	579,93
90	14.391,72	15.580,63	16.648,68	811,67
95	15.072,41	16.267,60	17.052,15	735,46
100	16.397,84	16.947,42	17.543,95	508,61
105	16.477,12	17.184,72	18.064,18	701,71
110	16.493,89	17.403,24	18.000,69	628,85
115	16.870,31	17.745,51	18.752,62	734,24
120	18.540,00	19.383,80	20.107,00	640,86

Table 4.8. Cost Results of Greedy.

N	Min.	Avg.	Max.	Std. Deviation
10	5723,20	5899,16	6127,90	174,26
15	6314,08	6927,76	7413,38	441,92
20	6826,53	7821,98	8704,52	829,87
25	7286,27	8763,31	9491,19	882,96
30	8260,33	9297,48	10456,79	785,58
35	10071,14	10862,81	12071,29	771,40
40	10392,26	11277,58	11994,45	603,14
45	10974,96	12246,57	13091,74	784,57
50	11742,59	12919,45	14099,61	1006,79
55	11301,36	13774,63	15313,77	1561,58
60	12390,35	14598,73	16151,44	1404,78
65	12435,59	14924,44	16840,63	1637,40
70	13237,05	15457,05	17589,30	1543,44
75	13529,62	15614,72	17500,83	1415,87
80	14558,61	15916,86	17036,07	886,44
85	15090,84	16633,29	18233,07	1113,54
90	15672,93	16979,26	17828,86	809,12
95	15741,33	17533,04	19096,40	1193,39
100	17603,51	18484,17	19606,09	936,60
105	18341,07	18955,68	19930,12	678,40
110	18359,13	18881,30	19437,25	510,29
115	17901,94	18973,65	19840,36	885,36
120	17987,22	19270,20	20201,76	1035,89

4.4. Time Results

The computational time of Greedy increases slightly with problem size but remains relatively stable compared to ACO, suggesting that the greedy algorithm scales well for larger problem instances.

The time results for ACO are consistently higher than those for the greedy algorithm across all problem sizes, indicating its higher computational intensity. The time results for the greedy algorithm are consistently low across all problem sizes, indicating its efficiency and ability to quickly compute solutions.

The computational time increases with problem size, and ACO requires more time to explore the solution space and find better-quality solutions, especially for larger problem instances. The computational time of Greedy increases slightly with problem size but remains relatively stable compared to ACO, suggesting that the greedy algorithm scales well for larger problem instances.

The statistical analysis of time data for the greedy algorithm shows low standard deviations, indicating consistency in solution quality and computational time across different runs.

In situations where rapid decision-making takes precedence and cost considerations are relatively less crucial, the Greedy method emerges as notably more advantageous than ACO as shown in Table 4.9. and Table 4.10. However, its functionality is especially pronounced in scenarios with extended waiting times or where adequate preparation is feasible, such as a brief pause before departure.

Table 4.9. Time results of ACO.

N	Min.	Avg.	Max.	Std. Deviation
10	0,52	0,53	0,56	0,02
15	1,66	1,67	1,69	0,01
20	3,87	4,12	4,50	0,31
25	8,08	8,60	8,77	0,29
30	12,70	12,78	12,87	0,08
35	20,05	20,20	20,36	0,15
40	29,90	30,10	30,49	0,24
45	42,59	42,86	43,13	0,26
50	58,43	58,71	59,02	0,26
55	77,90	78,18	78,38	0,22
60	101,75	102,84	103,95	1,03
65	128,86	129,63	130,60	0,63
70	161,95	163,42	164,17	0,90
75	200,42	201,36	202,46	0,72
80	285,18	302,01	316,39	13,99
85	350,66	368,99	387,46	14,91
90	356,55	397,75	446,77	43,98
95	425,31	478,98	511,86	32,04
100	527,78	575,44	618,27	37,83
105	665,43	684,42	710,73	21,24
110	727,34	756,42	789,84	22,69
115	814,00	868,81	903,39	35,21
120	1023,00	1087,00	1148,00	46,47

Table 4.10. Time Results of Greedy.

N	Min.	Avg.	Max.	Std. Deviation
10	0,00	0,00	0,01	0,00
15	0,01	0,01	0,01	0,00
20	0,01	0,01	0,01	0,00
25	0,02	0,02	0,02	0,00
30	0,03	0,03	0,03	0,00
35	0,04	0,04	0,04	0,00
40	0,05	0,05	0,05	0,00
45	0,06	0,06	0,06	0,00
50	0,07	0,07	0,07	0,00
55	0,08	0,08	0,09	0,00
60	0,10	0,10	0,10	0,00
65	0,12	0,12	0,12	0,00
70	0,13	0,14	0,14	0,00
75	0,15	0,16	0,16	0,00
80	0,18	0,20	0,21	0,01
85	0,21	0,24	0,27	0,03
90	0,22	0,23	0,26	0,02
95	0,25	0,29	0,33	0,04
100	0,30	0,34	0,37	0,03
105	0,33	0,36	0,40	0,04
110	0,43	0,44	0,44	0,00
115	0,35	0,45	0,52	0,07
120	0,39	0,43	0,51	0,05

4.5. Held-Karp Results

The Held-Karp algorithm guarantees the exact solution to the TSP by systematically exploring all possible routes through a dynamic programming approach. Its comprehensive examination of subsets and memoization of intermediate results ensure the identification of the optimal route, making it a precise and reliable method for solving the TSP.

Given the lengthy compute time required for the Held-Karp algorithm, it is impractical to test all possible N values. Instead, a smaller N value (N=15) is selected as a sample and solved using the Held-Karp algorithm. The algorithm runs 5 times as in ACO and Greedy algorithms. The results are presented below Tables 4.11, 4.12 and 4.13.

Table 4.11. Time Results of Held-Karp

N-Seed	1	2	3	4	5
15	68,72	77,44	76,52	79,86	79,49

Table 4.12. Cost Results of Held-Karp

N-Seed	1	2	3	4	5
15	6714,67	7267,94	7365,14	6570,48	6189,79

Cost results of Held-Karp, ACO, and Greedy for N=15 have been consolidated below.

Table 4.13. Held-Karp, ACO, Greedy Cost Result Comparison

N	Algorithm	Min.	Avg.	Max.
15	Held-Karp	6189,79	6821,60	7365,137
15	ACO	6.189,79	6.833,60	7.413,38
15	Greedy	6314,08	6927,76	7413,38

It is understood that ACO results differ approximately by 0,19% from the exact result. On the other hand, Greedy results differ approximately by 1,56% from the exact result. This crosscheck shows that using ACO and also Greedy is feasible considering the computing time of Held-Karp which is 61 times longer than ACO and 450 times longer than Greedy.

4.6. Statistical Data Analysis

Calculating the difference of end-to-end operation time of Greedy and ACO, it is seen that ACO has an average of 5% better results than the Greedy algorithm as shown in Table 4.14.

Table 4.14. Total Operation Time Comparison of Greedy and ACO.

N	Total Time of Greedy	Total Time of ACO	Difference
10	5899,16	5.831,54	-1%
15	6927,77	6.835,27	-1%
20	7821,99	7.620,66	-3%
25	8763,33	8.319,54	-5%
30	9297,51	9.003,24	-3%
35	10862,85	10.028,62	-8%
40	11277,63	10.478,99	-7%
45	12246,63	11.169,02	-9%
50	12919,52	11.800,78	-9%
55	13774,71	12.530,31	-9%
60	14598,83	13.164,50	-10%
65	14924,56	13.707,32	-8%
70	15457,19	14.252,64	-8%
75	15614,88	14.649,06	-6%
80	15917,06	14.923,63	-6%
85	16633,53	15.596,52	-6%
90	16979,49	15.978,38	-6%
95	17533,33	16.746,58	-4%
100	18484,51	17.522,86	-5%
105	18956,04	17.869,14	-6%
110	18881,74	18.159,66	-4%
115	18974,1	18.614,32	-2%
120	19270,63	20.470,80	6%

The curve of total operation time comparison of Greedy and ACO has been visualized in Figure 4.1.

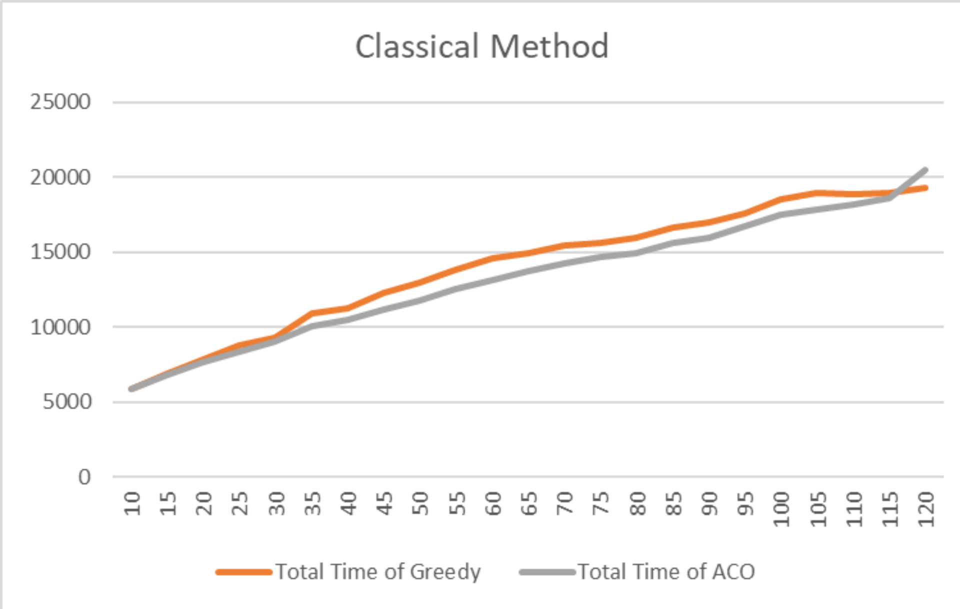

Figure 4.1. Total Operation Time Comparison of Greedy and ACO.

The paired T-Test is performed to evaluate significance. The calculation has been implemented using the Excel formula. Hypothesis has been shared below.

H_0: There is no acceptable difference between the means of the ant colony and greedy total time groups that is different from zero.

H_1: There is an acceptable difference between the means of the ant colony and greedy total time groups that is different from zero.

The Total Time of Greedy and Total Time of ACO values have been compared and the *p*-value is obtained as 4,0169E-06 which means there is difference between the solutions produced by the two algorithms is statistically significant. In this case, it can be said that there is a difference between the performance of the two algorithms and ACO is more useful with this dataset.

4.7. Findings

It's evident that while the greedy algorithm showcases higher variability in costs compared to ACO, its computational time is significantly lower. Recognizing the inherent efficiency of the greedy algorithm, a strategy has been proposed to enhance its impact by increasing its repetition across all node values. This approach involves multiplying the average time with the node values, effectively calculating end-to-end (E2E) time when each node is considered a starter node as shown in Table 4.15.

Consequently, the time matrix of the greedy algorithm expands, allowing us to consider the minimum cost of the greedy algorithm in every node value. As a result, the need for standard deviation is obviated. By adopting this approach, the E2E time, factoring in both runtime and cost considerations is anticipated to decrease. This innovative strategy not only harnesses the efficiency of the greedy algorithm but also mitigates its inherent variability, offering promising prospects for improved performance in the traveling salesman problem.

Table 4.15. N Times Run Results of Greedy

N	Min.	Avg.	N Times Run
10	0	0	0
15	0,01	0,01	0,15
20	0,01	0,01	0,2
25	0,02	0,02	0,5
30	0,03	0,03	0,9
35	0,04	0,04	1,4
40	0,05	0,05	2
45	0,06	0,06	2,7
50	0,07	0,07	3,5
55	0,08	0,08	4,4
60	0,1	0,1	6
65	0,12	0,12	7,8
70	0,13	0,14	9,8
75	0,15	0,16	12
80	0,18	0,2	16
85	0,21	0,24	20,4
90	0,22	0,23	20,7
95	0,25	0,29	27,55
100	0,3	0,34	34
105	0,33	0,36	37,8
110	0,43	0,44	48,4
115	0,35	0,45	51,75
120	0,39	0,43	51,6

By calculating the minimum cost of the Greedy algorithm and N times its runtime, the resulting end-to-end (E2E) operation time is observed to be 5% lower than the average cost of ACO combining 5 times the runtime average of ACO as shown in Table 4.16.

Table 4.16. New Approach Total Operation Time Comparison of Greedy and ACO.

N	Total Time of Greedy	Total Time of ACO	Difference
10	5723,2	5.831,54	2%
15	6314,23	6.835,27	8%
20	6826,73	7.620,66	12%
25	7286,77	8.319,54	14%
30	8261,23	9.003,24	9%
35	10072,54	10.028,62	0%
40	10394,26	10.478,99	1%
45	10977,66	11.169,02	2%
50	11746,09	11.800,78	0%
55	11305,76	12.530,31	11%
60	12396,35	13.164,50	6%
65	12443,39	13.707,32	10%
70	13246,85	14.252,64	8%
75	13541,62	14.649,06	8%
80	14574,61	14.923,63	2%
85	15111,24	15.596,52	3%
90	15693,63	15.978,38	2%
95	15768,88	16.746,58	6%
100	17637,51	17.522,86	-1%
105	18378,87	17.869,14	-3%
110	0,43	0,44	48,4%
115	0,35	0,45	51,75%
120	0,39	0,43	51,6%

This finding underscores the potential efficiency gains achievable through the strategic utilization of the Greedy algorithm, particularly when considering both cost and runtime constraints. Such optimization strategies can significantly enhance the overall performance of supply chain route planning, offering a compelling alternative to traditional ACO approaches.

The curve of the new approach results' total operation time comparison of Greedy and ACO has been visualized in Figure 4.2.

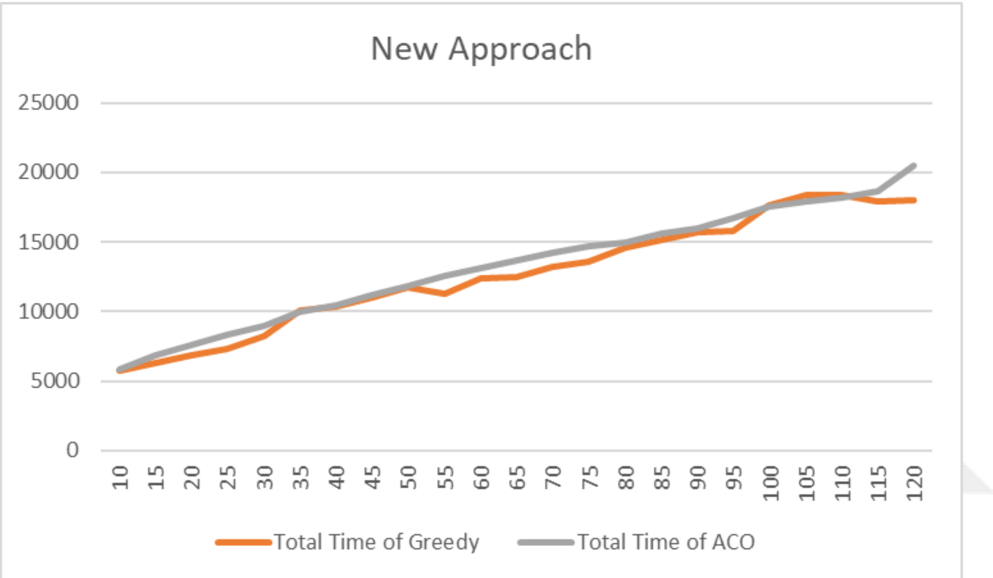

Figure 4.2. New Approach Total Time Results and Comparison of ACO and Greedy

The paired T-test is performed to evaluate the new approach results' significance. The hypothesis has been shared below.

H_0: There is no acceptable difference between the means of the ant colony and the new approach of greedy total time groups that is different from zero.

H_1: There is an acceptable difference between the means of the ant colony and the new approach of greedy total time groups that is different from zero.

The Total Time of Greedy and Total Time of ACO values have been compared and the *p*-value is obtained as 0,00030119 which means there is difference between the solutions produced by the two algorithms is statistically significant. In this case, it can be said that there is a difference between the performance of the two algorithms and ACO is more useful with this dataset.

Although ACO is widely used to find low-cost paths, the use of the new approach of the Greedy method has been determined as the most optimal solution to achieve a result that both produces the route quickly and has a low route cost. By doing a significance test on new approaches' results, it is proven that there is a significant difference between ACO and Greedy method's results which is an average of 5%.

5. CONCLUSION AND RECOMMENDATIONS

The present Traveling Salesman study critically interrogates the prevailing presumption asserting the inherent superiority of sophisticated algorithms, exemplified by the likes of Ant Colony Optimization. By doing so, it underscores the nuanced perspective that each algorithm may confer distinct advantages contingent upon specific contextual considerations. Within the discourse on optimization methodologies, this investigation places particular emphasis on elucidating the significance of harmonizing algorithmic selections with the unique requisites inherent in diverse problem domains.

Moreover, it accentuates the imperative of recognizing and responding to the multifaceted nature of problem scenarios. A key proposition posited herein is the paramount importance of tailoring algorithmic choices in consonance with the distinctive requirements of varying problem contexts. This strategic alignment, it posits, catalyzes unlocking the latent potential of algorithms in addressing complex optimization challenges. Focusing on the strong feature of the Greedy algorithm by iterating the paths as much as node value, the potential of Greedy is proven.

In delineating these principles, the study seeks to underscore the inadequacy of adhering rigidly to a universal mindset in the optimization domain. It offers a nuanced perspective that challenges the notion of a singularly dominant algorithmic paradigm, thereby contributing to the ongoing discourse on optimal algorithmic selection. In essence, the study strives to illuminate the intricate dynamics inherent in optimization scenarios, shedding light on the fallacy of assuming universal optimality in algorithmic approaches across diverse problem landscapes.

The examination of theoretical models assumes a pivotal role in the scholarly exploration of algorithmic optimization. This study underscores the fundamental significance of theoretical models in shaping our comprehension of the intricate dynamics inherent in optimization scenarios, thereby challenging the prevailing assumption of a universal paradigm. Within this context, theoretical models function as intellectual frameworks that facilitate the abstraction and systematic representation of intricate algorithmic processes. They play an essential role in facilitating the development of hypotheses, predictions, and empirical tests, thereby guiding researchers in their endeavor to elucidate the latent potential of algorithms across diverse problem domains. The study accentuates the iterative nature of theoretical model development, emphasizing the requisite continual refinement in response to empirical evidence and evolving conceptual paradigms. In essence, theoretical models emerge as indispensable instruments that not only underpin the study's critical examination of algorithmic superiority but also contribute substantially to the ongoing discourse on optimal algorithmic selection. They argue for a more nuanced and contextually aware approach to the intricate challenges posed by diverse optimization scenarios.

REFERENCES

Abdulkarim, H., & Alshammari, I. (2015, August 4). Comparison of algorithms for solving traveling salesman problem. *International Journal of Engineering and Advanced Technology*, 76-79.

Ant Colony Optimization. (2023). Ant Colony Optimization: https://seyedalimirjalili.com/aco adresinden alındı

Bellmore, M., & Nemhauser, G. (1968). The traveling salesman problem: a survey. M. Bellmore, & G. Nemhauser içinde, *Operations Research* (Cilt 3, s. 538-558).

Blum, C. (2005, 10 11). Ant colony optimization: Introduction and recent trends. *Physics of Life Reviews, 2,* 353-373.

Bollobás, B. (1998). *Modern Graph Theory.* Springer Science & Business Media.

Borgatti, S. P. (2011). On Network Theory. *Organization Science, 5*(22), 1168-1181.

Chauhan, C., Gupta, R., & Pathak, K. (2012). Survey of methods of solving tsp along with its implementation using dynamic programming approach. *International journal of computer applications*, 12-19.

Chivers, I., & Sleightholme, J. (2015). An introduction to Algorithms and the Big O Notation. I. Chivers, & J. Sleightholme içinde, *Introduction to programming with Fortran* (s. 359-365). Springer.

Cordon, O. H. (2002). A review on the ant colony optimization metaheuristic: Basis, models and new trends. *Mathware and Soft Computing*, *9*(2-3), 141–175.

Curtis, S. (2003). The classification of greedy algorithms. *Science of Computer Programming*, 125-157.

Çalişkan, K. (2011). Karınca kolonisi optimizasyonu ile araç rotalama probleminin maliyetlerinin kümeleme tekniği ile iyileştirilmesi. *Master's thesis, Fen Bilimleri Enstitüsü*. TOBB Ekonomi ve Teknoloji Üniversitesi - Fen Bilimleri Enstitüsü - Bilgisayar Mühendisliği Anabilim Dalı.

Danon, L. F. (2011). Networks and The Epidemiology of Infectious Disease. *Interdisciplinary Perspectives on Infectious Diseases, 2011,* 1-28. doi:10.1155/2011/284909

Davis, T. A. (2010). *MATLAB Primer.* CRC Press.

de Aragao, M. P. (2002). On the implementation of MST-based heuristics for the Steiner problem in graphs. *In Workshop on algorithm engineering and experimentation* (s. 1-15). Berlin, Heidelberg: Springer Berlin Heidelberg.

Dean-Leon, E., Nair, S., & Knoll, A. (2012). User friendly Matlab-toolbox for symbolic robot dynamic modeling used for control design. *IEEE International Conference on Robotics and Biomimetics (ROBIO)* (s. 2181-2188). IEE.

Donbosco, I. S., & Chakraborty, U. K. (2021). Comparing Performance of Evolutionary Algorithms - A Travelling Salesman Perspective. *11th International Conference on Cloud Computing, Data Science & Engineering (Confluence) Cloud Computing, Data Science & Engineering* (s. 416-425). IEEE.

Dorigo, M. B. (2006). Ant colony optimization. IEEE Computational Intelligence Magazine. *IEEE Computational Intelligence Magazine*, 28-39.

Dorigo, M. M. (1991). *The ant system: An autocatalytic optimizing process.* Milano: Dipartimento di Elettronica e Informazione Politecnico di Milano.

Dorigo, M., & Blum, C. (2005). Ant colony optimization theory: A survey. *Theoretical computer science*, 243-78.

Flood, M. M. (1948). A game theoretic study of the tactics of area defense. In *RAND Research Memorandum* (pp. 1-29).

Gil-Gala, F. J., Durasević, M., Sierra, M. R., & Varela, R. (2023). Evolving ensembles of heuristics for the travelling salesman problem. *Natural Computing: An International Journal*, 671-684.

Gómez, S. (2019). Centrality in Networks: Finding the Most Important Nodes. *Business and consumer analytics: New ideas*, 401-433.

Jaiswal, U. &. (2011). Ant colony optimization. *International Journal of Scientific & Engineering Research*, 1-7.

Jünger, M., Reinelt, G., & Rinaldi, G. (1995). Travelling Salesman Problem. In M. Jünger, G. Reinelt, & G. Rinaldi, *Handbooks in Operation Research and Management Science* (pp. 225-330).

Kim, H., Park, C., & Wang, M. (2018). Paired T-Test Based On Robustified Statistics. *대한산업공학회 추계학술대회 논문집*, (s. 2347-2353).

Kim, T. K. (2015). T test as a parametric statistic. *Korean journal of anesthesiology*, 540–546. doi:10.4097/kjae.2015.68.6.540

Kizilateş, G., & Nuriyeva, F. (2013). On the nearest neighbor algorithms for the traveling salesman problem. *Advances in Computational Science, Engineering and Information Technology*, 111-118.

Li, W. (2022). A new approach to the traveling salesman problem. *In Proceedings of the 2022 ACM Southeast Conference* (s. 52-59). Michigan: University of Michigan-Flint.

Li, W. X. (2022). An ant colony optimization algorithm with adaptive greedy strategy to optimize path problems. *Journal of Ambient Intelligence and Humanized Computing*, 1-15.

Limentani, G. B. (2005). Beyond the t-test: statistical equivalence testing. *Analytical Chemistry* (s. 222-226). içinde American Chemical Society.

Manfei, X. U. (2017). The differences and similarities between two-sample t-test and paired t-test. *Shanghai archives of psychiatry*, 184. doi:https://doi.org/10.11919/j.issn.1002-0829.217070

Mavrovouniotis, M. Y. (2014). Ant colony optimization with self-adaptive evaporation rate in dynamic environments. *computational intelligence in dynamic and uncertain environments (CIDUE)* (s. 47-54). IEEE.

Melo, T. N.-d.-G. (2008). Network design decisions in supply chain planning.

Olivos-Castillo, I. C.-M.-M.-A. (2019). An optimal greedy algorithm for the single access contention resolution problem. *IEEE Access*. doi:https://doi.org/10.1109/access.2019.2902358

Osaba, E. Y.-S.-G. (2016). An improved discrete bat algorithm for symmetric and asymmetric Traveling Salesman Problems. *Engineering Applications of Artificial Intelligence*, 59-71.

Parjadis, A. C. (2023). Learning Lagrangian Multipliers for the Travelling Salesman Problem. arXiv preprint arXiv:2312.14836.

Park, Y. B. (2001). A hybrid genetic algorithm for the vehicle scheduling problem with due times and time deadlines. *Int. J. Production Economics*(73), 175-188.

Rachmat, Z. Y., & Mandala, R. (2022). Improving Virus Colony Search Performance on Travelling Salesman Problem Case. *2022 5th International Conference on Information and Communications Technology (ICOIACT) Information and Communications Technology (ICOIACT)* (s. 468-472). IEEE.

Righini, G. (2021). Efficient optimization of the Held–Karp lower bound. *Open Journal of Mathematical Optimization*, 1-17.

Rubinstein-Salzedo, S. (2018). Big o notation and algorithm efficiency. *Cryptography* (s. 75-83). içinde Springer.

S. Dasgupta, C. P. (tarih yok). *Berkeley University Dynamic Programming Lecture* . https://people.eecs.berkeley.edu/~vazirani/algorithms/chap6.pdf adresinden alındı

Saraei, M. A. (2015). Solving of travelling salesman problem using firefly algorithm with greedy approach. *Fen Bilimleri Dergisi (CFD)*, 36.

Sathya, N., & Muthukumaravel, A. (2015). A review of the optimization algorithms on traveling salesman problem. *Indian Journal of Science and Technology*, 1-4.

Şeker, S. (2015). *Çizge Teorisi* (Vol. 2). YBS Ansiklopedi.

Wallisch, P. L. (2014). *MATLAB for neuroscientists: an introduction to scientific computing in MATLAB.* Academic Press.

Ye, C. Y. (2014). An efficient and scalable algorithm for the traveling salesman problem. *2014 IEEE 5th International Conference on Software Engineering and Service Science* (s. 335-339). Shanghai: IEEE. doi:https://doi.org/10.1109/icsess.2014.6933576

CURRICULUM VITAE

Name Surname : Merve Ece Görgün

EDUCATION:

- **Undergraduate** : 2018, Sakarya University, Engineering Faculty, Industrial Engineering
- **Graduate** : 2024, Sakarya University, Graduate School of Nature and Applied Science, Industrial Engineering

PROFESSIONAL EXPERIENCE AND AWARDS:

- 2018- 2021 International SC Allocation Specialist / LC Waikiki
- 2021 Logistics Process Development and Lean Project Engineer / LC Waikiki
- 2021- International SC Allocation MENA Region Manager / LC Waikiki

PUBLICATIONS, PRESENTATIONS AND PATENTS ON THE THESIS:

- Görgün, M. E., Denizhan, B. (2023, 25-26, December). Optimizing TSP: A Comprehensive Analysis of ACO and the Greedy Algorithm in Logistics. *International Congress – 3rd International Conference on Scientific and Academic Research,* Konya, Turkey.