**“Comparative Study of Ant Colony Optimization Algorithm**

**on**

**Travelling Salesman Problem ”**

***A***

***Project Report***

*submitted in partial fulfillment of the*

*requirements for the award of the degree of*

**BACHELOR OF TECHNOLOGY**

**in**

**COMPUTER SCIENCE & ENGINEERING**

**With**

**Specialization in**

**Internet of Things and Smart Cities**

**by**

|  |  |
| --- | --- |
| **Name** | **Roll No.** |
| **Harshit Gupta** | **R164217020** |
| **Divyank Kargeti** | **R164217017** |

***under the guidance of***

**Dr. Adarsh Kumar**

**Assistant Professor**

**Department of Systemics**

**SCHOOL OF COMPUTER SCIENCE**

**UNIVERSITY OF PETROLEUM & ENERGY STUDIES**

**Bidholi, Via Prem Nagar, Dehradun, December – 2019**

**CANDIDATE’S DECLARATION**

We hereby certify that the project work entitled **“ Comparative Study of Ant Colony Optimization Algorithm on Travelling Salesman Problem ”** in partial fulfilment of the requirements for the award of the Degree of BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE AND ENGINEERING with specialization in **Internet of Things and Smart Cities**  and submitted to the Department of Systemics at School of Computer Science, University of Petroleum & Energy Studies, Dehradun, is an authentic record of my/ our work carried out during a period from **August, 2019** to **December, 2019** under the supervision of **Mr. Adarsh Kumar , Assistant Professor, Department of Systemics.**

The matter presented in this project has not been submitted by me/ us for the award of any other degree of this or any other University.

**Harshit Gupta R164217020**

**Divyank Kargeti R164217017**

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Date: December 18, 2019

**Dr. Neelu Jyoti Ahuja Dr.Adarsh Kumar**

Head of Department Project Guide

Department of Systemics

School of Computer Science

University of Petroleum and Energy Studies, Dehradun

**ACKNOWLEDGEMENT**

We wish to express our deep gratitude to our guide **Dr. Adarsh Kumar**, for all advice, encouragement and constant support he has given us through out our project work. This work would not have been possible without his support and valuable suggestions.

We sincerely thank to our respected Program Head of the Department, **Dr. Neelu Jyoti Ahuja**, for his great support in doing our **project** at **SoCS**.

We are also grateful to **Mr. Manish Prateek, Professor and Director, SoCS** for giving us the necessary facilities to carry out our project work successfully.

We would like to thank all our **friends** for their help and constructive criticism during our project work. Finally we have no words to express our sincere gratitude to our **parents** who have shown us this world and for every support they have given us.

|  |  |  |
| --- | --- | --- |
| **Name** | **Harshit Gupta** | **Divyank Kargeti** |
| **Roll No.** | **R164217020** | **R164217017** |

**ABSTRACT**

1. commerce organizations have a separate department to handle logistics for timely delivery of products ordered by their customers. Each delivery has a separate customization and delivery address. The process, after the order has been accepted and is to be dispatched, begins with collection of goods from the warehouse and doorstep delivery via optimized path calculation, factoring in the essential components like fuel economy, load, and quantity. Cash on delivery and pre-processed payment options are considered, and automatic division to appropriate mode of transport for the goods is carried out. Hence, this automation leads to profitability, and reduction in cost of resources being used. An accurate way to calculate this path is via the TravellingSalesman Problem by W.R. Hamilton.

**Keywords:** Travelling-Salesman, Algorithms, Hamiltonian Cycle, E-commerce, Optimization.

**TABLE OF CONTENTS**

**Figure Page No**

**INTRODUCTION ...........................................……….. 6**

**PROBLEM STATEMENT ............................................6**

**PROJECT OBJECTIVES ..........................................…8**

**METHODOLOGY .................................……………….8**

**IMPLEMENTATION….......................…………………9 MODULES..............................….....................…………..10 PSEUDOCODE ........................………............................ 12**

**ALGORITHM USED ........................................................12**

**OUTPUT SCREENS ..........................................................13**

**RESULT ANALYSIS ...............……………………..........16**

**OTHER ALGORITHMS .............................……………..16**

**CONCLUSION AND FUTURE SCOPE ....................…..17**

**SYSTEM REQUIREMENTS .............……………………17**

**SCHEDULE(PERT CHART) .............................................18**

**REFERENCES AND BIBLIOGRAPH……....................…18**

**APPENDIX 1: CODE ..........................…..............................19**

**INTRODUCTION**

E-commerce organizations like Amazon are known to deliver their products with efficiency, and in a good state. It is because of the logistical support they establish to cover their regions. There are two main ways to conduct this task:

1. By establishing a delivery system of their own
2. 2. Outsourcing the logistics services

Each way has its own advantages and disadvantages, but the common ground for them is optimization of the route they will choose to have day-to-day deliveries being made at the doorstep of the customer. It is easy to understand that reduced costing with improved quality is one of the main aims of each organization. Since delivery includes manpower, resources like fuel, and transport maintenance, a major part of funds goes to logistics. With ever improving competition to be the best supplier of goods, effective systems become a need of the hour to facilitate this venture.

In the travelling-salesman problem, which is closely related to the Hamiltonian cycle problem, a salesman must visit n cities. Modeling the problem as a complete graph with n vertices, we can say that the salesman wishes to make a tour, visiting each city exactly once and finishing at the city he starts from. The salesman incurs a nonnegative integer cost c(A,B) to travel from city A to city B, and he wishes for this cost to be minimum, where the total cost is the sum of the individual costs along the edges of the tour.

Given the functionalities of programming language C, a possible implementation is with the defined complexity of O (n2\*2n).

**PROBLEM STATEMENT**

The main challenge in doing this project is to optimize the cost of the logistics system by devising an algorithm which involves minimum space and time. We will take orders from multiple users, from a specific city divided into sectors, from a list of products provided by us, and take the respective addresses to calculate the optimized path of delivery.

**PROJECT OBJECTIVES**

**OBJECTIVE 1:** To have an understanding of algorithm used, its complexity analysis, and implementation for minimizing the cost of transportation.

**OBJECTIVE 2:** To understand the working of delivery routing systems used by the logistics staff in an e-commerce organization, and to demonstrate the methodology used by this algorithm.

**ALGORITHM USED**

**The Bellman-Held-Karp**

It belongs to Dynamic Programming in TSP, and is the most efficient algorithm to be implemented for solving it:

 Take one starting vertex x arbitrarily  For the set of vertices S belonging to V\{S}, vertex v is in S, let: o B(S,V) = minimum length of path, that: o Starts in S o Visits S and all vertices in S (and no other vertices) o Ends in V

1. B({v},v) = d(s,v) 2. For j = 2 to |V| - 1: 1. For all sets S, |S| = j for all v belonging to S:

13

2. B(s,v) = minw-S\(v){B(S\{V},w}) + d(w,v)} 3. Return min{B(V\{s},v) + d(v,s) | v belongs to V}

The total running time is therefore O (n2\*2n).

**PSEUDOCODE**

The pseudocode for Bellman-Held-Karp is as follows:

function algorithm TSP (G, n)

for k := 2 to n do

C({k}, k) := d1,k

end for

for s := 2 to n-1 do

for all S ⊆ {2, . . . , n}, |S| = s do

for all k ∈ S do

C(S, k) := minm≠k, m∈S [C(S\{k}, m) + dm,k]

end for

end for

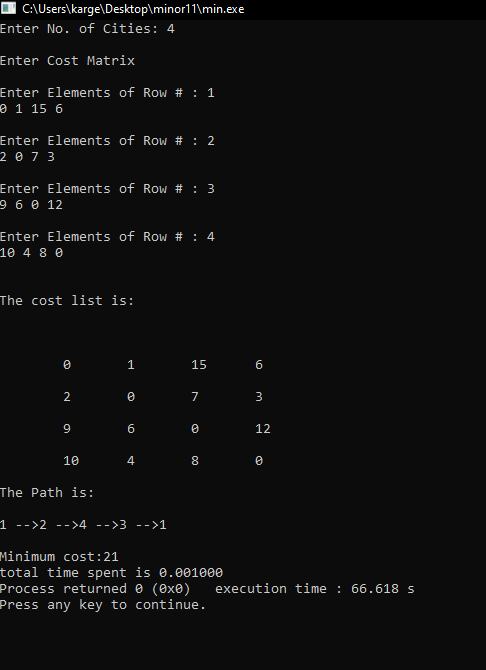
end for

opt := mink≠1 [C({2, 3, . . . , n}, k) + dk,1]

return (opt)

End

**OUTPUT SCREEN**



**Ant Colony Optimizatio(ACO)**

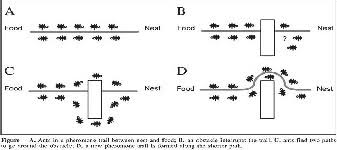
In the real world, ants (initially) wander randomly, and upon finding food return to their colony while laying down pheromone trails. If other ants find such a path, they are likely not to keep traveling at random, but instead follow the trail laid by earlier ants, returning and reinforcing it if they eventually find food.

Over time, however, the pheromone trail starts to evaporate, thus reducing its attractive strength. The more time it takes for an ant to travel down the path and back again, the more time the pheromones have to evaporate.

A short path, by comparison, gets marched over faster, and thus the pheromone density remains high .

**Shortest Route**

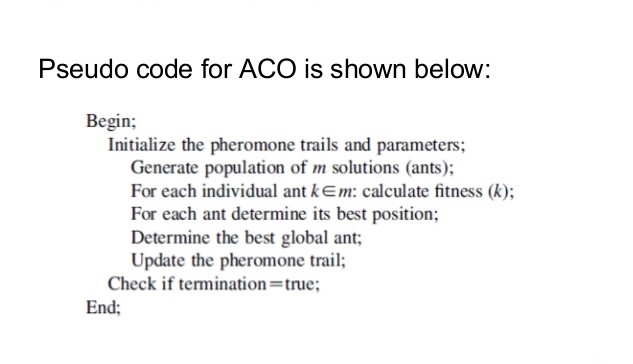
Shortest route is found using pheromone trails which ants deposit whenever they travel, as a form of indirect communication.



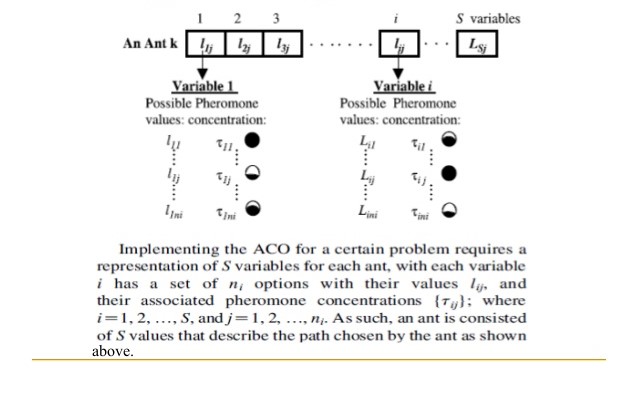
**ACO Algorithm**

The process starts by generating m random ants (solution).

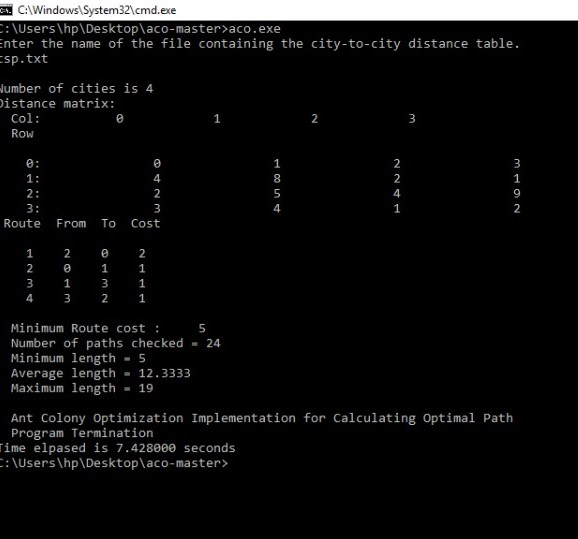
* An ant k (k=1,2…….,m) represents a solution string, with a selected value for each variable.
* An ant is evaluated according to an objective function.
* Accordingly, pheromone concentration associated with each possible route( variable value) is changed in a way to reinforce good solutions as follows:



**ACO algorithm Implementation**



**OUTPUT for ACO**



# CONCLUSION AND FUTURE SCOPE

The project could be updated for the additions of Address Indexing and Latitude and Longitude

-based addressing system for the users using web-based solutions. The problem could also be implemented for multiple vehicle routing, along with its allocation and implementation. The project has successfully implemented Traveling Salesman Problem using Bellman-Held-Karp Algorithm and Ant Colony Algorithm.

# SYSTEM REQUIREMENTS

**SOFTWARE REQUIREMENTS**

Operating System : MS-DOS 2.0 or higher

Programming Language : C

Compiler : GCC

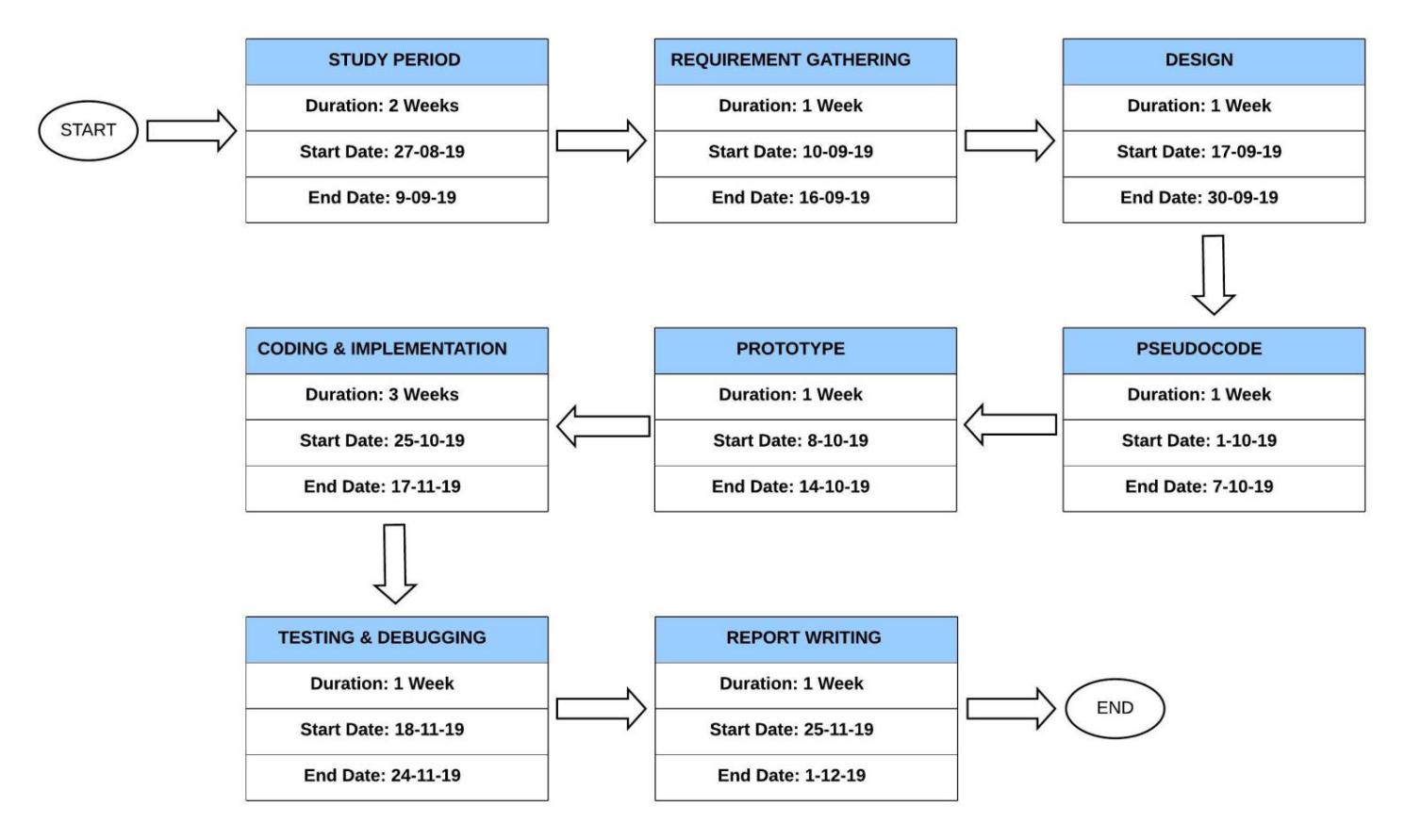
**HARDWARE REQUIREMENTS**

Processor : Pentium IV or higher

Disk Drive : Floppy or Hard Disk Drive

RAM : 512 MB or higher

# SCHEDULE (PERT CHART)



**CODE**

#include<stdio.h>

#include<time.h>

void get\_table();

int mindist(int c);

float mincost(int city);

void display\_result();

int node\_table[10][10];

int traversed[10];

int n,cost=0;

void get\_table(){

int i,j;

printf("Enter No. of Cities: ");

scanf("%d",&n);

printf("\nEnter the distance(cost) between nodes\n");

for(i=0;i < n;i++){

printf("\nEnter Elements of Row # : %d\n",i+1);

for( j=0;j < n;j++){

scanf("%d",&node\_table[i][j]);

traversed[i]=0;

}

printf("\n\nThe cost list is:\n\n");

for( i=0;i < n;i++){

printf("\n\n");

for(j=0;j < n;j++){

printf("\t%d",node\_table[i][j]);

}

}

int mindist(int c){

int i,nextcity=1000;

int min=1000,costmin;

for(i=0; i<n; i++){

if((node\_table[c][i]!=0)&&(traversed[i]==0)){

if(node\_table[i][c] + node\_table[c][i] < min){

0min = node\_table[i][0]+node\_table[c][i];

costmin = node\_table[c][i];

nextcity=i;

}

}

if(min!=1000)

cost+=costmin;

return nextcity;

}

float mincost(int city){

double time\_spent = 0.0;

clock\_t begin = clock();

int i,ncity;

traversed[city]=1;

printf("%d -->",city+1);

ncity=mindist(city);

if(ncity==1000){

ncity=0;

printf("%d",ncity+1);

cost+=node\_table[city][ncity];

return;

}

mincost(ncity);

clock\_t end = clock();

time\_spent += (double)(end - begin) / CLOCKS\_PER\_SEC;

return time\_spent;

}

void display\_result(){

printf("\n\nMinimum cost:");

printf("%d",cost);

}

int main(){

float time\_spent;

get\_table();

printf("\n\nThe Path is:\n\n");

time\_spent=mincost(0);

display\_result();

      printf("total time spent is %f", time\_spent);

return 0;

}

# 

# REFERENCES AND BIBLIOGRAPHY

[1] T.H. Cormen, C.E. Leiserson, R.R. Rivest, C. Stein. *Introduction to Algorithms: Third Edition.* The MIT Press, Cambridge, Massachusetts, London, England

[2] Cook, William J. *In Pursuit of the Traveling Salesman: Mathematics at the Limit of Computation*. Princeton, NJ: Princeton UP, 2012 [Print]

[3] C. Brucato. “The Travelling Salesman Problem,” 2013, pp. 10-14

[4]C.Woodford.“ECommerce.”Internet:https://www.explainthatstuff.com/ecommerce.html, Sept. 14, 2016 [Sept. 06, 2018]

[5] Y. Yu, X. Wang, R.Y. Zhong, G.Q. Huang. “E-commerce Logistics in Supply Chain Management: Practice Perspective,” 2016, pp. 179-185

[6] K.K. Aggarwal, Y. Singh. *Software Engineering: Third Edition.* New Age International Publishers, Dariya Ganj, New Delhi, Delhi

[7]S.K.Pal.“SoftwareEngineering|IterativeWaterfallModel.”Internet:https://www.geeksforgeeks.org/software-engineering-iterative-waterfall-model, Aug. 26, 2016 [Sept. 06, 2018]

[8]Wikipedia.“HeldKarpAlgorithm”Internet:https://en.wikipedia.org/wiki/Held%E2%80%93Karp\_algorithm#Dynamic\_programming\_approach, Apr. 1, 2018 [Oct. 28, 2018]

[9] Q. Wang. “Improving TSP Problem Allowing Multiple Vehicle Distribution,” 2016

[10] M. Assaf, M. Ndiaye. “Multi Travelling Salesman Problem Foundation,” 2017, pp. 292-295

# 

# 