T.R.

GEBZE TECHNICAL UNIVERSITY

FACULTY OF ENGINEERING

DEPARTMENT OF COMPUTER ENGINEERING

MINIMUM GENERAL ROUTING PROBLEM

MOHAMMAD ASHRAF YAWAR 161044123

SUPERVISOR
DOÇ.DR.DIDEM GÖZÜPEK

GEBZE
20.01.2022

T.R.
GEBZE TECHNICAL UNIVERSITY
FACULTY OF ENGINEERING
COMPUTER ENGINEERING DEPARTMENT

MINIMUM GENERAL ROUTING PROBLEM

MOHAMMAD ASHRAF YAWAR 161044123

SUPERVISOR
DOÇ.DR.DIDEM GÖZÜPEK

20.01.2022
GEBZE

This study has been accepted as an Undergraduate Graduation Project in the Department of Computer Engineering on 20/01/2022 by the following jury.

JURY

Member
(Supervisor) : Doç.Dr.Didem GÖZÜPEK

Member : Yrd.Doç.Dr. Zafeirakis Zafeirakopoulos

ABSTRACT

Graph routing issues have received a great deal of attention in the domains of operations research, computer science, and engineering due to its pervasiveness and breadth of applications. The traveling salesman problem is a good illustration of a routing problem. in this report we will consider a variation of TSP called general routing problem and show you the possible approaches. in our problem: we are given a complete undirected graph $G = (V, E)$ with a metric cost function c and For a given subset V' of V and subset E' of E, the task is to find a walk T with minimum cost such that it visits each vertex in V' exactly once and covers each edge in E' at least once.In GRP, we are given a vertex subset V and an edge subset E', as well as a graph G and the metric cost function c. The key distinction between TSP and GRP is that the viable solution of GRP does not need to visit the vertices in V, but only the vertices in V'. Furthermore, the viable GRP solution must cross through each edge of E' at least once. The objective is to design a solution that fits the aforementioned restrictions while being as inexpensive as possible. $Jansen(1992)$ devised an approximation approach with ratio 1.5 for GRP based on $Christofides'$ algorithm for metric TSP. Because GRP is a broadening of TSP,it is obvious that the aforementioned method is equally well-known for GRP. Any additional advancement on the GRP approximation ratio will result in an improvement for TSP.We generalize the known heuristic of Christofides for the TSP with triangle inequality and approximate ratio $3/2$ to the GRP.

Keywords: tsp, grp, christofide's algorithm,graph, routing.

ÖZET

Graf yönlendirme Problemleri, yaygınlığı ve uygulama genişliği nedeniyle yöneylem araştırması, bilgisayar bilimi ve mühendislik alanlarında büyük ilgi görmüştür.TSP yada Gezgin satıcı problemi adıyla meşhur olan bu problem, bir yönlendirme probleminin iyi bir örneğidir. Bu raporda, genel yönlendirme problemi olarak adlandırılan bir Genel Graf yönlendirme Problemli (GRP) varyasyonunu ele alacağız ve size olası yaklaşımları göstereceğiz. bizim sorunumuzda: bize bir metrik maliyet fonksiyonu c olan tam bir yönsüz $G = (V, E)$ grafı veriliyor ve Vnin belirli bir V' alt kümesi ve E'nin E' alt kümesi için, görev bir yürüyüş T bulmaktır. V''daki her köşeyi tam olarak bir kez ziyaret edecek ve E''daki her kenarı en az bir kez kaplayacak şekilde minimum maliyetli bir algoritmadan bahsedcez. GRP'de, bize bir köşe alt kümesi V' ve bir kenar alt kümesi E' verilir. TSP ve GRP arasındaki temel fark, uygulanabilir GRP çözümünün Vdeki köşeleri değil, sadece V''daki köşeleri ziyaret etmesi gerekmesidir. Ayrıca, uygulanabilir GRP çözümü, E''ın her bir kenarından en az bir kez geçmelidir. Amaç, yukarıda belirtilen kısıtlamalara uyan ve mümkün olduğu kadar ucuz olan bir çözüm tasarlamaktır. $Jansen(1992)$, metrik TSP için $Christofides'$ algoritmasına dayalı olarak GRP için 1.5 oranlı bir yaklaşım yaklaşımı tasarladı. GRP, TSP'nin bir uzantısı olduğu için, yukarıda belirtilen yöntemin GRP için eşit derecede iyi bilindiği açıktır. GRP yaklaşıklık oranındaki herhangi bir ek ilerleme, TSP için bir iyileştirme ile sonuçlanacaktır. TSP için Christofides'in bilinen buluşsal yöntemini, üçgen eşitsizliği ve yaklaşık $3/2$ oranıyla GRP'ye genelleştiriyoruz.

Keywords: tsp, grp, christofide's algorithm,graph, routing.

ACKNOWLEDGEMENT

First and foremost, praises and thanks to the God, the Almighty, for His showers of blessings throughout my research work to complete the research successfully. I would like to express my deep and sincere gratitude to my research supervisor, Doç.Dr.Didem GÖZÜPEK and Yrd.Doç.Dr. Zafeirakis Zafeirakopoulos for giving me the opportunity to do research and providing invaluable guidance throughout this research. Didem Hocam: her dynamism, vision, sincerity and motivation have deeply inspired me. she has taught me the methodology to carry out the research and to present the research works as clearly as possible. It was a great privilege and honor to work and study under her guidance. I am extremely grateful for what she has offered me.

I would also like to thank her for her assistance empathy, and great sense of humor. Didem Hocam has been extremely supportive to no to give up on this research and continue until the end and i specially want to thank her, Prof Zafeirakis has been also excellent supportive and helpful through out this research and i also want to thanks him for his support and assistance.

Mohammad Ashraf YAWAR 161044123

INTRODUCTION

Finding an ideal route for a single vehicle on a given network is a tough combinatorial optimization issue. A network is defined as a connected graph G with a set V of vertices, a set E of edges, and positive weights (or costs, distances) on the edges $c : E - iw+$. The traditional traveling salesman issue involves visiting all vertices in a given network with the lowest possible cost (TSP). The Chinese postman problem is the difficulty of covering all edges of a network at the lowest overall cost (CPP). The rural postman problem arises when just a subset E' of all edges must be traveled (RPP). The general routing problem (GRP) seeks a minimal cost cycle that visits each vertex in a required subset V' precisely once and travels each edge in a necessary subset E' of the network.

consider the general routing problem and the following graph $G|V', E' = (\{v \mid v \subset e \subset E'\} \cup V', E')$. At first we assume that this graph is connected. A GRP tour can only exists if there is no vertex $v \subset V'$ with degree greater than 2 in the graph $G|v', E'$. Therefore we consider the case that all vertices $v \subset V'$ have degree $0 < d(v) \leq 2$ and that the other have degree $d(v) > 0$.

A GRP tour requires k extra edges with odd degrees between vertices $\overline{\mathrm{V}}$. An edge of this type can alternatively be a route with intermediate vertices in $V \backslash V'$. To begin, we may build the shortest path over these edges for each pair of vertices with an odd degree. This may be accomplished by modifying a traditional shortest-path method. We acquire k alternative pairings of vertices with odd degrees and hence k pathways with intermediate vertices in $V \backslash V'$ by solving the weighted matching problem with these costs. Such a match cannot exist; imagine a three-length path where both edges belong to E' and the center vertex belongs to V'.

Keywords: GRP, TSP, Christofides',graph, routing prblems.

LIST OF SYMBOLS AND ABBREVIATIONS

Symbol or Abbreviation	:	**Explanation**
GRP	:	General Routing Problem
TSP	:	Travelling Salesmen Problem
$\subset$	:	Subset of
$\in$	:	In
$V \backslash V'$	:	All items in V which are not in V'
V'	:	V Prime

CONTENTS

LIST OF FIGURES

1. GENERAL ROUTING PROBLEM

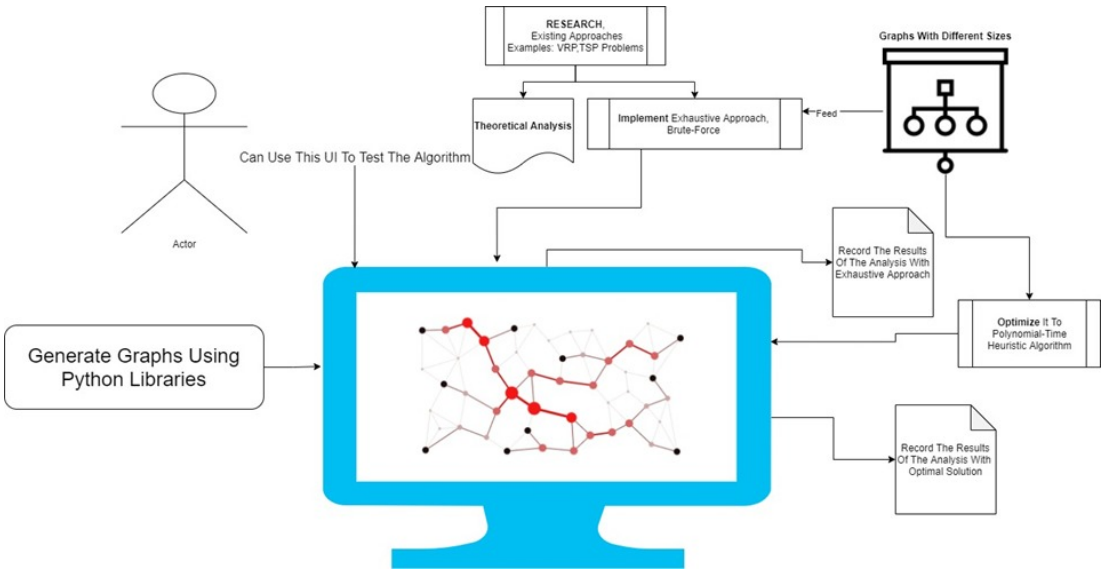


Figure 1.1: Project Work Flow

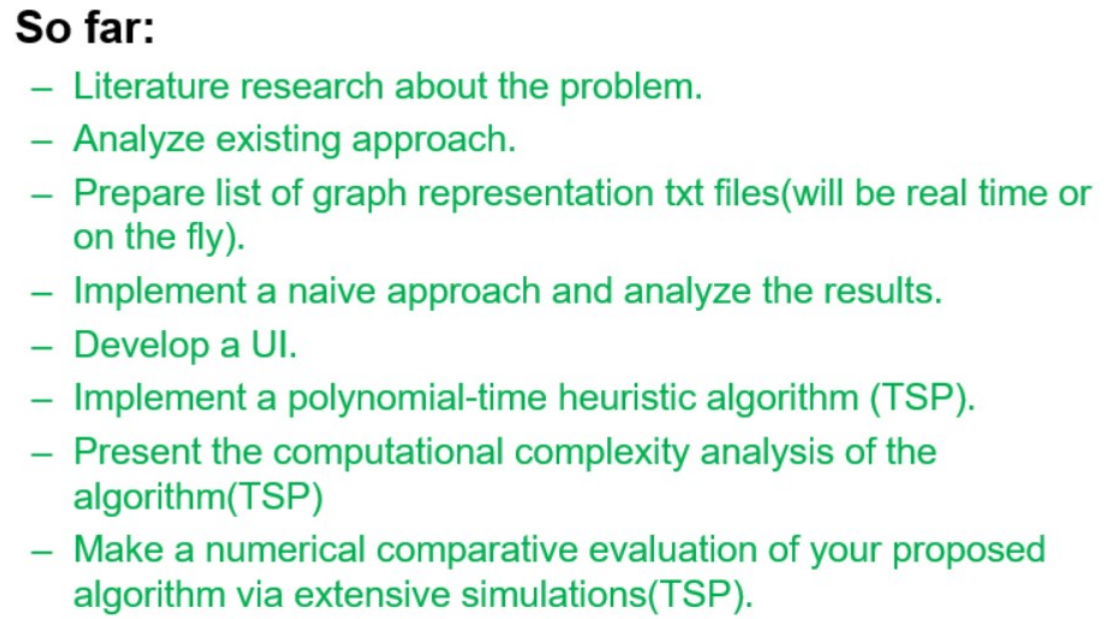


Figure 1.2: Projects Overall Progress

2. CHRISTOFIDS' ALGORITHM FOR TSP

Since our resultant algorithm is the generalized version of the christofides' approach of TSP Befor we explain the algorithm for General Routing Problem, let's first explain the christofid's approximation algorithm for $TravellingSalsemenProbelm$:

The Christofides algorithm, also known as the Christofides–Serdyukov algorithm, is a method for finding approximate solutions to the travelling salesman problem where the distances form a metric space (they are symmetric and obey the triangle inequality).It is named after Nicos Christofides and Anatoliy I. Serdyukov, who separately developed it in 1976. It is an approximation method that ensures that its answers will be within a factor of $3/2$ of the ideal solution length. Since the algorithm is multistep in nature, it's running time and complexity varies based on the running time its components. The upper bound on computations for Christofides' Algorithm is roughly $O(|V|^4)$ which is significantly better than any of the exact solution approaches. christofides' explained his algorithm with 5 steps, let's explain then one at a time:

2.1. Step 1 » Finding Minimum Spanning Tree:

Let us explain this algorithm on a real-life example: consider $Figure2.1$ where each red dot indicates a vertex in the complete undirected graph and The first step of Christofides' Algorithm is to take the given graph and construct an undirected minimum spanning tree This MST by definition will have the lowest total cost of edges that are needed to make all nodes share a common path. This is usefull because it means we now have just have to add a few more edges to the MST to satisfy the TSP constraints (namely that each node can only be visited once thus no backtracking allowed). to calculate the MST we can make use of wellknown algorithm such us: $Kruksal'salgorithm$ or $Primes'algorithm$, $Primes'algorithm$ is slidely better than the $Kruksal's$ one where it can calculate the MST in $O(|E| + |V|log(|V|))O(E + Vlog(V))$ time where E is the number of edges and V is the number of vertices. Note that since the TSP works on a complete graph, we can simplify $E = V(V - 1)$. we have MST as $Figure2.2$.

Figure 2.1: Each red dot indicates a Vertex in the graph.

Figure 2.2: Minimum Spanning Tree for above graph.

2.2. Step 2 » Constructing The Odd Degree Vertices:

The second step of Christofides' Algorithm is to calculate all the vertices in the MST that have an odd degree (Note degree referes to BOTH in and out edges). We do this because the MST by definition is acyclic and as a result we want to add edges to the MST that would allow traversal throughout all the vertices in a single cycle without repeating nodes.

Observe the $Figure 2.3$ We focus on odd degree vertices in particular because when we traverse the edges of the graph for a solution to the TSP, we are essentially going to "walk though" the graph though the edges and each time select an edge that we haven't visited before. If we do this with a graph with an odd degree vertex then eventually we will get to a point where the vertex has only one in edge and once we get to the vertex, no "out" edge that we have not traversed before which will break the TSP constraint. Assuming that each vertex has a count of its degree, this step takes $O(|V|)O(V)$ time.

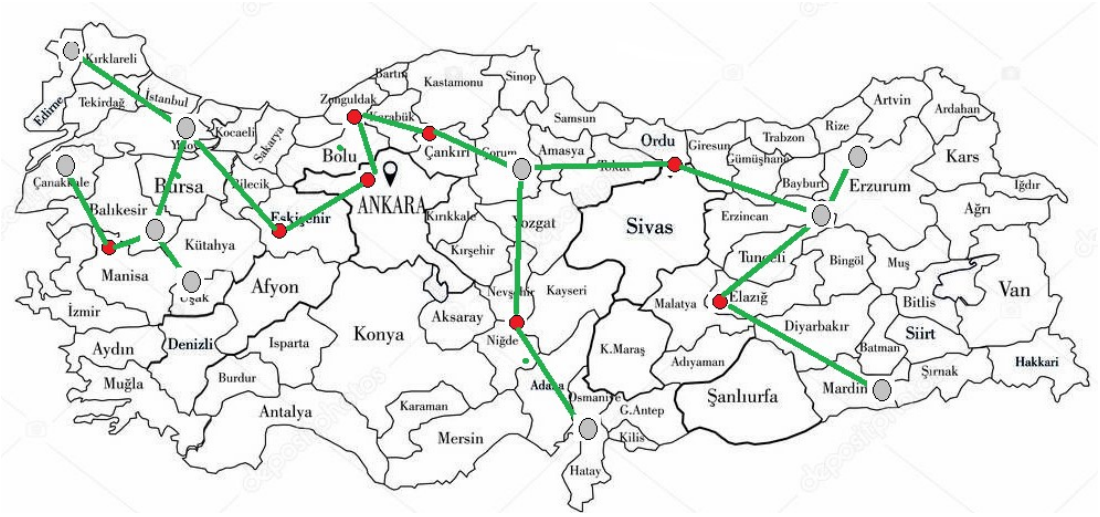


Figure 2.3: Grey Colored Vertices indicates the Odd Degree Vertices.

2.3. Step 3 » Finding Perfect Matching:

The algorithm's next step is to find a minimal perfect match of the odd degrees we discovered. This is done so that we may construct an Eulerian multigraph, as seen in the next step. We go through all the potential pathways connecting the graph's odd degree nodes to discover the combination of matchings that covers the shortest distance.

Finding the perfect match of a full graph necessitates the use of an algorithm with a runtime of $O(V^4)$. This is the most expensive step in Christofides' method, and it is the reason Christofides has a runtime of $O(V^4)$. $Figure 2.4$

Figure 2.4: Pairs of perfect vertex matches.

2.4. Step 4 » Finding Eulerian Circuit

The following step is to combine the matching and the minimum spanning tree from earlier to build an Eulerian multigraph, which is effectively a graph with an Eulerian tour and numerous edges between two identical nodes.

In this stage, we simply copy over all nodes and edges of each graph, removing any duplicate nodes (duplicate edges are fine!) and forming a new graph from it. The goal is to have all of the graph's odd degree nodes have an even degree so that we may continue on to the next stage, which is to discover the tour.

We are now going to make an Eulerian tour of the multigraph. In this stage, we recursively compute the path that visits all of the cities by traveling along each edge of the graph once.

In the $Figure 2.5$ you can see the tour of the multigraph is constructed. Edges added to the multigraph from the MST are highlighted in light-green while edges added from the matchings are highlighted in blue. Purple edges indicate places where there are two edges of the same length.

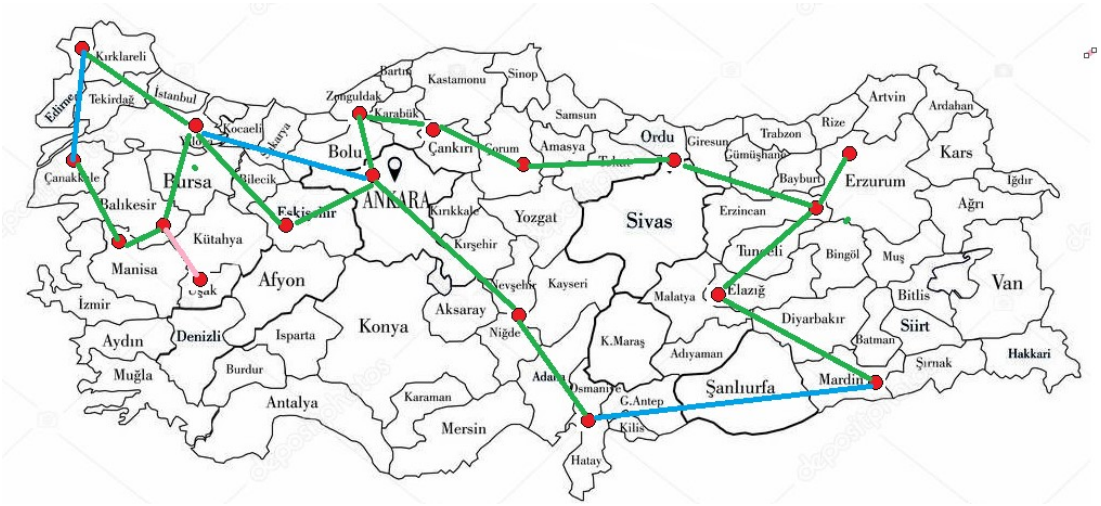


Figure 2.5: Eulerian Circuit.

2.5. Step 5 » Finding The Hamiltonian Circuit

The very last step of the algorithm is to remove any excess visits to cities we have by creating a Hamiltonian path. we do this by walking along the Eulerian tour from the last step and checking at each city whether it has already been visited. If it has, we skip that city and move on to the next one.The graph we have satisfies the triangle inequality so doing this can only reduce the length of our path. we can see the result on $Figure 2.6$

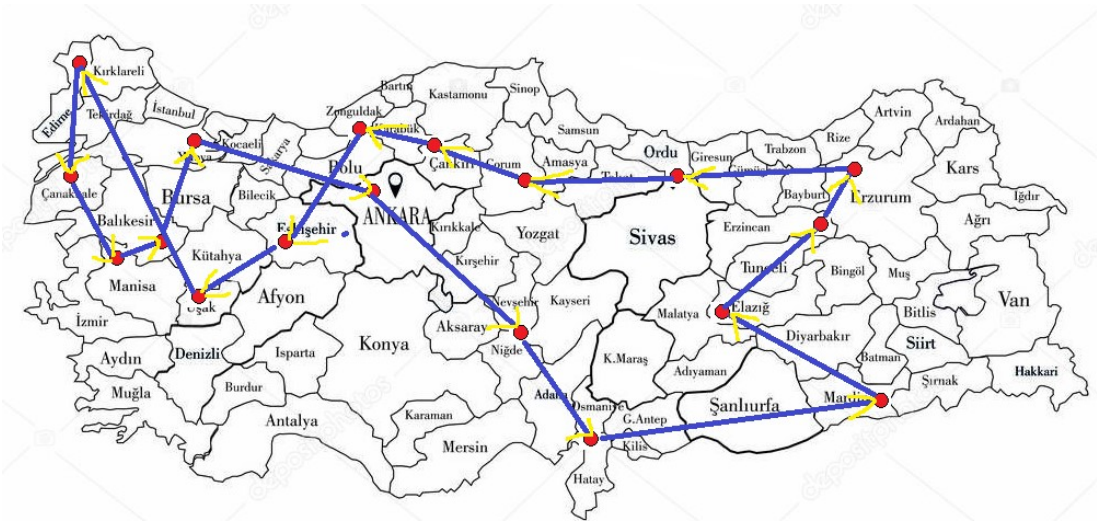


Figure 2.6: Hamiltonian Circuit.

3. CHRISTOFIDES FOR GRP

In above section we explained the christofide's approximation algorithm for TSP with ratio of $3/2$ with the optimal solution. here we will generalize christofides' algorithm in order to solve the GRP.

consider the general routing problem and the following graph $G \mid v', E' = (\{v \mid v \subset e \subset E'\} \cup V', E')$. At first we assume that this graph is connected. A GRP tour can only exists if there is no vertex $v \subset V'$ with degree greater than 2 in the graph $G|v', E'$. Therefore we consider the case that all vertices $v \subset V'$ have degree $0 < d(v) \leq 2$ and that the other have degree $d(v) > 0$.

A GRP tour requires k extra edges with odd degrees between vertices $\overline{\mathrm{V}}$. An edge of this type can alternatively be a route with intermediate vertices in $V \backslash V'$. To begin, we may build the shortest path over these edges for each pair of vertices with an odd degree. This may be accomplished by modifying a traditional shortest-path method. We acquire k alternative pairings of vertices with odd degrees and hence k pathways with intermediate vertices in $V \backslash V'$ by solving the weighted matching problem with these costs. Such a match cannot exist; imagine a three-length path where both edges belong to E' and the center vertex belongs to V'.

3.1. Theorem:

Let $G = (V, E)$ be a graph with weights $c : E \longrightarrow R^+$, let $V' \subset V$ and $E' \subset E$ and let $G|V', E'$ be the connected graph. Let $\overline{\mathrm{V}}$ be the vertices of odd degree and let $\overline{\mathrm{G}} = (\overline{\mathrm{V}}, \overline{\mathrm{E}})$ be a graph with an edge $v, w \in E$ iff $(v \neq w)$ and if there is a path from v to w in G with intermediate vertices in $V \backslash V'$.

(1) If one vertex $v \in V'$ has degree $d(v) > 2$ or if there exists no matching in G,then a GRP tour cannot exist.

(2) If all vertices $v \in V'$ hate degree $1 \leq d(v) \leq 2$ and if there is a matching in $\overline{\mathrm{G}}$, the below Algorithm 1 will generate an optimal solution.

3.2. Proof:

(1) If we have a vertex $v \in V'$ with $d(v) > 2$, we cannot generate a GRP tour which visits v exactly once. Now assume that $1 \leq d(v) \leq 2$ for each $v \in V'$. For the GRP tour we must enlarge the graph $G_{V',E'}$ by adding edges $e \in A$ into an eulerian multi-graph G_A with degree equal to 2 for each vertex $v \in V'$. To show that a matching in $\overline{\mathrm{G}}$ exists, take one vertex v of odd degree in $G_{V',E'}$ and generate a path to another vertex $w \neq v$ with $x \in \overline{\mathrm{V}}$ using only edges from A. Since the vertex v has odd degree, an incident edge $e \in A$ exist. If the other endpoint v' has odd degree, we have found such a vertex. Since in the other case the vertex v' has another incident edge from A, we can iterate this argument and find a path between vertices of $\overline{\mathrm{V}}$ using edges of A with intermediate vertices of $V \backslash V'$. Deleting this path in the graph G_A, only the endpoints of the path have odd degree. Therefore, we can construct a path between another pair of vertices in $\overline{\mathrm{V}}$ and can iterate this argument to get a matching in $\overline{\mathrm{G}}$.

(2) In the first part we have shown that the GRP tour can be generated by adding a matching between the vertices of $\overline{\mathrm{V}}$. Therefore, an optimal solution is given by a minimum weighted matching in $\overline{\mathrm{G}}$.

3.3. Algorithm:

Algorithm 1 Generalized Christofide's For Complete Weighted Graph

Require: $G(V, E)$, V', E'
Ensure: Minimum Cycle Path
$d(v) \leftarrow$ degree of vertices in $G' = G_{V',E'}$
$vert \leftarrow V \backslash V'$
while n in $d(v)$ **do**
  if n > 2 **then**
    return false
  end if
end while
Compute MST of G
Let $mst \leftarrow MST$ of G
Compute $odd - degree$ vertices of mst
Let $\overline{\mathrm{V}}$ be the vertices of odd degree
Compute the shortest path for each pair of $v, w \in \overline{\mathrm{V}}$ with $v \neq w$ in G with intermediate vertices in $V \backslash V'$
Compute $Minimum - Perfect - Matching$ for $\overline{\mathrm{V}}$
if No matching exists **then**
  return false
end if
get k pairs of vertices and add the corresponding path to G'
Compute the eulertour using depth-first-search.
$R \leftarrow$ Minimum cycle path.
$Return\ R$

3.4. Example

let us explain the above algorithm with an example: consider E' as the yellow line (edge) and V' as red dotted vertices in $figure$ 3.1 In the first step the edges (A, B), (B, C) and B, E with length 3 for the spanning tree are generated to connect the components, see $figure$ 3.2 After that we solve the matching problem with the odd-degree vertices $\overline{\mathrm{V}} = (A, B, D, E)$. The best solution takes the edges (A, B), (D, E). The generated graph can be seen in $figure$ 3.3

An eulertour is for example given by C, D, E, B, A, B, C. This tour must be shortened, because the vertex $B \in V'$ appeares twice. The reduced tour is C, D, E, B, A, C which is also given as graph see 3.4

Figure 3.1: Initial graph

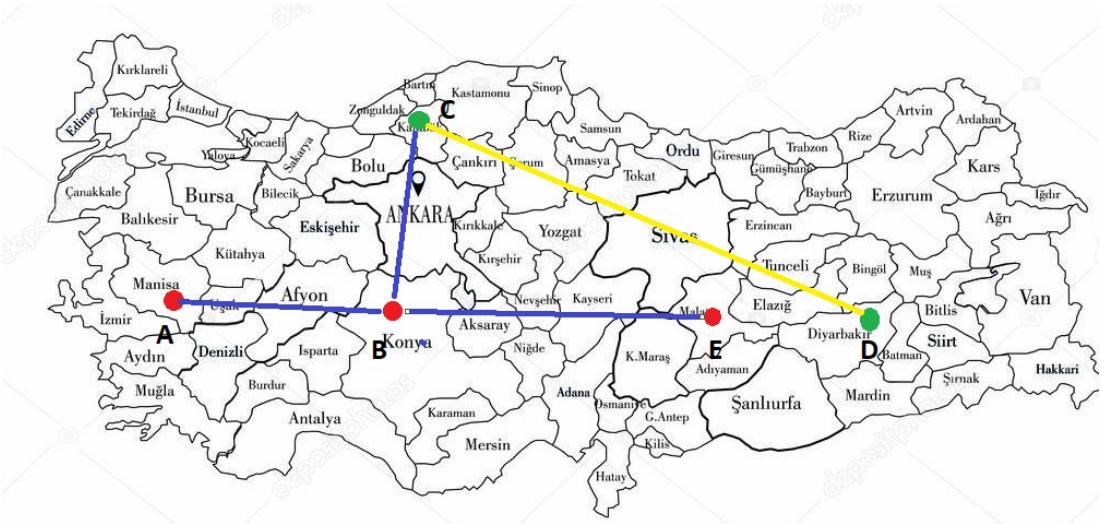


Figure 3.2: MST of the graph

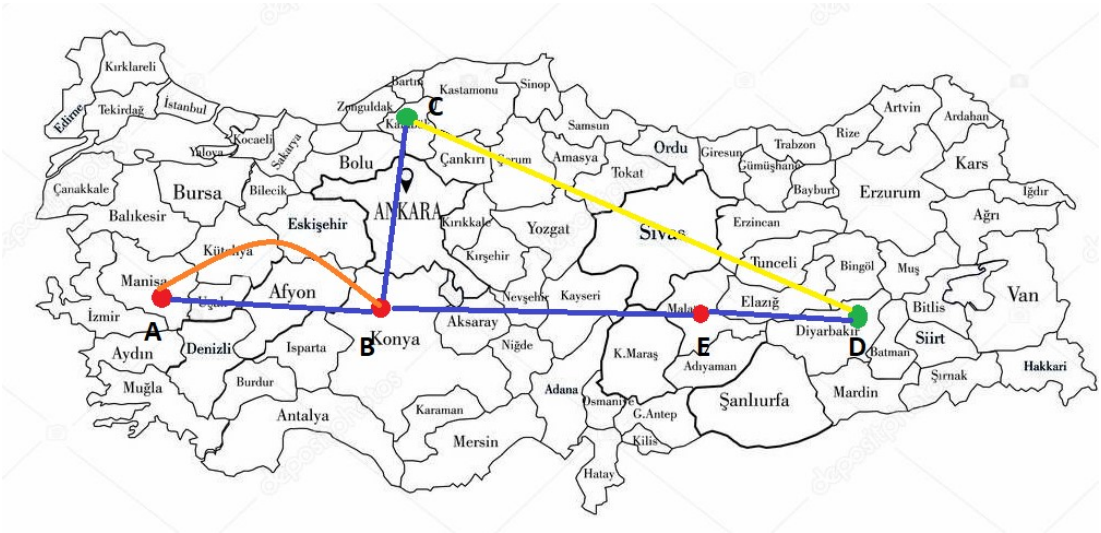


Figure 3.3: Eulerian graph

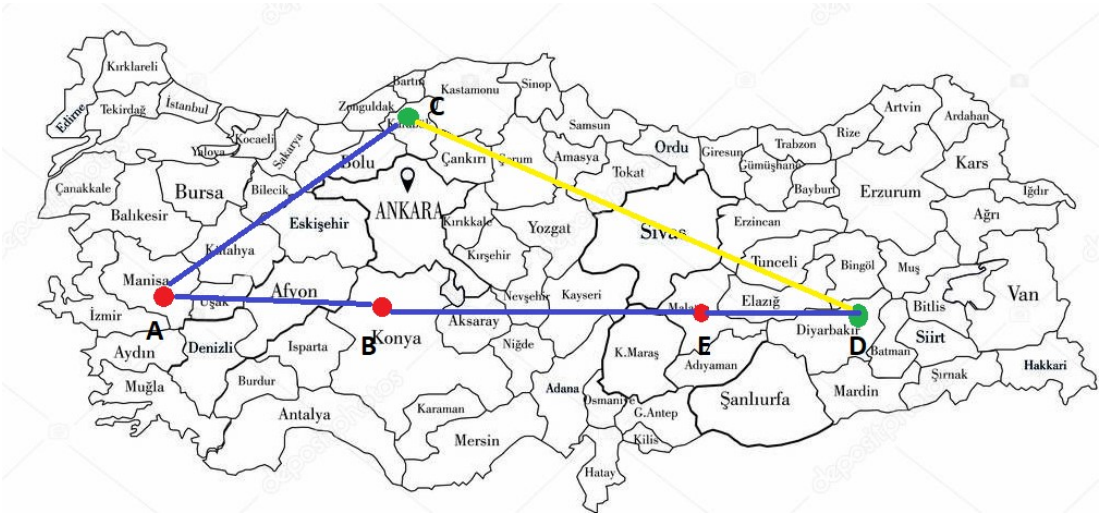


Figure 3.4: A GRP Tour

4. IMPLEMENTATION OF TSP

Christofides algorithm is one of the best algorithm in order to find the Travelling Salesmen Problem, here we have developed a simulation where we can test the christofides approximation algorithm with a naive approach in order to solve the TSP and compare the experiments and draw graphs with different inputs.

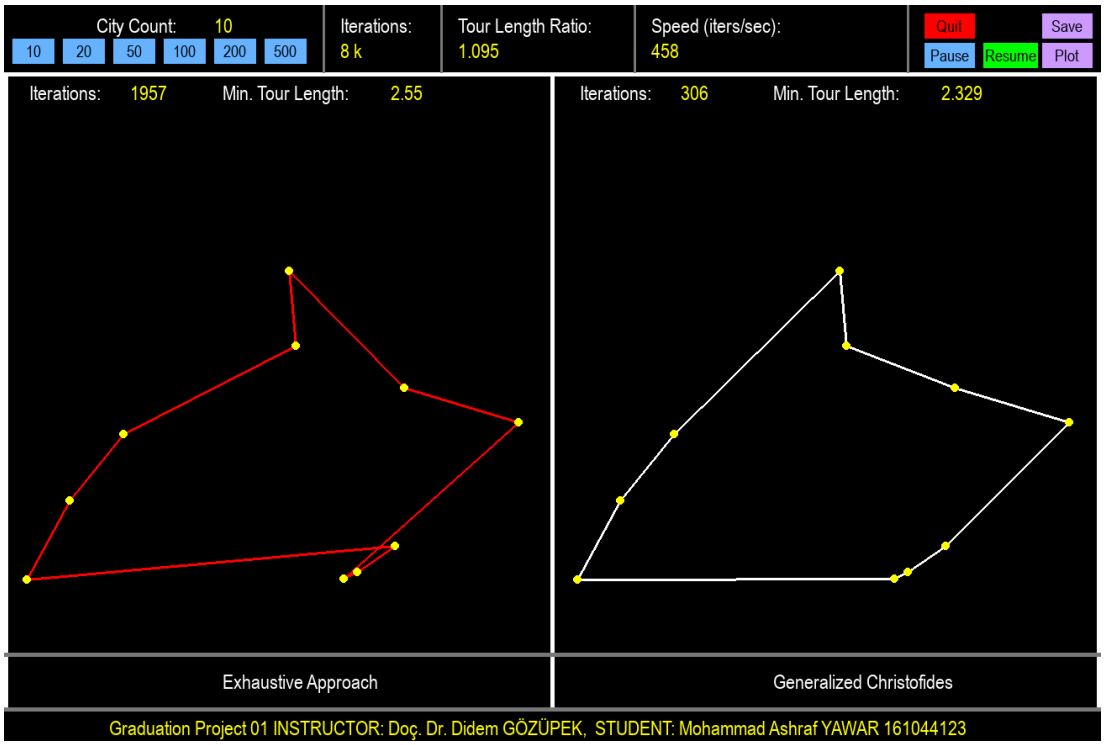


Figure 4.1: TSP with 10 cities (vertices)

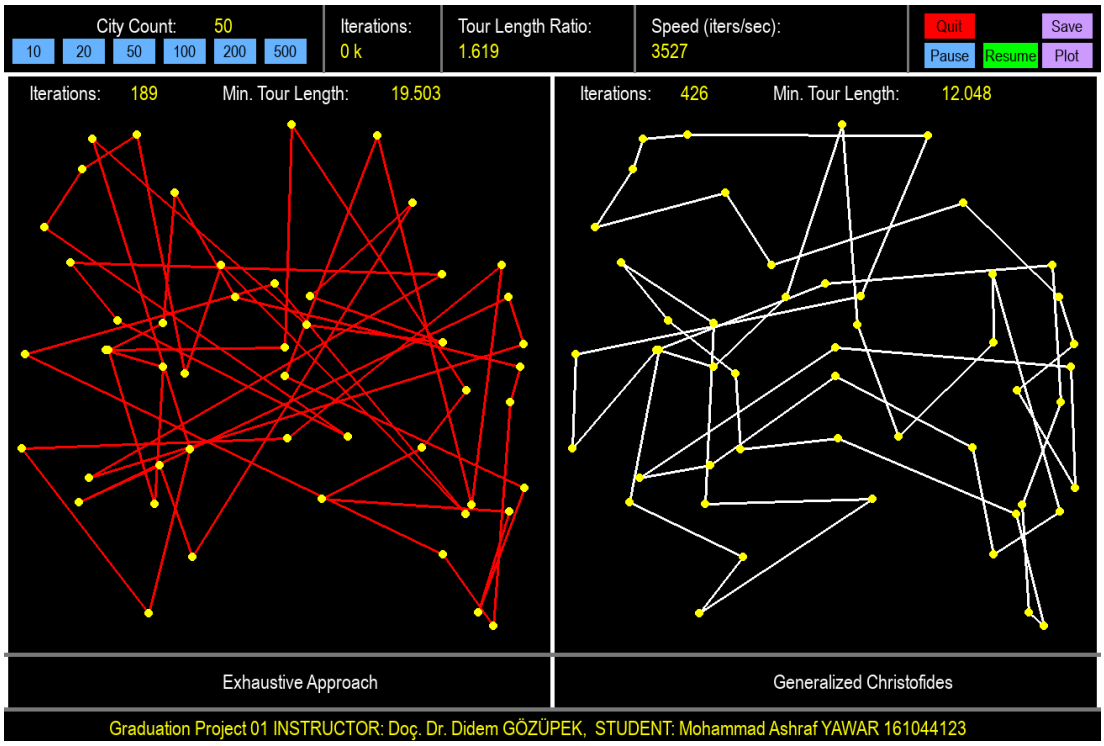


Figure 4.2: TSP with 50 cities (vertices)

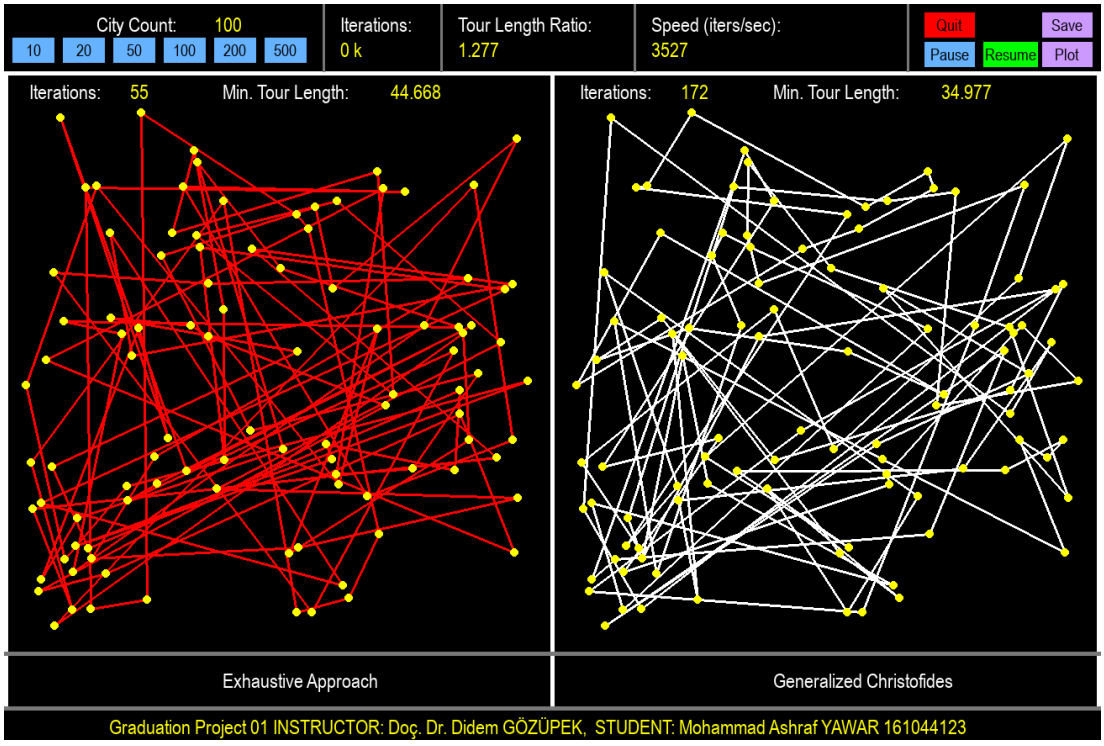


Figure 4.3: TSP with 100 cities (vertices)

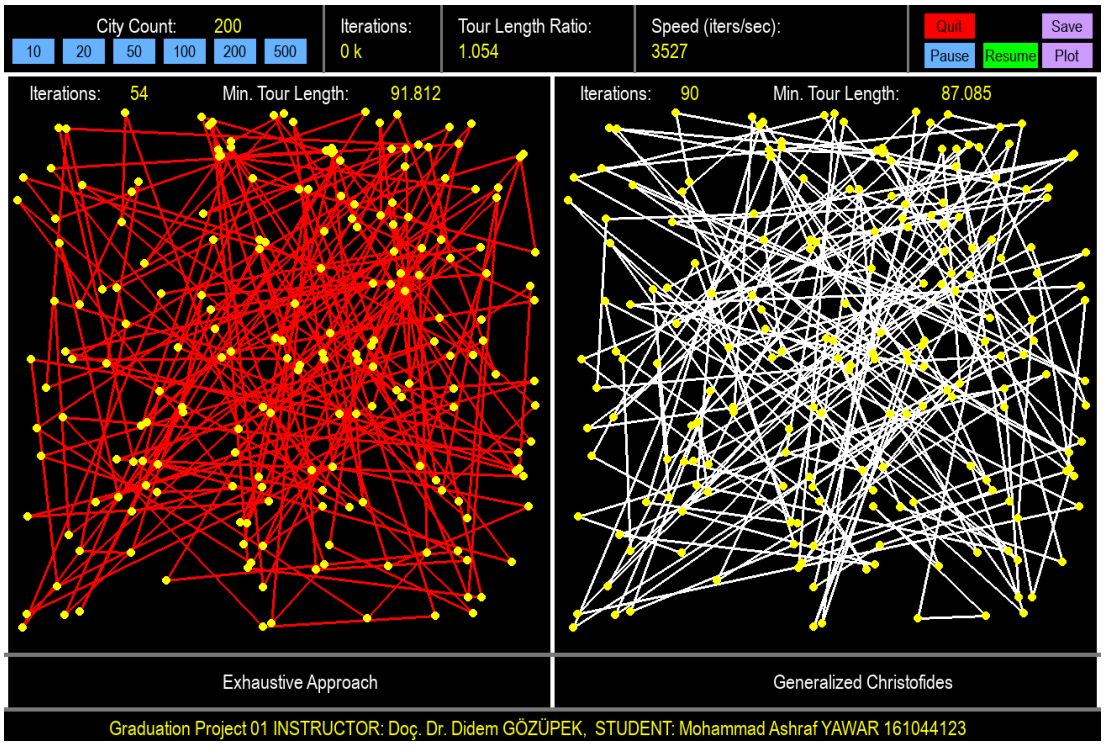


Figure 4.4: TSP with 200 cities (vertices)

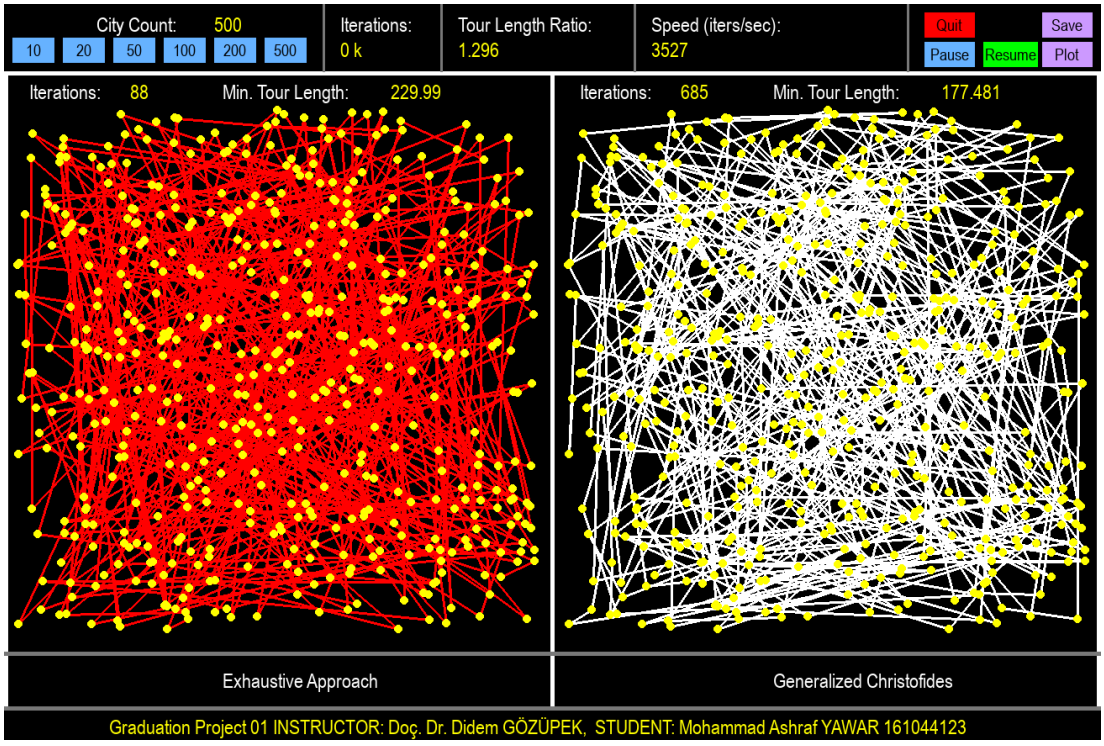
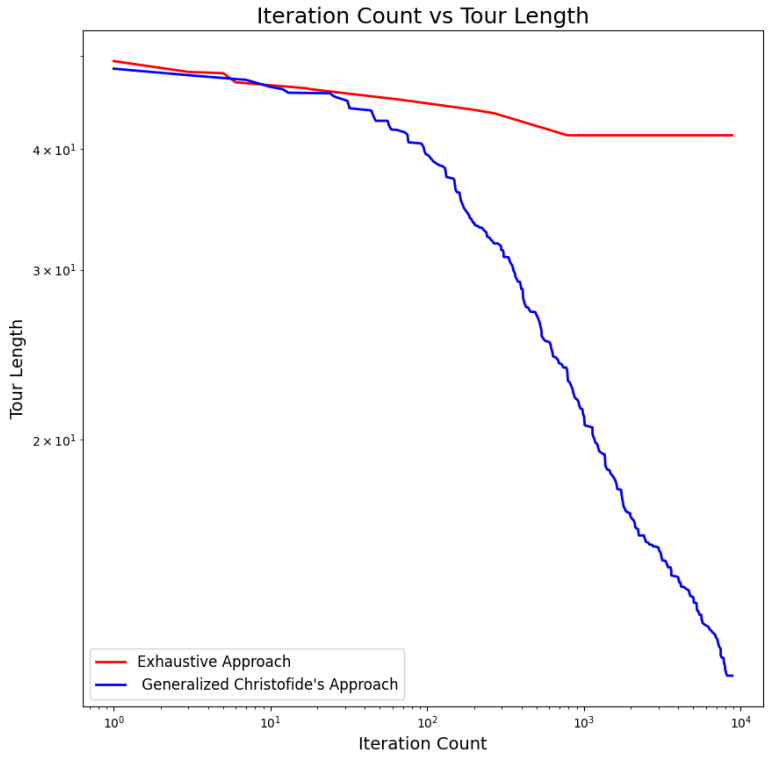
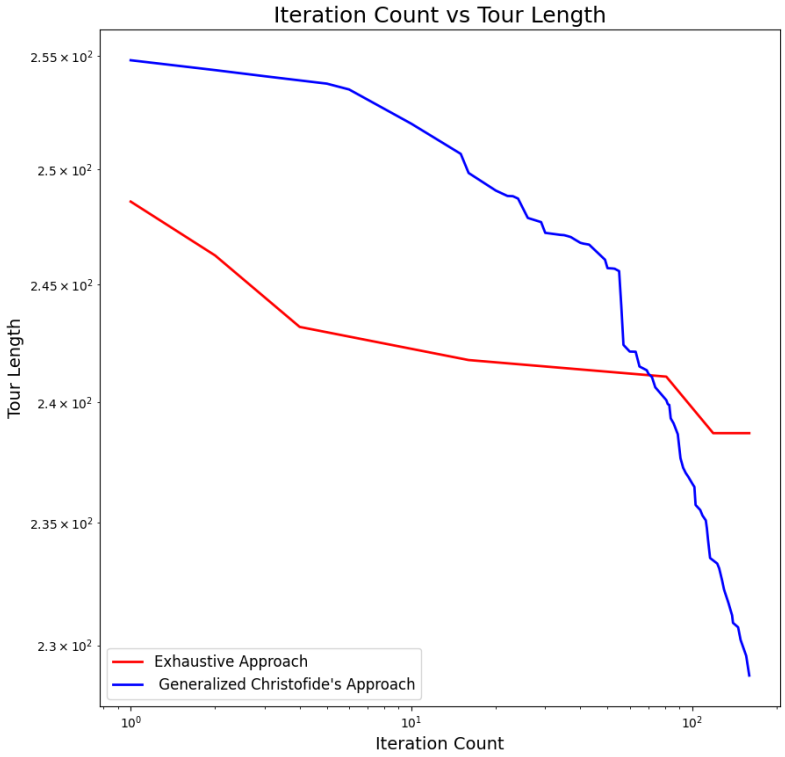


Figure 4.5: TSP with 500 cities (vertices)


Iteration Count vs Tour Length
Tour Length
Iteration Count
Exhaustive Approach
Generalized Christofide's Approach


Iteration Count vs Tour Length
Tour Length
Iteration Count
Exhaustive Approach
Generalized Christofide's Approach

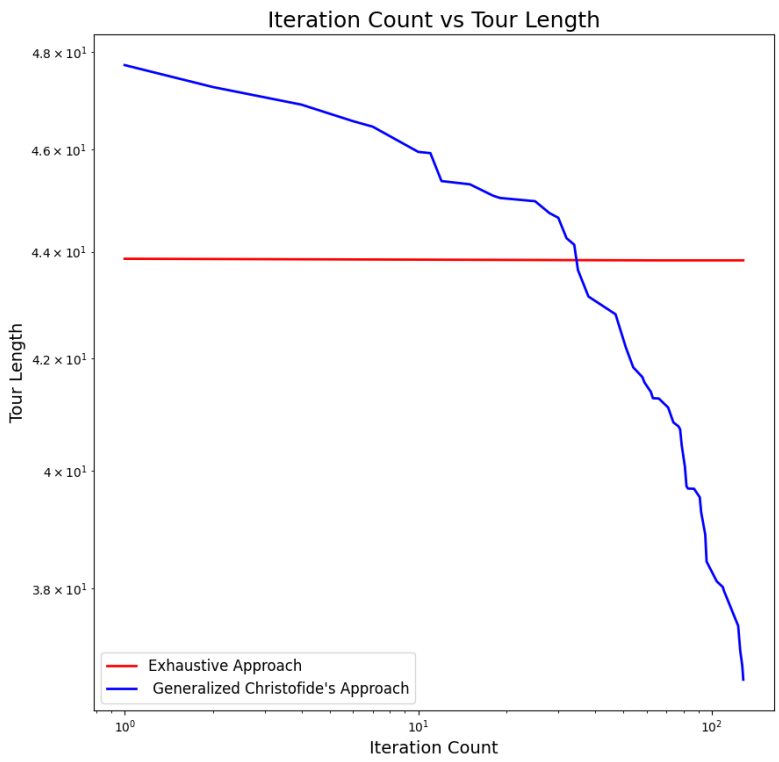


Figure 4.6: Iteration-Count vs Tour-Length (vertices)

Mohammad Ashraf Yawar

SOFTWARE ENGINEER & WEB DEVELOPER

CONTACT

Darıca/Kocaeli

5349228234

mayawar@gtu.edu.tr

GoToGitHub

GoToMyWebSite

GoToLinkedIn

INTERESTS

- FullStack Web Development
- Software Development
- NLP, Machine Learning

SKILLS

- HTML5
- CSS3
- SAAS
- JAVASCRIPT
- ES5,ES6
- C,C++
- C#
- PYTHON
- JAVA
- COMMON LISP
- MIPS ASSEMBLY
- VERILOG HDL
- PROLOG
- AJAX
- REACT
- JSON, XML
- BOOTSTRAP
- JQUERY
- ORACLE SQL
- Angular 12
- Spring-Boot

TECHNOLOGIES

- Bitbucket,SourceTree
- MySql Workbench
- SOUP UI
- JAVA SWING
- ANDROID STUDIO
- VC(Git)
- CMS
- DJANGO framework
- ADOBE XD
- JIRA ATLASSIAN

CAREER OBJECTIVE

To be able to work for an encouraging and stable company that will assist me in developing, improving, and obtaining the necessary skills in order to become the best engineer possible.

EDUCATION

UNIVERSITY

4th Year Computer Science Student at Gebze Technical University

IMPORTANT SUBJECTS:

Data Structures And Algorithms,

Algorithm Design,

Computer Organization ,

Software Engineering ,

Object Oriented Programming,

Functional Programming,

Design Patterns,

Database Management Systems,

Computer Networks.

EXPERIENCE

SOFTWARE ENG, WEB DEVELOPER

2021-Present

Developing Web Applications Using, Natural Language Proccessing, Angular12,Spring-Boot,Mysql Database.
Go To Project

GUI DESIGNER

2020 - Airport Management System Using Java And Java Swing

We developed terminal based and GUI based Airport Managment System Using java and java Swing for an artificial airport . Go To Project ☺

CMS MANAGER ON WORDPRESS

2019 | As a Freelancer

Worked as content management system Manager on wordpress to maintain a company's website .

ANDROID APP DEVELOPER & WEB DEVELOPER

2020 – BALL BOUNCER PROJECT

We developed hardware and simulation system to balance and bounce the ball on a plate using image processing , webservices , 3d modeling and more . Go To Project ☺

FULL STACK DEVELOPER & SCRUM MASTER (AGILE TEAM)

2020 – BOOKWORM PROJECT

in this project, we developed a full-stack web app which recommends books and we used the KNN machine learning model to predict books according to users' inputs and suggest a list of books. Go To Project

HONORS AND AWARDS

Yurtdışı Türkler ve Akraba Topluluklar Başkanlığı (***YTB***) Full scholarship (2016-To date).

SPOKEN LANGUAGES

- English
- Turkish
- Persian
- Urdu
- Turkmany
- Uzbeky

CURRENTLY WORKING OR STUDING :

- Java Spring
- Systems Programming
- Angular
- REST API
- NLP, Machine Learning

REFFERENCE LINKS

https://www.sciencedirect.com/science/article/abs/pii/002001909290161N

https://link.springer.com/article/10.1007/s10878-021-00772-8

https://en.wikipedia.org/wiki/Christofides$_algorithm$

$https://jlmartin.ku.edu/courses/math105-F11/Lectures/chapter6-part3.pdf$

$https://www.researchgate.net/profile/Klaus-Jansen$

$https://medium.com/musoc17-visualization-of-popular-algorithms/travelling-salesman-problem-e3b98653b11a$

$https://scale.iti.kit.edu/_media/resources/supplemental/fct_talk.pdf$

$shorturl.at/knxCW$

$shorturl.at/gCIW3$

$shorturl.at/lBLOS$

$https://www.geeksforgeeks.org/hamiltonian-cycle-backtracking-6/$

$https://arxiv.org/pdf/1911.02453.pdf$