Travelling Salesman Problem using hybrid ACO and Cuckoo Search algorithm

A dissertation report submitted

By

Sandeep Kumar

Reg-Id: 11310901

To

Department of Computer Science & Engineering

In partial fulfilment of the Requirement for the Award of the Degree

Master of Technology in CSE

Under the supervision of

Mr. Sudhanshu P. Tiwari

(Assistant Professor)

Lovely Professional University

(April 2015)

LOVELY PROFESSIONAL UNIVERSITY

Transforming Education, Transforming India

School of: Computer Science and Engineering

DISSERTATION TOPIC APPROVAL PERFORMA

Name of the Student: Sandeep Kumar   Registration No: 11310901

Batch: 2013-15   Roll No. B69

Session: 2014-15   Parent Section: K2305

Details of Supervisor: Designation: Ass. Professor

Name: Sudhanshu Prakash Tiwari   Qualification: M.Tech

U.ID: 15813   Research Experience: 3 Years

SPECIALIZATION AREA: Soft Computing (Optimization Techniques) (pick from list of provided specialization areas by DAA)

PROPOSED TOPICS

1. ~~TSP using advance Ant Colony Optimization~~ Hybrid Ant Colony optimization and cuckoo search algorithm for Travelling Problem
2. ....
3. ....

Signature of Supervisor

PAC Remarks:

Topic '1' is approved

APPROVAL OF PAC CHAIRPERSON: Signature: Date: 26/9/14

*Supervisor should finally encircle one topic out of three proposed topics and put up for approval before Project Approval Committee (PAC)

*Original copy of this format after PAC approval will be retained by the student and must be attached in the Project/Dissertation final report.

*One copy to be submitted to Supervisor.

Abstract

Ant colony optimization (ACO) algorithm is an artificial intelligence based optimization technique that works on the pheromone-laying behaviour of ants. This algorithm can be used to get solutions of highly complex problems that search the optimal path through a graph. Travelling Salesman Problem is an important NP-hard optimization problem; its popularity is due to the fact that Travelling Salesman Problem is easy to formulate, difficult to solve and has a large number of applications, in which we have to cover each city exactly once and return back to starting city. Many existing algorithms were used to solve Travelling Salesman Problem. Our approach will provide enhanced algorithm using Ant colony optimization and avoids problems related to Ant colony optimization. A comparative study will be done to show that our approach is better than previous algorithms.

CERTIFICATE

This is certifying that **Sandeep Kumar** has completed M. Tech dissertation-II titled **Travelling Salesman problem using Hybrid Ant colony optimization and Cuckoo search algorithm** under my guidance and supervision. To the best of my knowledge, the present work is the result of her original investigation and study. No part of the dissertation has ever been submitted for any other degree or diploma.

The dissertation is fit for the submission and the partial fulfilment of the conditions for the award of M. Tech Computer Science & Engineering.

Date:

Signature of Advisor

Name : Mr. Sudhanshu P. Tiwari

UID: 15813

DECLARATION

I hereby declare that the dissertation entitled, **"Travelling Salesman Problem using hybrid Ant Colony optimization and Cuckoo search algorithm"** submitted for the M. Tech Degree is entirely my original work and all ideas and references have been duly acknowledged. It does not contain any work for the award of any other degree or diploma.

Investigator

Date: Reg. No-11310901

ACKNOWLEDGEMENT

I am deeply grateful to my mentor, **Mr. Sudhanshu Prakash Tiwari**, for his guidance, patience and support. I consider myself very fortunate for being able to work with a very sincere, considerate and encouraging professor like him. Without his offering to accomplish this research, I would not be able to do my dissertation-II work at Lovely professional University. I'm much obliged to my friends. Their enlightening suggestions and encouragements made me feel I was not isolated in my research.

---Sandeep Kumar

Table of Content

List of Figures

Chapter 1
INTRODUCTION

1.1 Optimization Algorithm

Optimization algorithm [1] is an algorithm in which we minimize the cost and effort required in the problem. Here effort shows no. of iteration and cost shows time required to execute the algorithm. The representation of Optimization problem is as follows:

Input: f : A⟶R**,** where f is a function and R is set of real numbers.

Output: We try to find out an element x_1 in A so that

$f(x_1) \leq f(x)$ for all x in A (represents minimization)

$f(x_1) \geq f(x)$ for all x in A (represents maximization)

A combinatorial optimization problem [2] is that problem in which shows an optimal ordering and in which selection of options is desired.

1.2 Ant Colony Optimization (ACO)

Ant Colony Optimization (ACO) [1, 2, 3] algorithm is a metaheuristic optimization technique [1, 2] that is used to minimize the cost and maximize the efficiency of an optimization problem. ACO is a nature based computational technique for which main basis is behaviour of ants. Ant colony optimization algorithm is inspired food searching behaviour of real ant colonies. This natural behaviour of artificial ant colonies does not search the exact solutions but provides approximate solutions to continuous and discrete optimization problems [7], like travelling salesman problem, vehicle routing problem, job scheduling problem etc. The main thing that works in ACO is simulated pheromones [6, 9] that attract ants to best trail in the graph because they choose the path based on the pheromone density. Ants are naturally blind, deaf & dumb so when they move, leave pheromone (chemical material) in its path. On the basis of this pheromone density other ants follow the path. ACO uses four main parameters that are influence of pheromone on direction (α) [1, 6], influence of adjacent node distance (β) [1, 4], pheromone evaporation coefficient (ρ) [2, 4, 5] and pheromone depositing factor (Q) [5, 6, 8]. This algorithm updates the trail on the basis of current pheromone density.

In the given figure the behaviour of ants has been shown. In the part (a) ants are normally going from nests to food source. In the subsequent parts of the diagram when any obstacle comes in the path of ants then ant chooses path based upon pheromone density. In the shorter distance pheromone density will increase fast as compare to longer distance with same number of ants. Since pheromone density will be higher in the shorter path so ants will choose finally this path for going from nest to food source. Given diagram shows it more clearly.

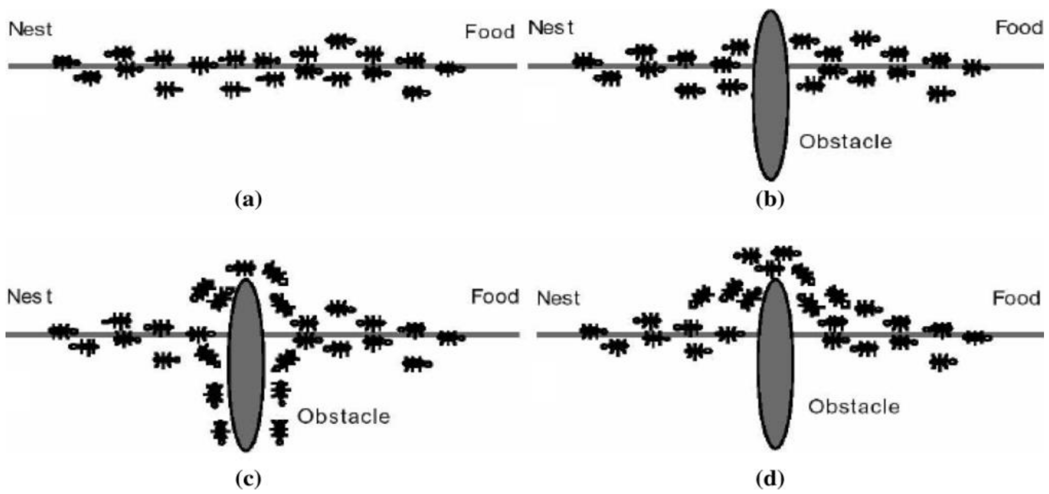


Fig1.1-Ants foraging behaviour

Following flow chart shows working of ACO algorithm-

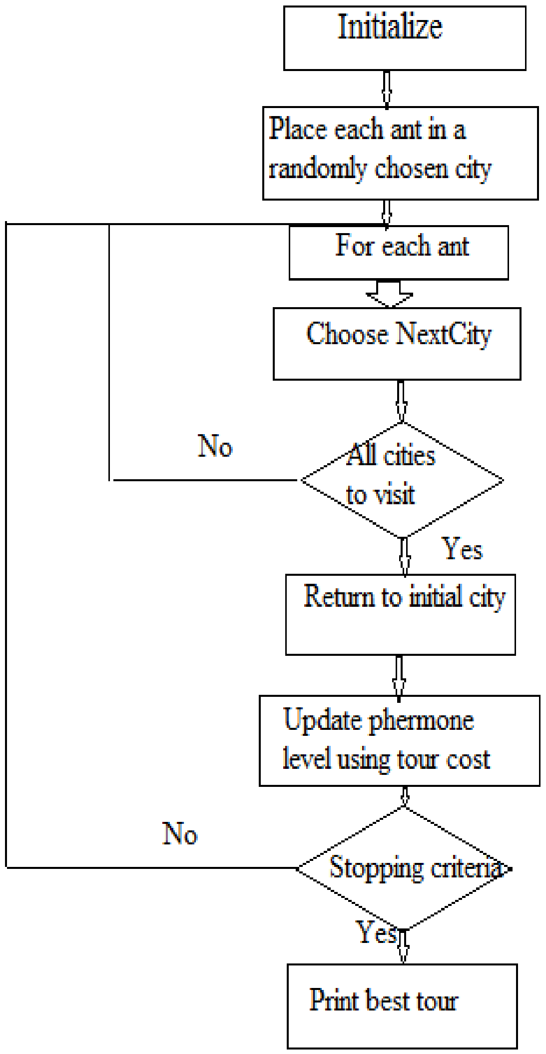


Fig 1.2- Flow chart of working of ACO algorithm

1.3 Travelling Salesman Problem (TSP)

Travelling Salesman Problem [5, 6, 8, 9] was given by the Irish mathematician W.R. Hamilton and British mathematician Thomas Kirkman in 1800s. But this problem was firstly formulated by Karl Menger as a mathematical problem in 1930.

Let there are number of cities and distance between each pair of cities is given, the objective of the Travelling Salesman Problem is to find the shortest distance that visits all the cities exactly once and return back to departure city. In other words we can say that in TSP goal is to minimize the total roundtrip distance. Graphical representation of TSP is given below:

Let G = (V, E) be a graph.

- V is collection of vertices or cities.
- E is collection of edges.

If distance D between city 'x' and city 'y' is same as distance between city 'y' and city 'x' i.e. $d_{x,y} = d_{y,x} = D$ then it is called symmetric travelling salesman problem (STSP) otherwise, it is called an asymmetric TSP (ATSP).

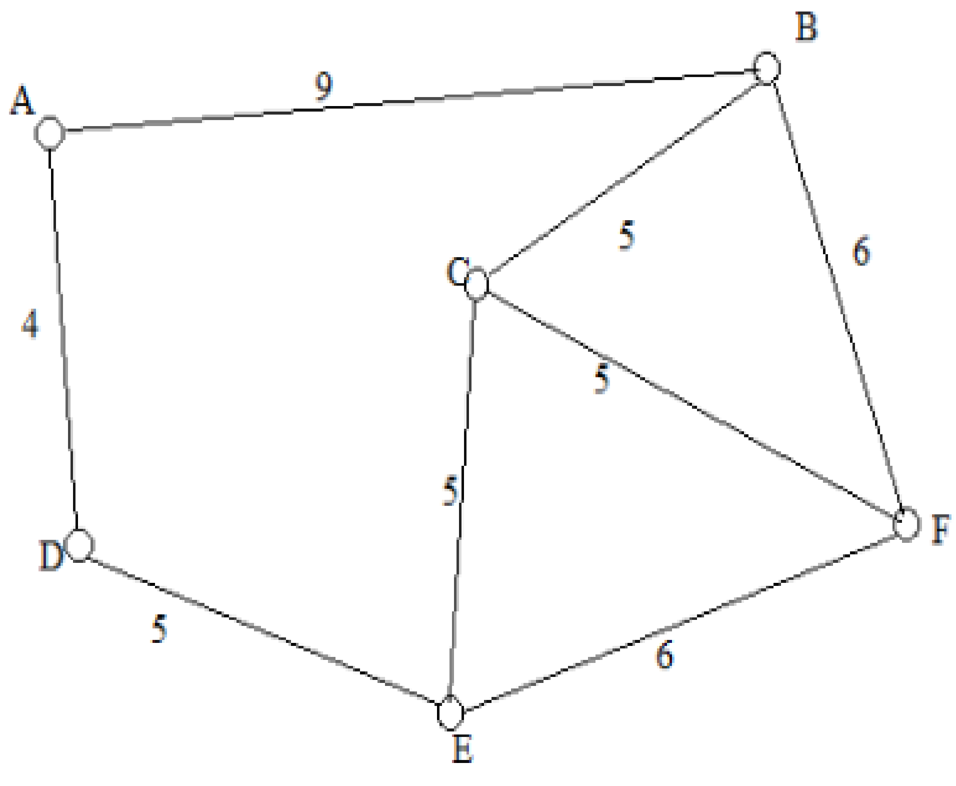


Fig1.3- Graphical view of TSP

In above graph each node represents a city and weight assigned to each edge represents distance between two cities. In TSP task is to find the minimum distance so that one can travel each city exactly once and return back to starting city.

Assumptions for the Travelling Salesman Problem are:

1. There should be finite no. of cities.
2. Each city should connect to next city.

Constraints for the Travelling Salesman Problem are:

1. Each city should traverse exactly once.
2. Salesman should reach at starting city after traversing all the cities.

If number of nodes in the TSP is less, it can be solve easily in a reasonable time and effort. But as we increase the number of nodes in the TSP then complexity increases rapidly. So to save the time and effort we use metaheuristic algorithm. Let suppose 100 cities are given in the TSP. Every individual city is connected to every other city in the problem and distance between them is given. Any city can be chosen as departure city and we can go in any direction, either forward or backward. To find the solution of TSP with this amount of cities with traditional method would be a difficult task because it will give [(100 - 1)!/2] number of possible solutions. Out of this to find the best path will take huge amount of time and effort. So to reduce this we use ACO algorithm that reduces the time and effort and gives optimal solution when there is large no. of cities in the problem.

1.4 Cuckoo search (CS)

Cuckoo search [10, 11, 12, 14] is a latest metaheuristic optimization algorithm and was developed in 2009 by Xin She yang and Suash Deb. Cuckoo bird does not make nests for laying its egg, it uses other bird's nest for laying eggs. This optimization algorithm is inspired by this behaviour of cuckoo bird. If host bird recognizes the eggs of cuckoo bird, host bird will either destroy the eggs or leave the nest. Each cuckoo can lays its egg in the nest of host bird within a specific region; this region is called Egg Laying Region (ELR) [11]. The working method of Cuckoo search is this behaviour and it is used to solve different kinds of optimization problems. The main advantage of cuckoo search is that it uses very less number of parameters which makes it better than other metaheuristic algorithms. Each egg in a nest shows a solution, and a cuckoo egg shows a new solution. Our goal is to find new solution (cuckoos) among all the existing solutions present in the nest.

Working on Cuckoo Search is depending on following three rules:

1. Each cuckoo can lay one egg at a time, and can dump its egg in any random selected nest.
2. The nest which would have high quality of eggs will called best nest ant it will go to the next generation;
3. The number of available host nests is fixed and the egg laid by a cuckoo is discovered by the host bird with a probability $p_a \in (0,1)$.

In Cuckoo Search, there are only two parameters, population size n probability function. The convergence rate [11] of cuckoo search is not related to the parameter p_a [11,15] . It means that we do not have to change these parameters for a specific problem. The major advantage of CS is that it is more robust and generic and for number of optimization problems, when it is compared with other metaheuristic algorithm and performance of local search by this are much faster.

Flow chart of Cuckoo Search Algorithm is given as follows:

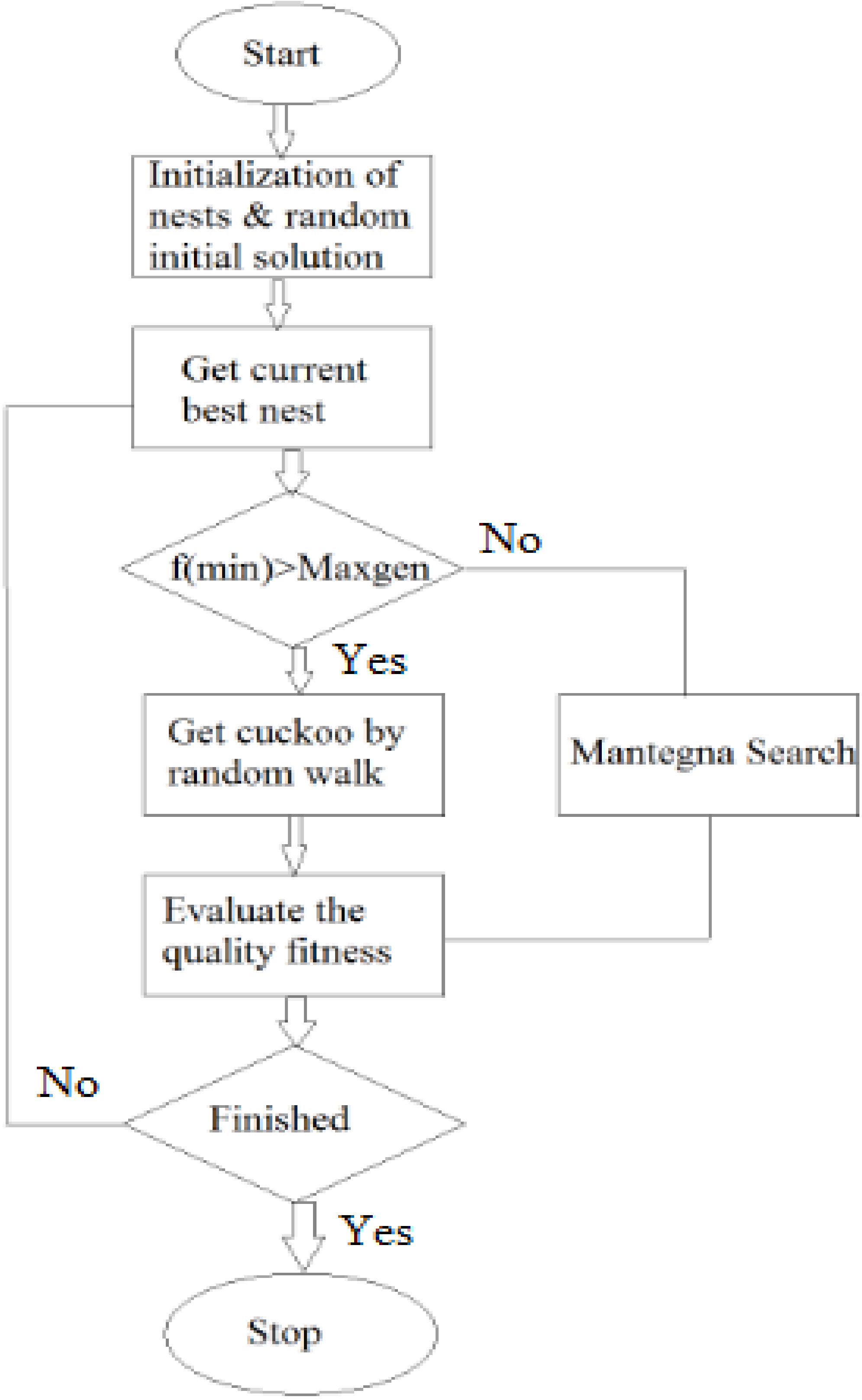


Fig 1.4- Flow chart of Cuckoo Search Algorithm

Fig1.4 shows the flow chart standard cuckoo search algorithm which is used in used in optimization algorithm. It is developed by Suash Deb in 2009. Further in 2011 Ramin Rajabioun has proposed optimized cuckoo search algorithm. Complete flow chart of this algorithm is given as:

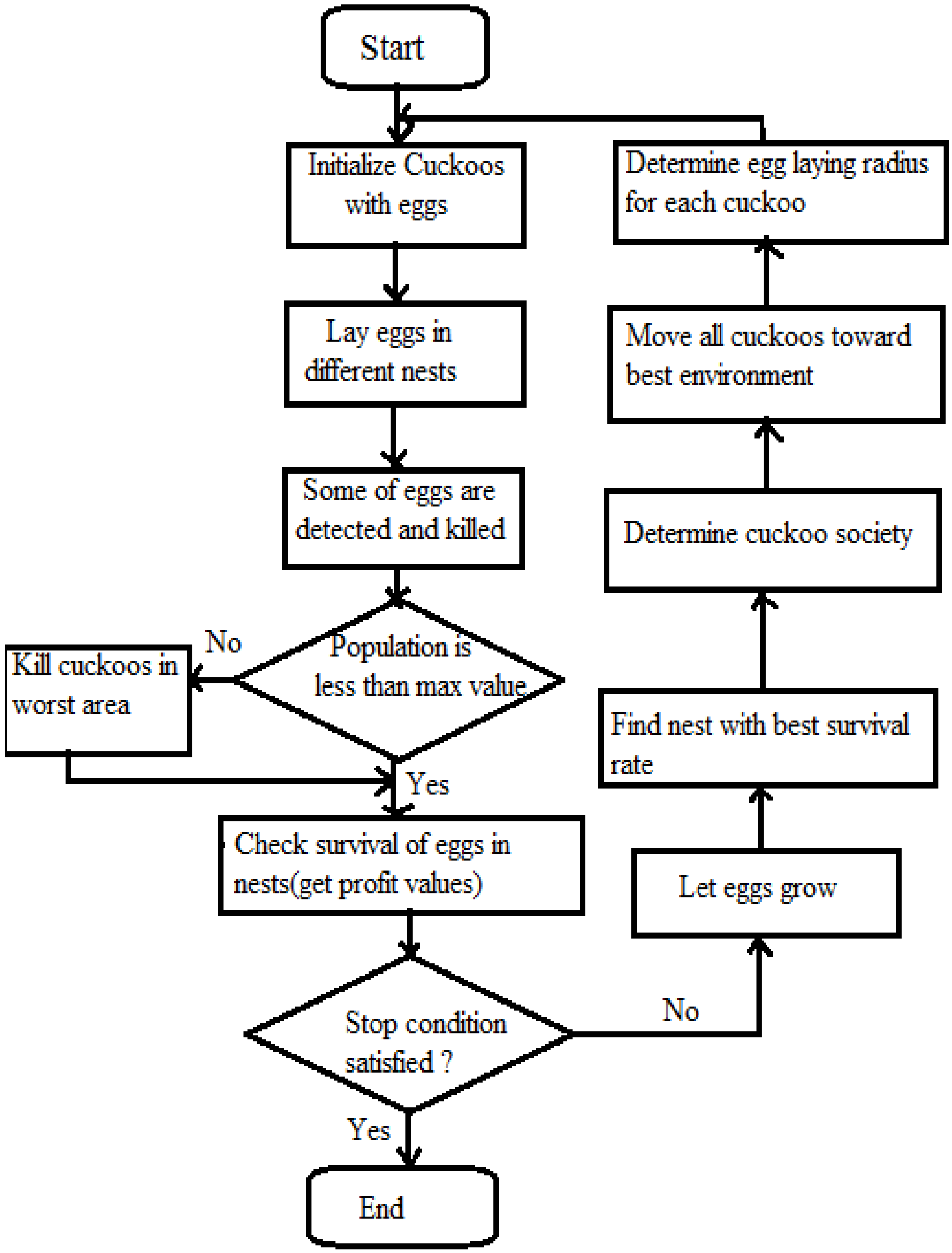


Fig 1.5: Flow chart of cuckoo search optimization algorithm

The working process of Cuckoo Search Optimization Algorithm is as follows:

1. Initialization of cuckoo habitats with some random points on the profit function.
2. Initialize some eggs to each cuckoo.
3. Define ELR (Egg Laying Region) for each cuckoo.
4. Let cuckoos to lay eggs inside the corresponding ELR.
5. Destroy those eggs that are detected by host birds.
6. Let eggs hatch and chicks grow.
7. Evaluate the habitat of each newly grown cuckoo.
8. Limit maximum number of cuckoos in environment and kill those who live in worst habitats.
9. Collect cuckoos and find best group and select goal habitat.
10. Let new cuckoo population immigrate toward goal habitat.
11. If it satisfied the condition then stop else go to step 2.

After the study of ACO and CS (metaheuristic algorithms) we can conclude that metaheuristic algorithms have advantage over traditional algorithms. It is very difficult to solve the NP complete problem (for which we cannot find exact solution) using traditional algorithms especially when problem size is large. In that case metaheuristic algorithm works better. The main advantage of metaheuristic algorithms is as follows:

1. Traditional algorithm is not robust when we need to make changes dynamically in the problem, to make the changes traditional algorithms require completely restart of the solution steps. On the other hand metaheuristic algorithms are robust to make the changes dynamically.
2. Metaheuristic problems can be applied to all the problems which can be shown as optimization problem.
3. Metaheuristic algorithms are robust to hybrid with other algorithms.
4. Metaheuristic algorithms have ability to solve the problems that cannot be solving by human expertise.

Chapter 2

TERMINOLOGY

Brood parasite birds- This kind of birds does not make nests for laying eggs. They lay eggs in other birds' nest. Ex- Cuckoo.

Combinatorial Problem- A combinatorial problem is that problem where an optimal ordering and selection of options is desired.

Cost - Time and space taken by algorithm for execution.

Hybrid- Merging of two algorithms.

Local Search- Local search is a metaheuristic method to solve NP complete problems. It moves from one solution to another in the search space till the time an optimum solution is not found or time is not over.

Metaheuristic algorithms- These algorithms are nature inspired algorithm. These kinds of algorithms are problem independent algorithms that can be applied a broad range of problems.

NP- Complete Problem- The problem for which we cannot find exact solution for a given input. In this kind of problem complexity increases rapidly as the input size are increases.

Chapter 3

REVIEW OF LITERATURE

There are many algorithms have been proposed by researches to solve the Travelling Salesman Problem out of which metaheuristic algorithms are most efficient algorithms to solve this problem. There are many advancements have been also done in Ant Colony optimization algorithm to solve the Travelling Salesman Problem. Some of the research papers have been discussed below:

Nada M. A. Al Salami [1] proposed an ACO algorithm for optimization problem. In this paper author has explained that ACO algorithm can generate redundant states in the problem, so to overcome this problem genetic programming has merged with it. Since ants walks chooses its path based on high pheromone density. Ants cannot adapt themselves according to the problem, so genetic programming has been used to adapt the ants according to the problem. It can perform computations like recursion on different kinds of variables to get the desired result.

Utkarsh Jaiswal et al. [2] have also proposed ACO algorithm. In this paper authors have presented ACO algorithm as a strong algorithm for solving hard combinatorial problems such as TSP. The main advantage of ACO is that it saves lot of computation time and provides high level of optimum result. Author has given idea to improve ACO to find out better results. To enhance the performance of ACO author has given suggestion to hybrid it with PSO (Particle Swarm Optimization). It will produce better results for combinatorial problems.

John E. Bell et al. [3] proposed ACO algorithm to solve vehicle routing problem. In this paper author has solved routing problem with ACO. To improve the results author has taken two ACO methods with multiple ant colonies. Multiple ant colonies provide better performance than traditional method and gives better results for large problems. Here candidate list size plays important role to achieve good results for the problem. Algorithm gives good results when candidate list size is 10 to 20 but when its range goes higher, performance degrades for the problem.

M. Duran Toksari [4] has proposed ACO algorithm to search global minimum. In this paper author has presented the ACO to find out global minimum. This algorithm is based upon idea that every ant will walk only those paths which has given the best solutions in the previous iteration. In this author has associated m ants with m initial vectors. Then modifies each vector based on the pheromone trail. In this proposed ACO algorithm the amount of pheromone increases only through the best objective function. The proposed algorithm gives good results to find out global minimum but it is slows.

Majid Yousefikhoshbakht et al. [5] proposed a modified ACO algorithm to solve multiple travelling salesman problems. In the proposed algorithm authors have presented a modified ACO algorithm which is

New Modified Ant Colony Optimization (NMACO) for solving multiple travelling salesman problems. In this new modified algorithm authors have hybrid the ACO algorithm with swap, insert and 2-opt algorithm. In the first step of the algorithm using transition rule and candidate list n solutions has been build. In the second step algorithm uses swap, insert and 2-opt to find out current best solution and then updates the global pheromone value. In this way new modified ACO algorithm does not trap in local optimum points and provides better results than classic ACO. The flow chart to solve the TSP by author is proposed as given:

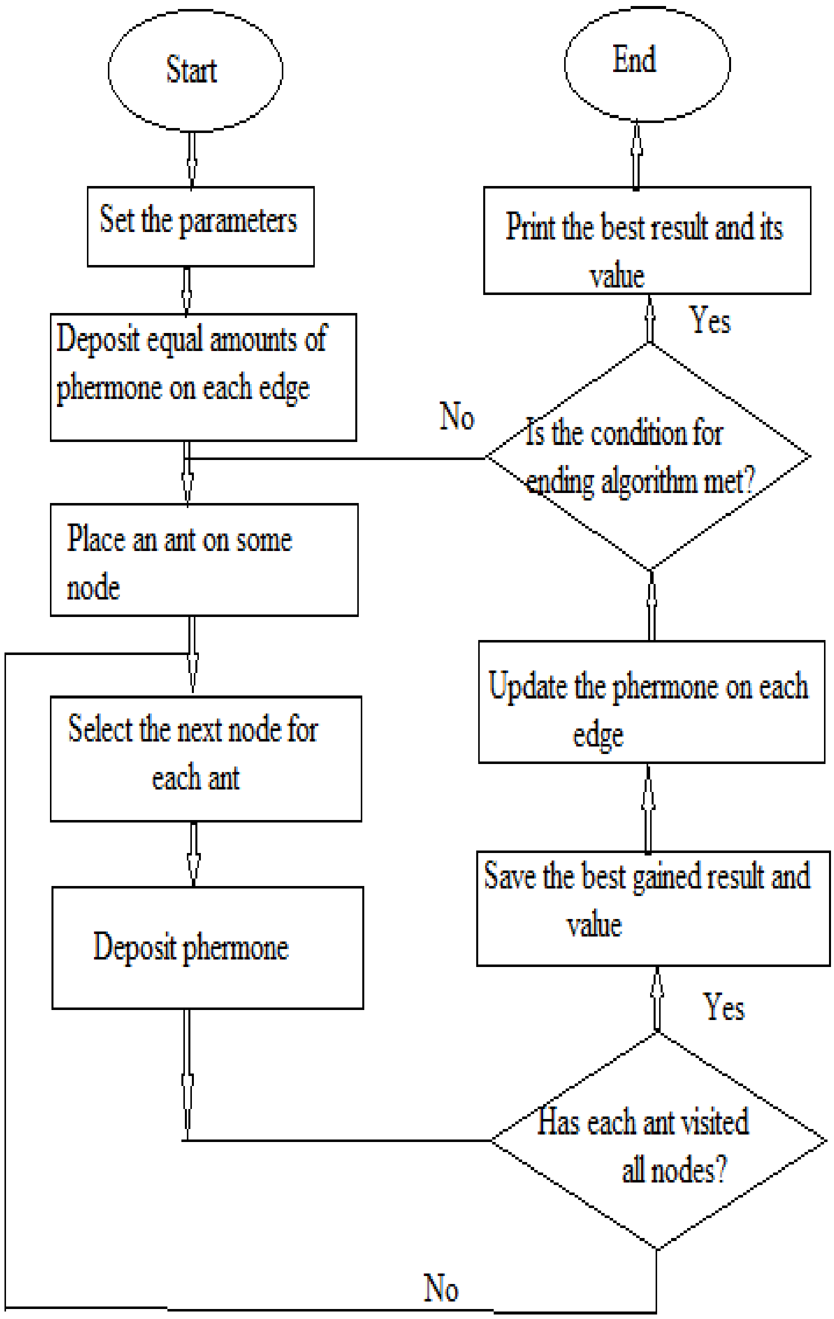


Fig 3.1- Flow chart of solving TSP using ACO

Hui Yu [6] has proposed more optimized ACO algorithm with updating of local pheromone. In this paper author has proposed, local pheromone update strategy, city-select strategy, local optimization strategy and optimum solution prediction strategy to optimize ant colony algorithm for TSP. In this paper author has overcome the problem of ACO of trapping in the local optimum by presenting an improved ant colony algorithm. Since the standard ACO uses global pheromone update method which is slow so to overcome this problem local pheromone update method is used. In local pheromone update method nearest neighbour algorithm and greedy algorithm is used to find the next city by ants to overcome the delay effect of global pheromone update.

Isa Maleki et al. [7] proposed hybrid ACO algorithm and Chaos theory for solving dynamic travelling salesman problem. In this paper author has used concept of chaos theory with ACO algorithm which provides more optimal solution than that of ACO for dynamic travelling salesman problem. Proposed algorithm gives better results in terms of time and effort. Complete proposed algorithm is represented in above flow chart. After updating the pheromone information next shortest distance city is selected using chaos operator. It will make faster to choose the next city.

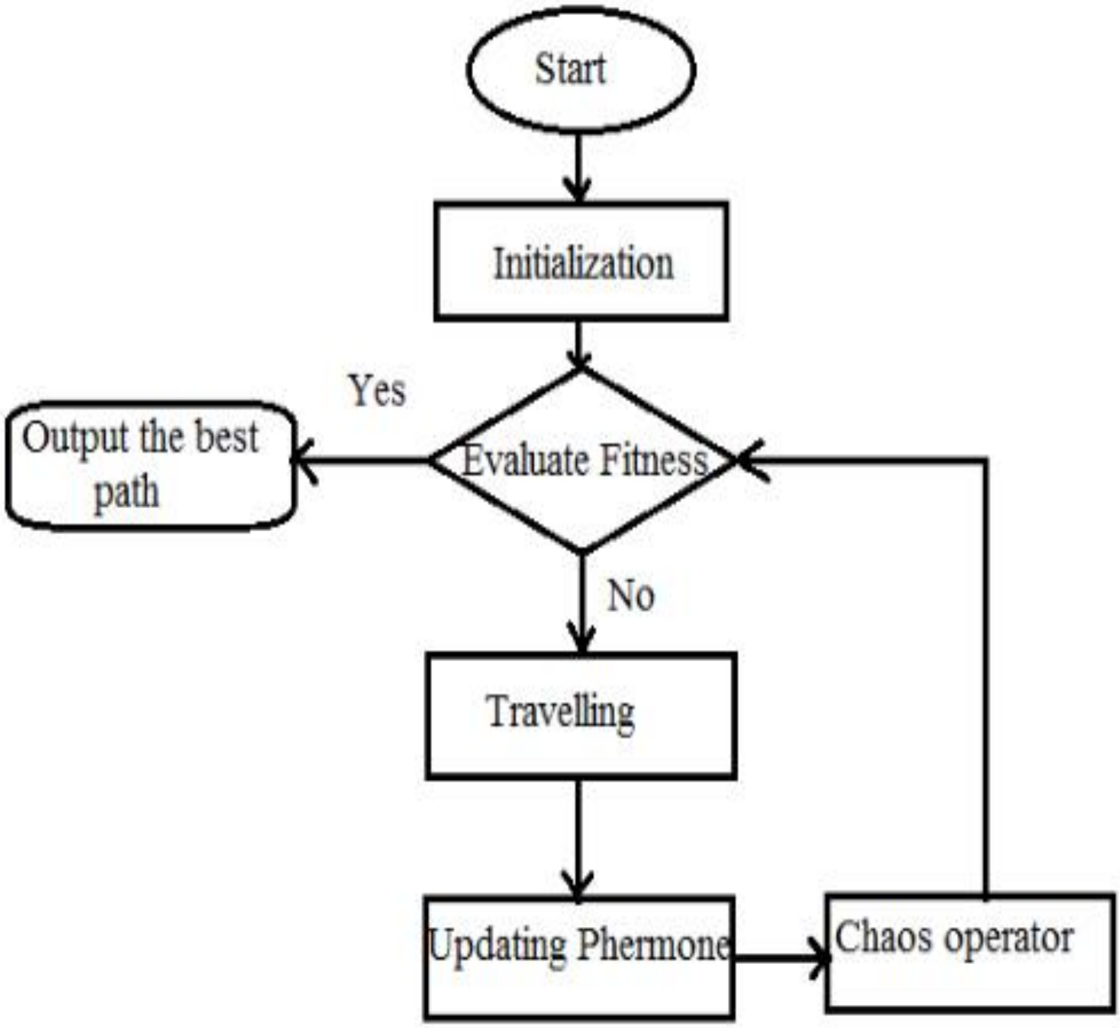


Fig 3.2- Hybrid algorithm for TSP

Ivan Brezina Jr. Zuzana Cickova [8] has proposed solution of famous travelling salesman problem by using ACO algorithm. In this paper author has formulated ACO algorithm for travelling salesman problem. Author has presented that if no. of ants is less then pheromone density on edge will be less. So probability to change the path is higher longer path might be chosen. But if there is high no. of ants then pheromone density would be high on the edges and ants will choose the shorter path. So finally this is concluded that higher no. of ants will produce better results by using less no. of iterations and short time.

Zar Chi Su SuHlaing et al. [9] have also proposed solution of famous travelling salesman problem by using ACO algorithm. This paper presents an improved ant colony optimization for solving travelling salesman problem. In this paper author has discussed how to avoid mature convergence and stagnation behaviour. Since heuristic information plays important role producing paths of high quality paths. So author has taken a dynamic heuristic parameter which updates based on entropy. In starting stage of learning the pheromone density is less on edge so heuristic parameter will be high. In that case entropy information will be highest. As the pheromone density will be high on the edges, heuristic parameter will degrade. So entropy information will start degrading. If entropy information is zero in that case pheromone density will be highest.

Xin-She Yang et al. [10] proposed Cuckoo Search algorithm which uses levy flight for step size. In this paper author has formulated a new metaheuristic algorithm named as Cuckoo Search (CS) with levy flights to solve optimization problems. Cuckoo search works based upon the breeding behaviour of cuckoo bird. This algorithm works better than PSO (Particle Swarm Optimization) and GA (genetic algorithm). The main advantage of CS is that it has only two essential parameters, population size n and probability function $p_{a.}$ More importantly we don’t need to fine tune these parameters according to the problem. Apart from it this algorithm makes local search faster as well.

Aziz Ouaarab et al. [11] proposed a discrete version of cuckoo search algorithm to solve travelling salesman problem. In this algorithm author has proposed new kinds of cuckoo’s population which is more effective to solve continuous problems as well as combinatorial problems. In the proposed algorithm authors have introduced new fraction p_c (set of smart cuckoos) to improve the solutions of current search. Cuckoos can move from one area to another using levy flights or Mantegna search. The main benefit of this new algorithm is that it locates best solution in every region without trapping in local optimum.

Ramin Rajabioun [12] proposed optimized cuckoo search algorithm. Author has developed this optimization algorithm by inspiring egg laying behaviour of cuckoo bird. The result of this algorithm is compared with standard GA and PSO. It provides much better global optima achievement and much faster convergence rate. The convergence rate given by Cuckoo Optimization Algorithm (COA) is in less iteration than GA and PSO. Complete flow chart of proposed Cuckoo Optimization Algorithm is represented in figure-

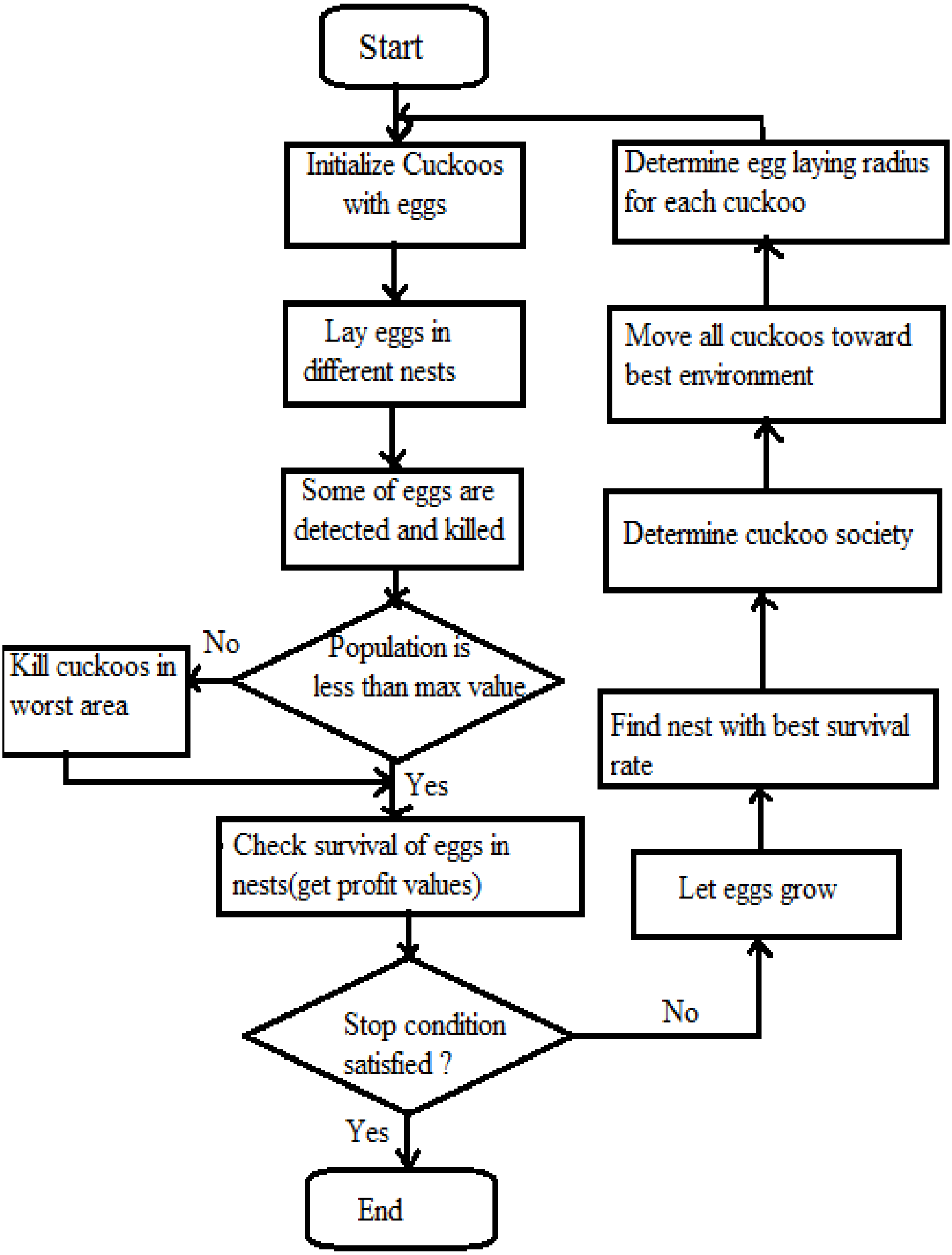


Fig 3.3- Flow chart of COA

.

Li-Pei Wong el al. [13] proposed solution of travelling salesman problem with Bee Colony Optimization algorithm (BCO) which uses local search method. In this paper author has presented BCO algorithm for TSP that works based upon foraging behaviour of bees. The problem in BCO is that it easily trapped in local optimum. So to overcome this problem of BCO authors have introduced 2-opt heuristic algorithm. By using this algorithm it does not trap in local optimum. This can be further explore by using 3-opt heuristic algorithm with BCO.

Sangita Roy et al. [14] proposed a review on cuckoo search algorithm using levy flights. Cuckoo search process is a optimization process which was developed in 2009. It is created upon the procedure of cuckoo birdies which lay their broods in the host birdie's nest. Some Birdies can distinguish the offspring of Cuckoo. If a host bird recognizes the eggs, it will either throw of cuckoo's egg or it will abandon the nest. Host bird identifies the cuckoo's egg on some set of worst nests which are not having good fitness value. Cuckoo search procedure is the best procedure among all the metaheuristic processes. Cuckoo search process is now being used to resolve numerous optimization problems. It is very fast algorithm as equate to other metaheuristic algorithm because it uses discrete a single factor for searching. It can be useful for various optimization difficulties due to its fast and efficient result as compare to other evolutionary algorithm. Result shows that it can perform better than other metaheuristic algorithms in many applications because it uses only a single factor for searching.

J.B. Jona et al. [15] proposed Ant-cuckoo Colony Optimization in feature selection. In this paper authors have justified why they have employed cuckoo search in ACO. The main problem of ACO is to find local search. Since ants walks only through the path where pheromone density is high. So till the time pheromone density will not be high on shorter path, the path will not be chosen. So to overcome this problem of ACO authors have applied cuckoo search to make local search faster.

Belaid Ahiod et al. [16] suggested a distinct kind of peculiar examine method to explain travelling salesman problem. In this method writer has announced new classes of cuckoo's populace that is precise in influence to solve nonstop difficulties along with combinatorial complications. In this process writers have announced new fraction p_c to increase the results of present pursuit. Cuckoos travel from lone zone to additional by impose hops. The main benefit of novel procedure is that it finds finest result in each area devoid of duping in confined optimum.

Aziz Ouaarab et al. [17] proposed solution of travelling salesman problem using cuckoo search algorithm. In this algorithm authors have used random key encoding scheme to pass a combinatorial space from continuous space. In the proposed algorithm displacement of spaces is done using levy flights. Random key encoding scheme uses vector of real numbers in which each number is associated with weights. These weights generate a combination as a solution. Random numbers are generated using $[0.1)^m$, where m represents the size of TSP. Complete flow chart of proposed algorithm is shown as below:

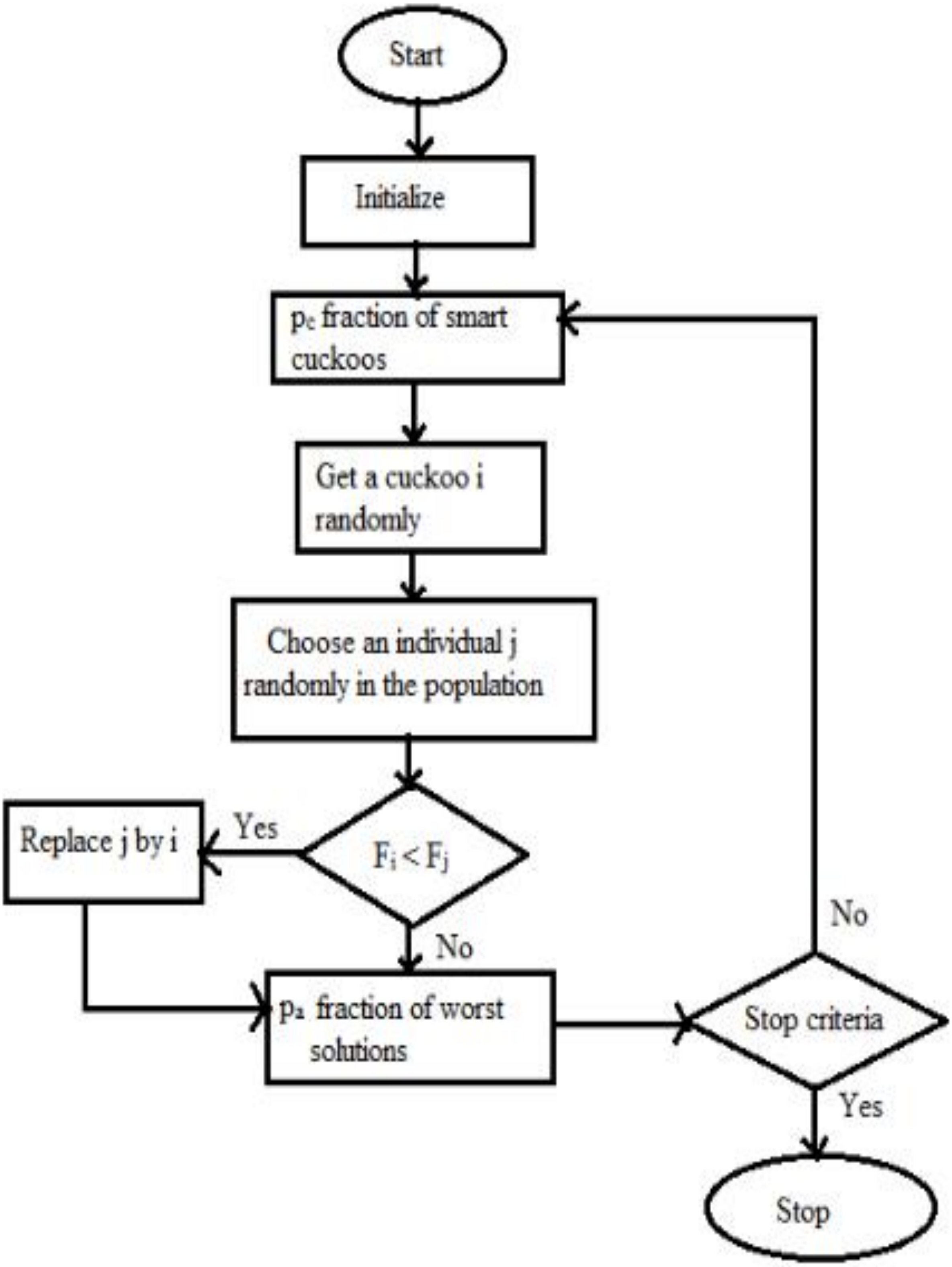


Fig 3.4- Flow chart of RKCS algorithm

Byung-In Kim et al. [18] proposed different heuristic and metaheuristic algorithms to solve famous Travelling Salesman problem. In this paper authors have solved TSP using 2-opt, 3-opt, genetic algorithm, greedy approach and simulated annealing. Author has shown comparison of all five algorithms. Authors have also designed improved algorithm for TSP. The result is improved by removing crossed edges. The diagram below shows how authors have improved the results.

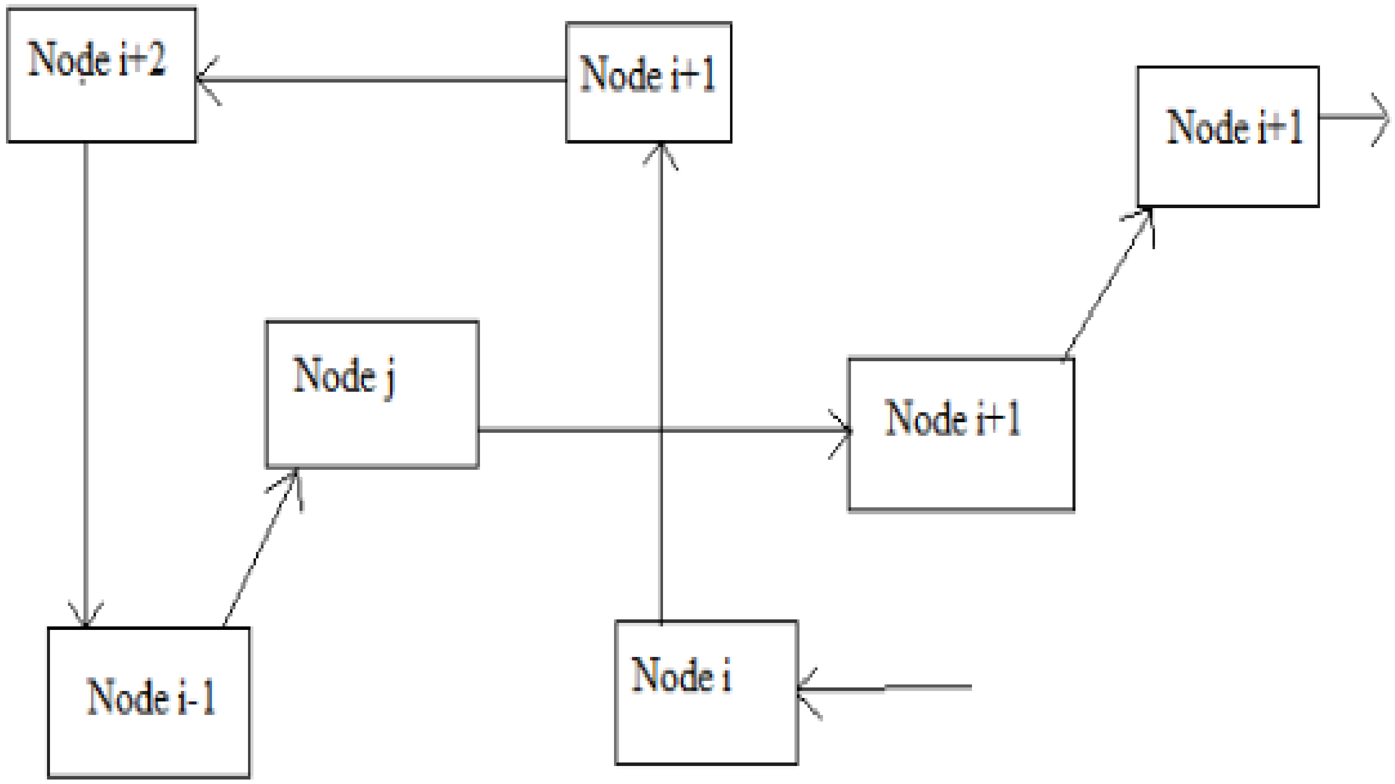


Fig 3.5 Original Result of traversing the graph

In above figure each node is representing a city and each edge is representing path between cities. In this case covered distance will be high. The reason behind it is that there is a cross point which is making distance larger. To improve this author has removed cross path. The result of improved diagram is shown in next diagram. To improve the results of traversing the above graph firstly crossed edge is removed and added an edge from node i to i-1. In both the cases we have same no. of nodes and edges. The next diagram shows it clearly.

•

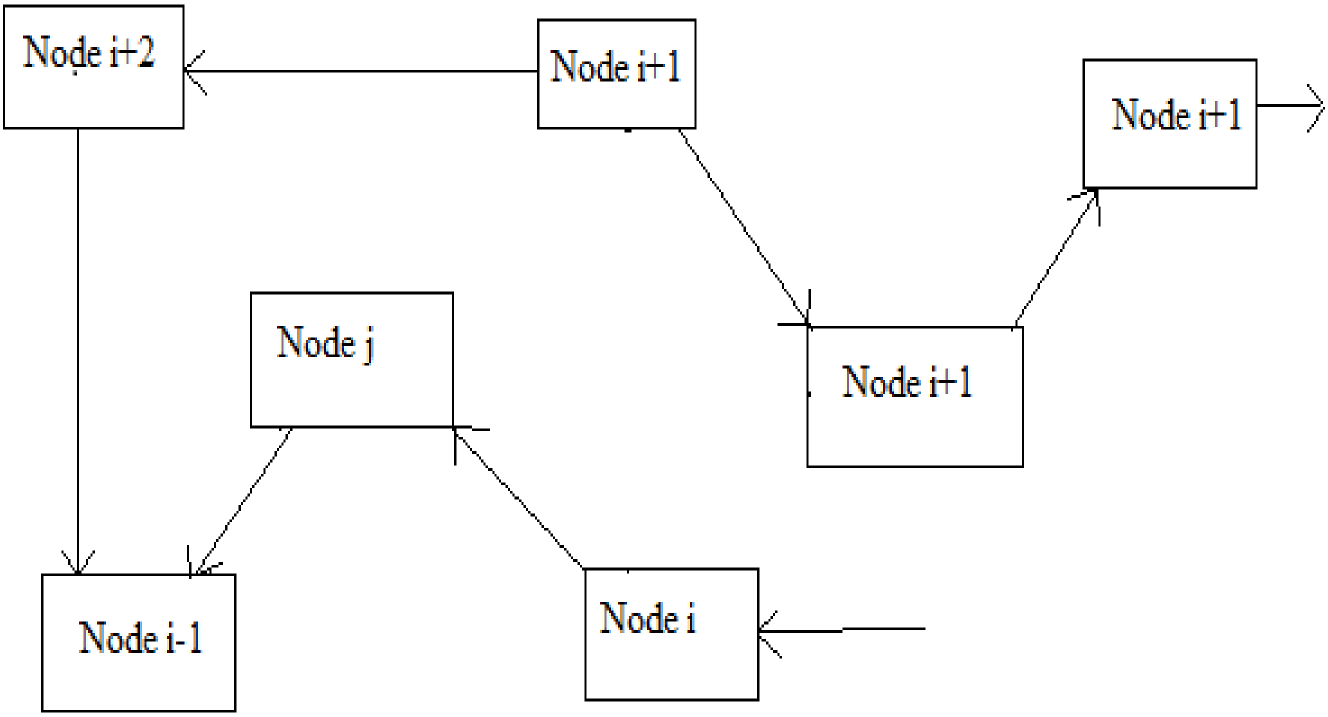


Fig 3.6- Improved Result of traversing the graph

Ala'a Abu-Srhan et al. [19] suggested a hybrid algorithm to solve TSP. In this paper author combines the advantages of both cuckoo search and genetic algorithm. Traveling salesman problem is a NP–hard problem and many algorithm has been applied to solve it .Genetic algorithm provides better result as compare to other algorithms. But difficulty of trapping in local optima makes it unsuitable for solving NP –hard problem. The main problem in genetic algorithm is that it easily traps in local optima so to remove this problem authors have used cuckoo search algorithm which is very fast and efficient algorithm. Cuckoo search algorithm is an efficient algorithm because of less parameter.

M. Z. Rashad et al. [20] suggested a hybrid genetic and cuckoo search algorithm for improving speed in solving optimization algorithms. Author suggested a hybrid algorithm by considering optimization as main factor. The proposed algorithm can be applied in many fields.

David S. Johnson et al. [21] have proposed various algorithms to solve Travelling Salesman Problem with local optimization. Authors have solved TSP using Tabu Search, Simulated Annealing, Genetic Algorithm and Neural Networks. Authors have shown that how these algorithms can be apply on TSP and evaluated the success of each algorithm. In case of 2-Opt how TSP is solved is mentioned by authors. Here a, b, c and d are nodes of the graph. The clear diagram is on next page.

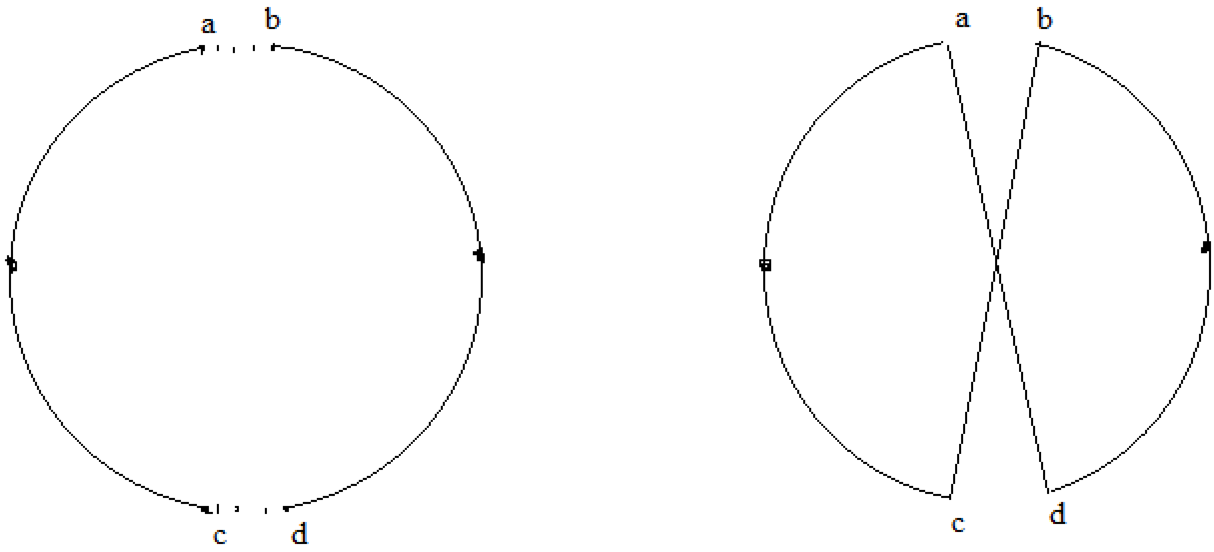


Fig 3.7-Opt move (before and after)

Chapter 4

SCOPE OF THE STUDY

TSP is a NP complete problem it means that we cannot find exact solution of this type of problems. As we increase the size of the TSP, running time and cost increases continuously. TSP is simple problem when no. of nodes is less but it becomes a challenging problem when no. of nodes increased in the problem. In that case it is difficult to solve it using traditional method.

In our research we are using ACO algorithm with Cuckoo Search algorithm for traversing the nodes. In our proposed algorithm we have made local search faster using Cuckoo Search. In future we can modify it by using Random key cuckoo search algorithm to enhance the performances of our proposed algorithm. This proposed algorithm can be used in vehicle routing problem. Our proposed algorithm is a light weighted algorithm so we can use it in various optimizations routing problem. Due to light weighted nature of proposed algorithm, the execution time is much faster than other proposed algorithms. This can be especially useful when large scale problem instances are considered.

Chapter 5

OBJECTIVE OF THE STUDY

If number of nodes in the TSP is less, it can be solve easily in a reasonable time and effort. But as we increase the number of nodes in the TSP then complexity increases rapidly. Our main objective is to solve Travelling Salesman Problem with large no. of nodes. Our second main objective is to remove the problem of Ant Colony Optimization algorithm which we have done by merging Ant Colony Optimization algorithm with Cuckoo Search algorithm. A comparative study will also be done to analyse the enhancement of proposed algorithm. The designed algorithm has following properties -

- Algorithm should not trap in the local minima.
- Algorithm should perform local search fast.
- Algorithm should provide better results to NP-hard problems.
- Algorithm should optimize the problem heuristically.
- If the size of the problem is increased the complexity should not increase.
- To find always a faster global best solution.
- Reduction of the search space.

Chapter 6

RESEARCH METHODOLOGY

The global update pheromone method is used by the ACO to find the best path. The pheromone updating is partly postponed by the global pheromone update method of ACO. So in that case, the ant colony optimization algorithm does not able to provide an optimal solution. In the process of searching, the ant colony always moves towards the path of the highest pheromone density exists. The main drawback of ACO is that it cannot make local search faster. So ACO is entangled in local optimum and better solution cannot be found. We are going to propose a fast and efficient algorithm which will remove the drawback of Ant Colony Optimization algorithm. So to overcome the problem of ACO we have designed a hybrid algorithm in which we have used Cuckoo search algorithm with ACO algorithm to make local search faster for traversing the node.

6.1 Existing System

ACO is a good metaheuristic algorithm to solve the travelling salesman problem. This existing algorithm uses the behaviour of real ants to search the shortest path between nests and food source. While walking, ant leaves a chemical material on its path. Based upon the density of pheromone, ant moves. Higher the pheromone density, higher will be probability to choose the path by rest of the ants. So probability of high amount of pheromone on shorter path has higher probability. By this way ants always choose shortest path.

Operation

The ACO algorithm to solve TSP involves three steps: Initialization, construction of solutions and updating of trails.

Initially, we put each on a randomly selected city. During the construction phase ants select the next city based upon the following probabilistic decision rule. Let ant is on city i, the probability of moving of ant from city i to city j is given as:

$$p_{ij}^{k}(t) = \begin{cases} \dfrac{[\tau_{ij}(t)]^{\alpha} \cdot [\eta_{ij}]^{\beta}}{\sum_{k \in allowed_k} [\tau_{ij}(t)]^{\alpha} \cdot [\eta_{ij}]^{\beta}} & \text{if } k \in allowed_k \\ 0 & otherwise \end{cases}$$

where τ_{ij} is the trail intensity between edge i & j and η_{ij} is the heuristic visibility of edge (i, j) is given as η_{ij} and $\eta_{ij}=1/d_{ij}$. α is pheromone influence factor, β is local node influence. After the completion of its tour by each ant, the amount of pheromone on each path is updated by following equation:

$$\tau_{ij}(t+1) = (1-\rho)\,\tau_{ij}(t) + \Delta\tau_{ij}(t)$$

where ρ is pheromone evaporation coefficient and

$$\Delta \tau_{ij}(t) = \sum_{k=1}^{m} \Delta \tau_{ij}^{k}(t)$$

$$\Delta \tau_{ij}^{k}(t) = \begin{cases} \dfrac{Q}{L_k}, & \text{if } (i, j) \in \text{tour done by ant } k \\ 0 & \text{otherwise} \end{cases}$$

(1- ρ) is the pheromone decay parameter (0<ρ<1) .L_k is the length of the path traversed by ant *k* and *m* is the number of ants. Complete flow chart is given below:

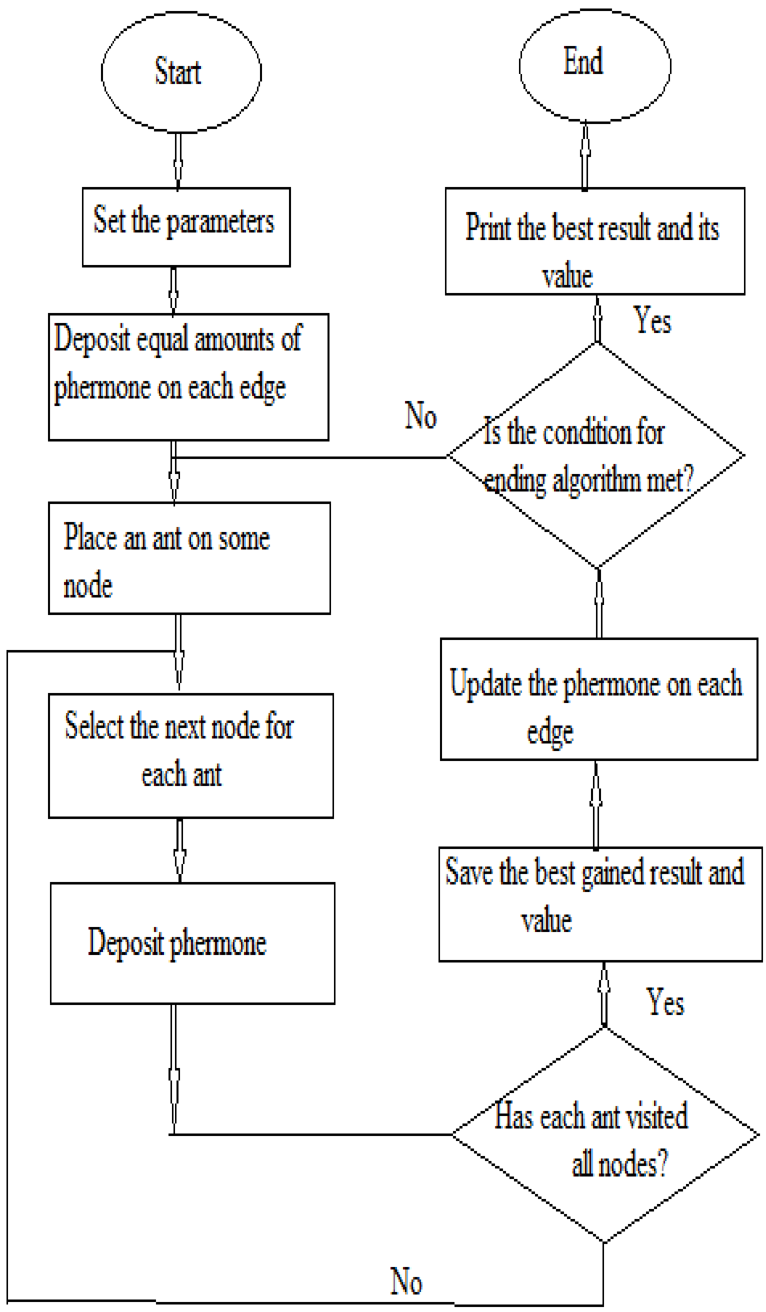


Fig 6.1- Flow chart to solve TSP using ACO algorithm

6.2 Problem Faced by ACO algorithm

There are some problems faced by ACO algorithm which are given as follows:

- In ACO ants always walks through that path where pheromone density is high, so till the time pheromone density will not be high ants will not walk through that path, this makes whole process slow.
- ACO traps in local minima.
- It has search space large.
- It provides slow global best solution.

6.3 Proposed System

In our research we proposed a new fast solution for Travelling Salesman Problem by using Ant Colony Optimization algorithm with cuckoo search algorithm. To solve this problem first we have formulated it as an optimization problem. In this case first we have formulated the Travelling salesman problem according to proposed hybrid ACO and CS algorithm. Our proposed algorithm combines the advantages of Ant Colony Optimization and Cuckoo search. The major drawback in the ACO is that while trying to solve the combinatorial optimization problems the search has to perform much faster, but in ACO ant walk through the path where the chemical substance called pheromone density is high. In order to overcome this drawback, Cuckoo search is used.

Complete algorithm is shown as below:

Algorithm to solve TSP

1. Initialization.
2. Insert no. of nodes in the graph to traverse.
3. Traverse each node using hybrid algorithm (ACO and CS).
4. Traverse each node exactly once and return back to starting node.
5. If all nodes traversed then terminate else repeat step 2 to 5.

Ant Colony Optimization algorithm

1. Initialize (pheromone trails, ant solution construction).
2. Assign each ant to different nodes to traverse the path.
3. Perform local search using cuckoo search algorithm and then update the path.
4. For each ant calculate fitness function and update the value of pheromone.

If total iteration < maximum iteration then go to step 2 else terminate.

Cuckoo Search algorithm

1. Initialize nests and random initial solution
2. Get the current best nests.
3. While (F_{min} > max generation)

   Get the cuckoo by random walk, if not replace it by Mantegna search

4. Evaluate the quality fitness, randomly choose nests among n (call as j)
5. If ($F_i < F_j$) then replace j value by new solution.
6. Retain the best solution and nest.

End while else go to step 2

Hybrid Algorithm to solve TSP

1. Initialization.
2. Insert no. of nodes in the graph to traverse.
3. Traverse each node using hybrid algorithm (ACO and CS).
4. Assign each ant to different node
5. Perform local search using cuckoo search
6. Initialize nest
7. Get the current best nest
8. **While** (F_{min} > max generation)
9. Get the cuckoo by random walk, if not replace it by Mantegna search
10. **End while**
11. Evaluate the quality fitness, randomly choose nests among n (call as j)
12. If ($F_i < F_j$) then replace j value by new solution.
    Else retain the best solution and nest.
13. Choose next city according to fitness value of cuckoo search
14. For each ant update the value of pheromone and calculate fitness function.
15. If total iteration < maximum iteration then go to step 2
    Else terminate
16. Traverse each node exactly once and return back to starting node.
17. If all nodes traversed then terminate
    Else repeat step 3 to 17.
18. Stop

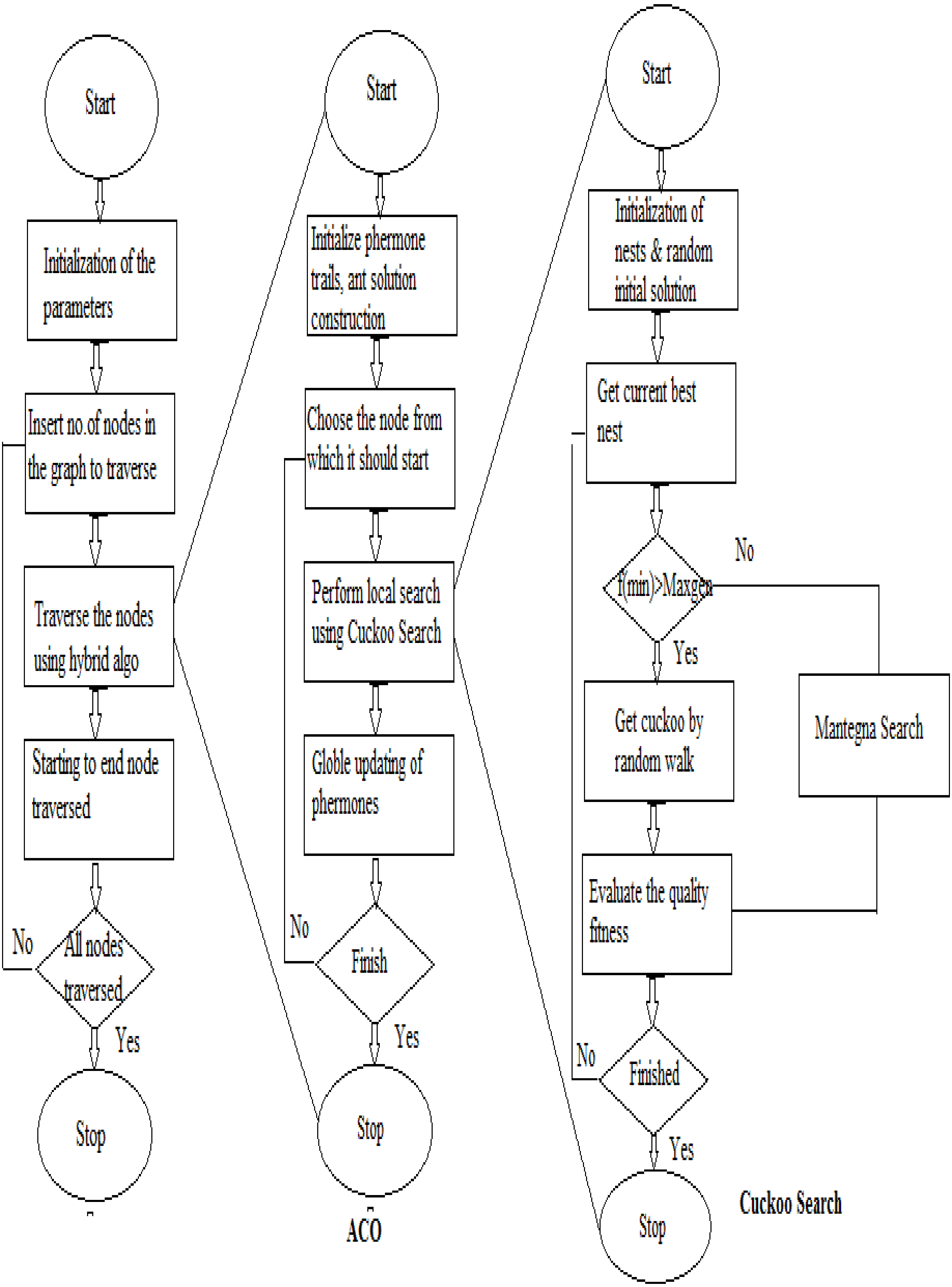


Fig 6.2- Flow chart for solving TSP using hybrid ACO and Cuckoo Search

Given diagram shows the complete flowchart to solve the Travelling Salesman Problem using hybrid algorithm. The proposed hybrid algorithm works on traversing of nodes. In traversing the local search is performed using Cuckoo Search. The local search in the existing algorithm (ACO) was slow, this drawback of existing algorithm (ACO) is removed using Cuckoo Search.

In cuckoo search main component is Mantegna algorithm which is used to find the random points in the space. It is used to generate random numbers on the basis of symmetric levy stable distribution. The value of the distribution's parameter α lies from 0.3 to 1.99. If generated no. is lies outside this range, error message will display. The Mantegna algorithm has following three phases:

First is to calculate

$$v = \frac{x}{|y|^{1/\alpha}}$$

Here x and *y* are normally distributed variables with standard deviations, respectively. The value of variable x and y is calculated with the help of following formula:

$$\sigma_x = \left[\frac{\Gamma(1+\alpha)\sin(\pi\alpha/2)}{\Gamma((1+\alpha)/2\alpha 2^{(\alpha-1)/2}} \right]^{1/\alpha}$$

$$\sigma_y = 1 .$$

With the help of this Mantegna algorithm the no. which is generated is use further to find out next city that has to visit by ants.

Chapter 7

RESULTS AND DISCUSSION

Experimental Setup

In each and every algorithm control parameters plays very important role in efficient working of algorithm. Hence, there are some control parameters for ant colony optimization algorithm with cuckoo search which is our proposed algorithm. We did an extensive literature survey to find out the specific numerical values of the control parameters of proposed algorithm. From this literature survey and experiment we have taken standard and suitable values for proposed algorithm. The control parameters of ACO are α (influence of pheromone on direction), β (influence of adjacent node distance), ρ (pheromone evaporation coefficient) and Q (pheromone depositing factor) and values is taken as 1, 2, 0.5 and 2.0 respectively. In cuckoo search value of p_a is taken 0.25. For mantegna search the value of alpha is taken from 0.3 to 1.99. The next parameter in our experiment is maximum number of population and is taken from 5 to 30. Another control parameter is number of ants and we have taken according to no. of cities. The entire implantation is done in Matlab 2013a.

Experimental Result

We analyse the result obtained by our proposed hybrid algorithm with existing algorithm. To test the efficiency of our algorithm results of our hybrid ACO and CS is compared with real simple ACO algorithm results. In a TSP we already have the information about the number of cities. We have to find out fitness function which is total travelling distance and sequence of path which will provide us the optimal results. We conducted the experiment by changing the number of cities and then compared our results with that of ACO algorithm.

1. In the first experiment we have taken 5 cities. The distance between each pair of cities is given following by a symmetric matrix of 5*5:

```
D= [  0  18   3  12  10
     18   0  10   5  17
      3  10   0   9  10
     12   5   9   0  12
     10  17  10  12   0 ];
```

Time taken by ACO is 1.1572 second and by proposed algorithm is 0.9499 second. The snapshot of result is shown in fig 7.1 and fig 7.2.

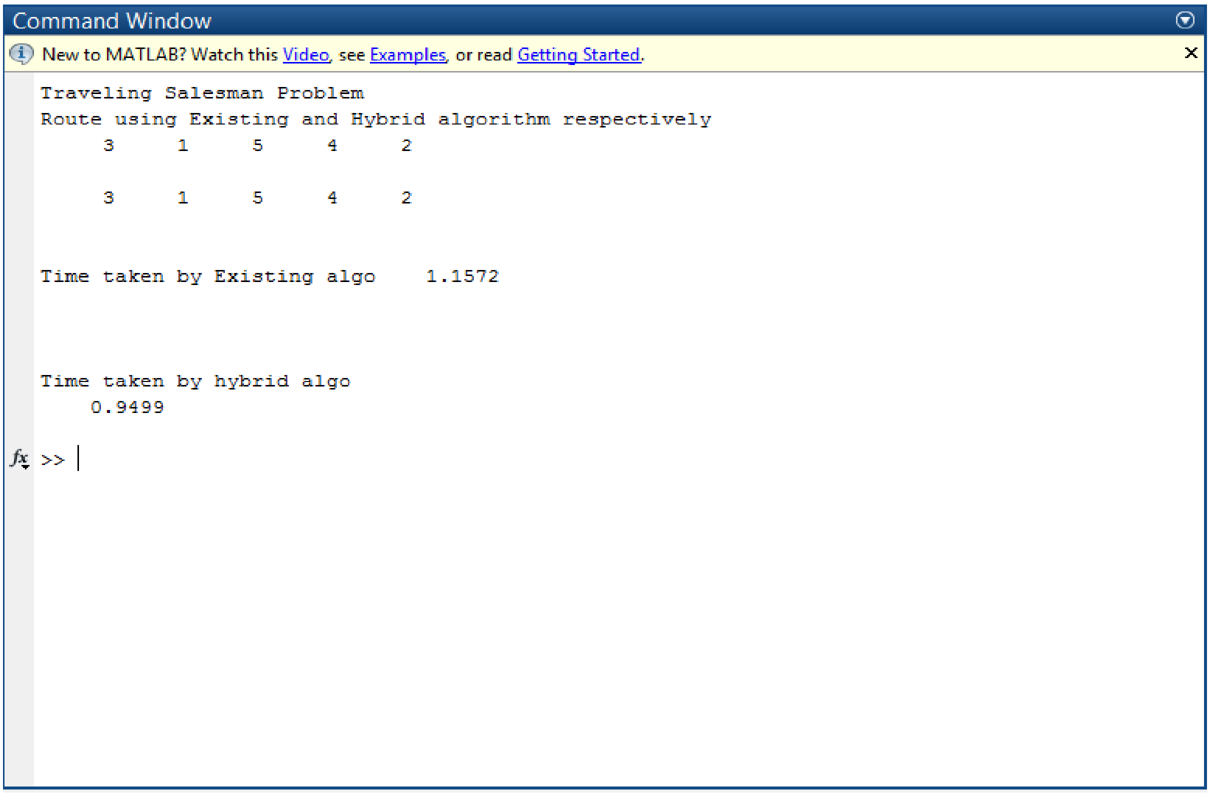


Fig 7.1- Time taken by ACO and Hybrid algorithm to travel 5 cities

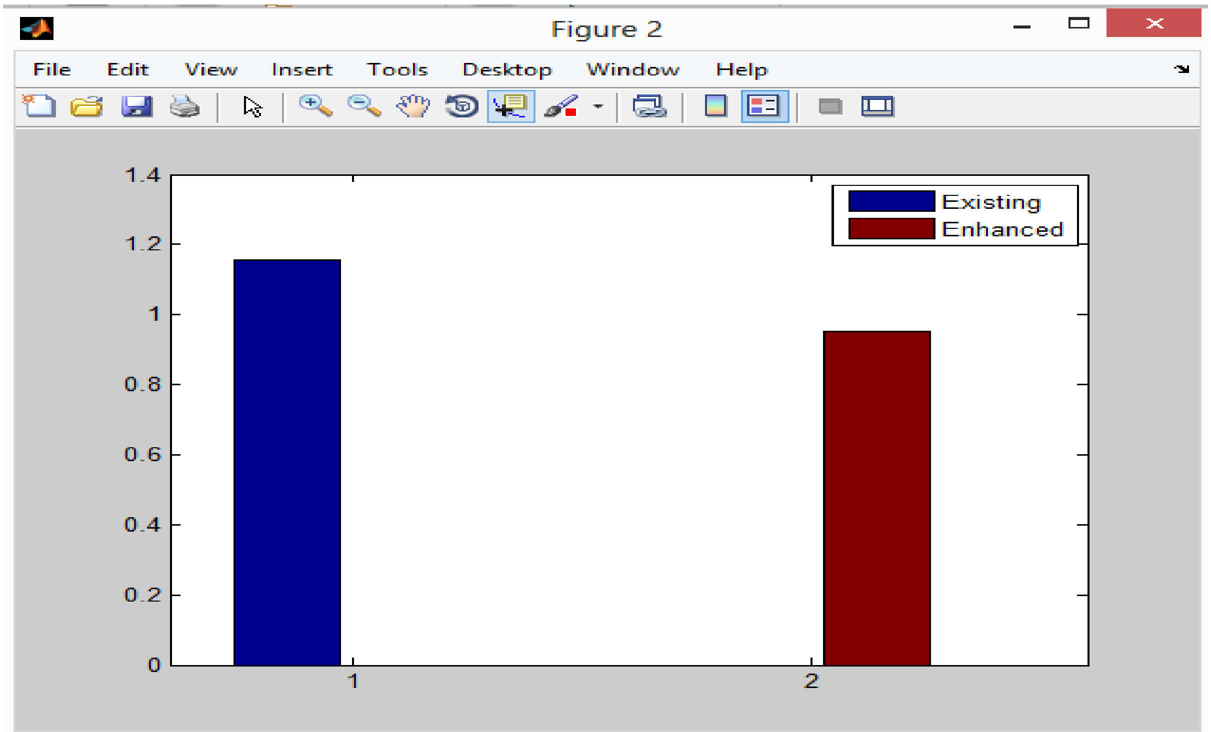


Fig 7.2- Bar chart of time taken by ACO and Hybrid algorithm to travel 5 cities

2. In the second experiment we have taken 10 cities. The distance between each pair of cities is given following by a symmetric matrix of 10*10:

```
D=[ 0  18   3  12  10   7  10   8   9  10
   18   0  10   5  17  12   7  12  13  19
    3  10   0   9  10   9  14  16  12   6
   12   5   9   0  12  10  13   9   5  11
   10  17  10  12   0  15   7  13  10   4
    7  12   9  10  15   0   7  14  10   7
   10   7  14  13   7   7   0   8   9   9
    8  12  16   9  13  14   8   0   9   6
    9  13  12   5  10  10   9   9   0   9
   10  19   6  11   4   7   9   6   9   0]
```

Time taken by ACO is 5.7805 second and by proposed algorithm is 3.9142 second. The snapshot of result is shown in fig 7.3 and fig 7.4.

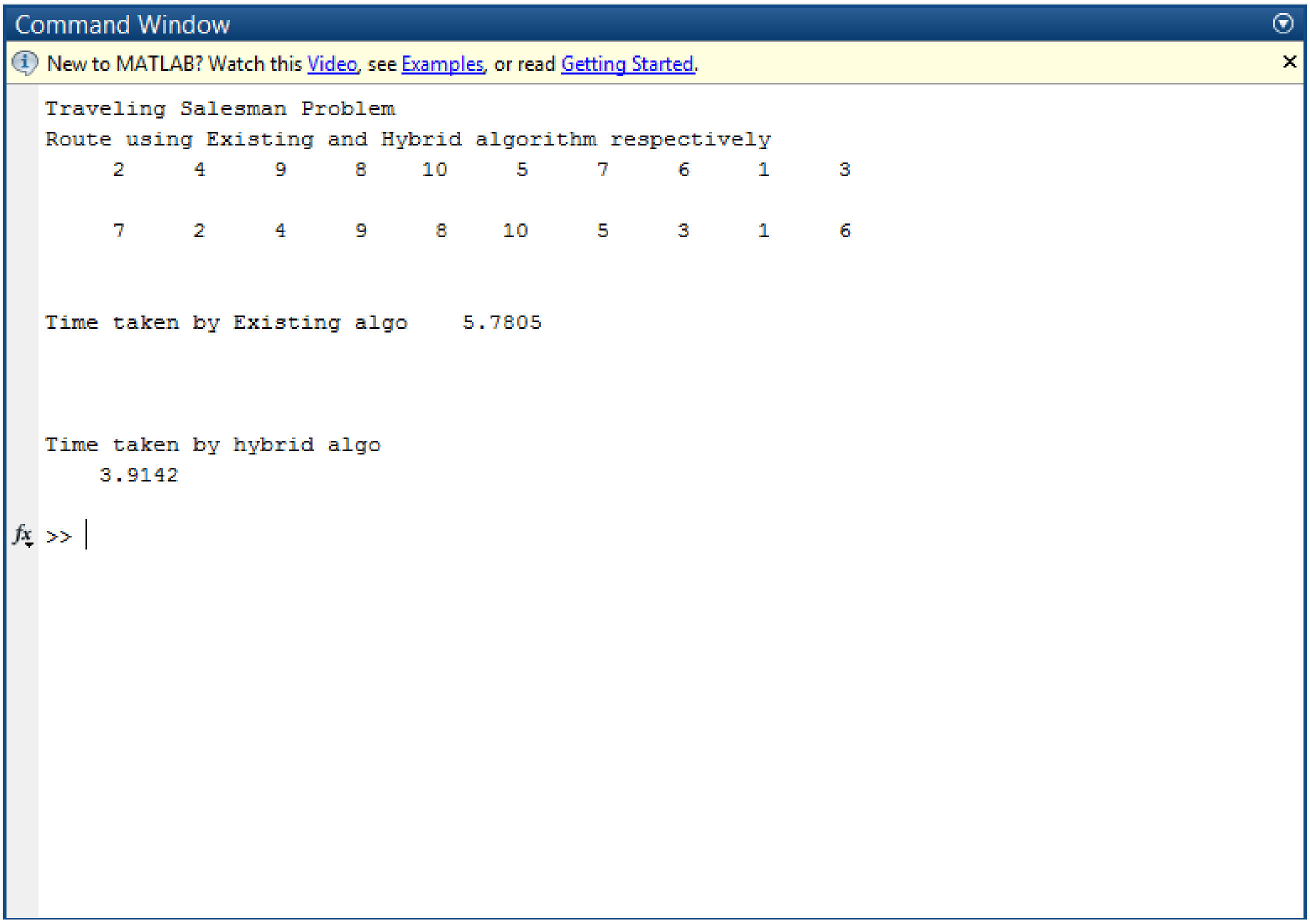


Fig7.3- Time taken by ACO and Hybrid algorithm to travel 10 cities

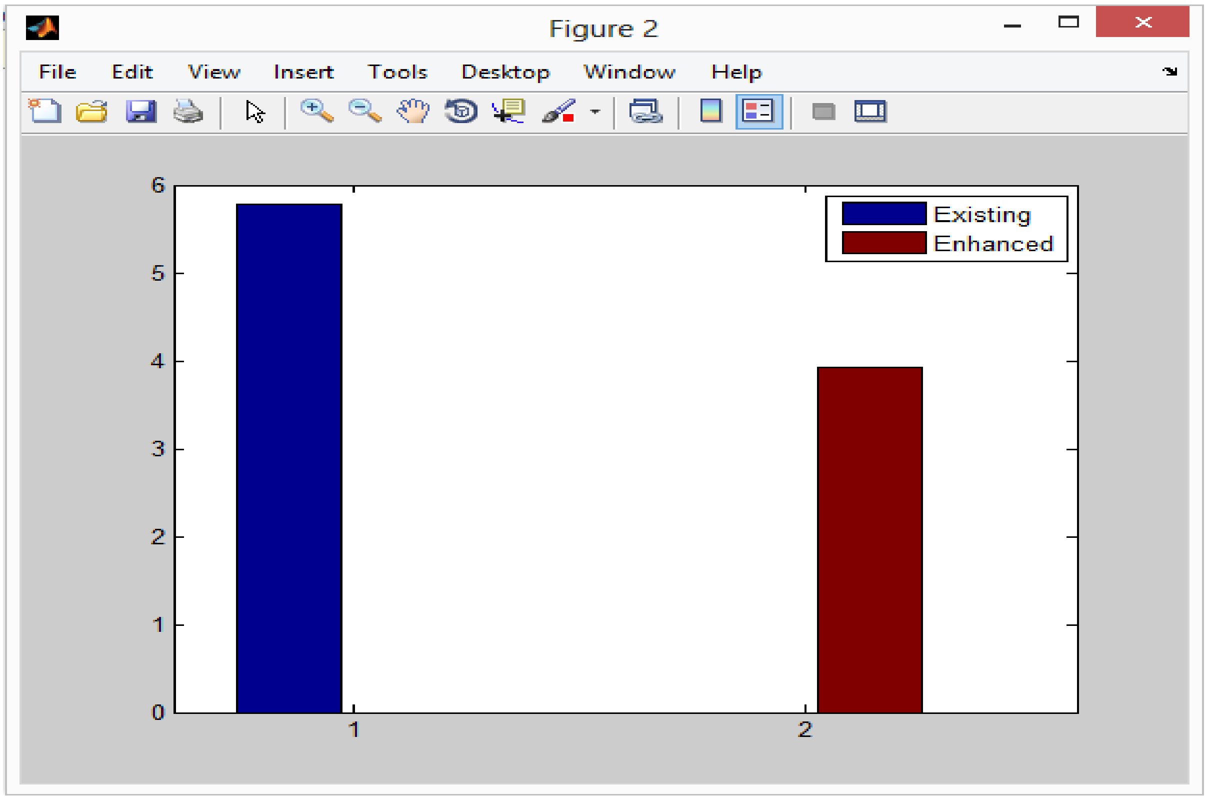


Fig7.4- Bar chart of time taken by ACO and Hybrid algorithm to travel 5 cities

3. In the third experiment we have taken 15 cities. The distance between each pair of cities is given following by a symmetric matrix of 15*15:

```
D=[ 0 18 3 12 10 7 10 8 9 10 10 12 16 16 10
    18 0 10 5 17 12 7 12 13 19 9 18 5 11 7
    3 10 0 9 10 9 14 16 12 6 10 11 10 7 7
    12 5 9 0 12 10 13 9 5 11 11 12 10 12 5
    10 17 10 12 0 15 7 13 10 4 2 3 8 11 19
    7 12 9 10 15 0 7 14 10 7 8 6 12 6 3
    10 7 14 13 7 7 0 8 9 9 14 14 7 15 11
    8 12 16 9 13 14 8 0 9 6 10 10 6 12 13
    9 13 12 5 10 10 9 9 0 9 11 10 8 14 5
    10 19 6 11 4 7 9 6 9 0 11 8 14 12 8
    10 9 10 11 2 8 14 10 11 11 0 11 1 5 6
    12 18 11 12 3 6 14 10 10 8 11 0 15 6 11
    16 5 10 10 8 12 7 6 8 14 1 15 0 9 12
    16 11 7 12 11 6 15 12 14 12 5 6 9 0 3
    10 7 7 5 19 3 11 13 5 8 6 11 12 3 0];
```

Time taken by ACO is 14.9135 second and by proposed algorithm is 9.5106 second. The snapshot of result is shown in fig 7.5 and fig 7.6.

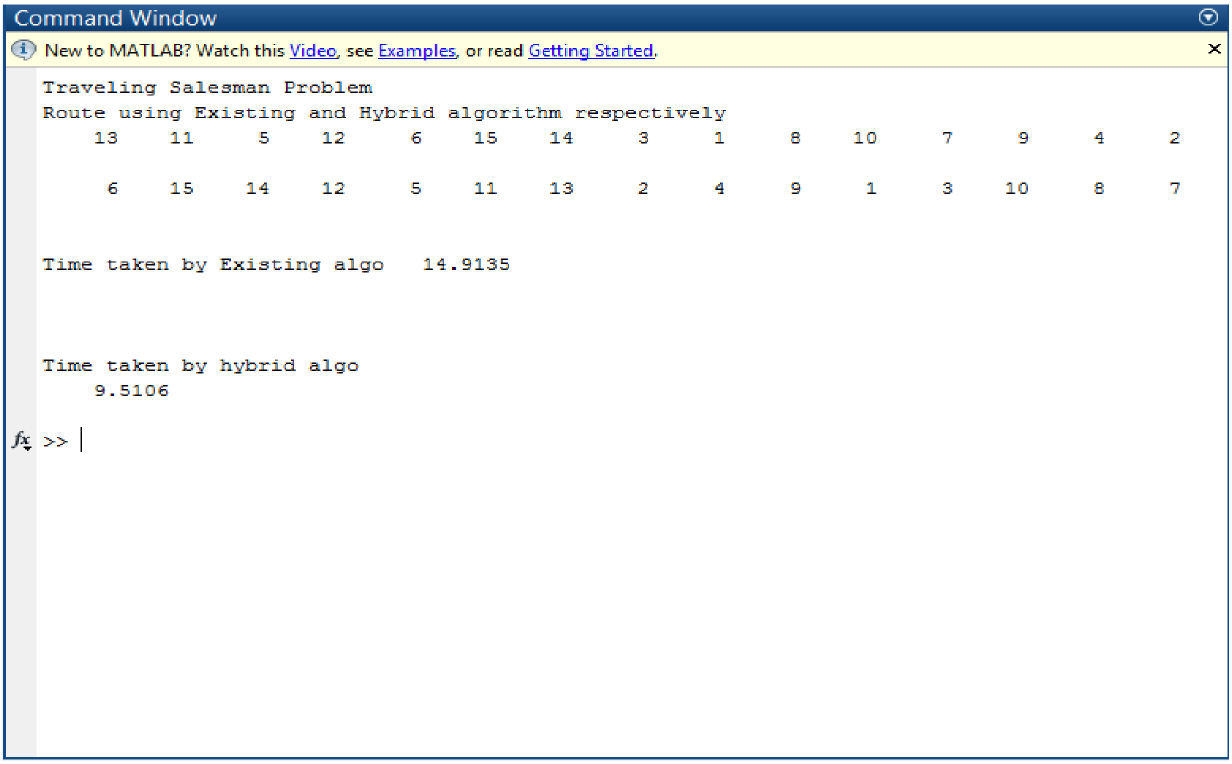


Fig7.5- Time taken by ACO and Hybrid algorithm to travel 15 cities

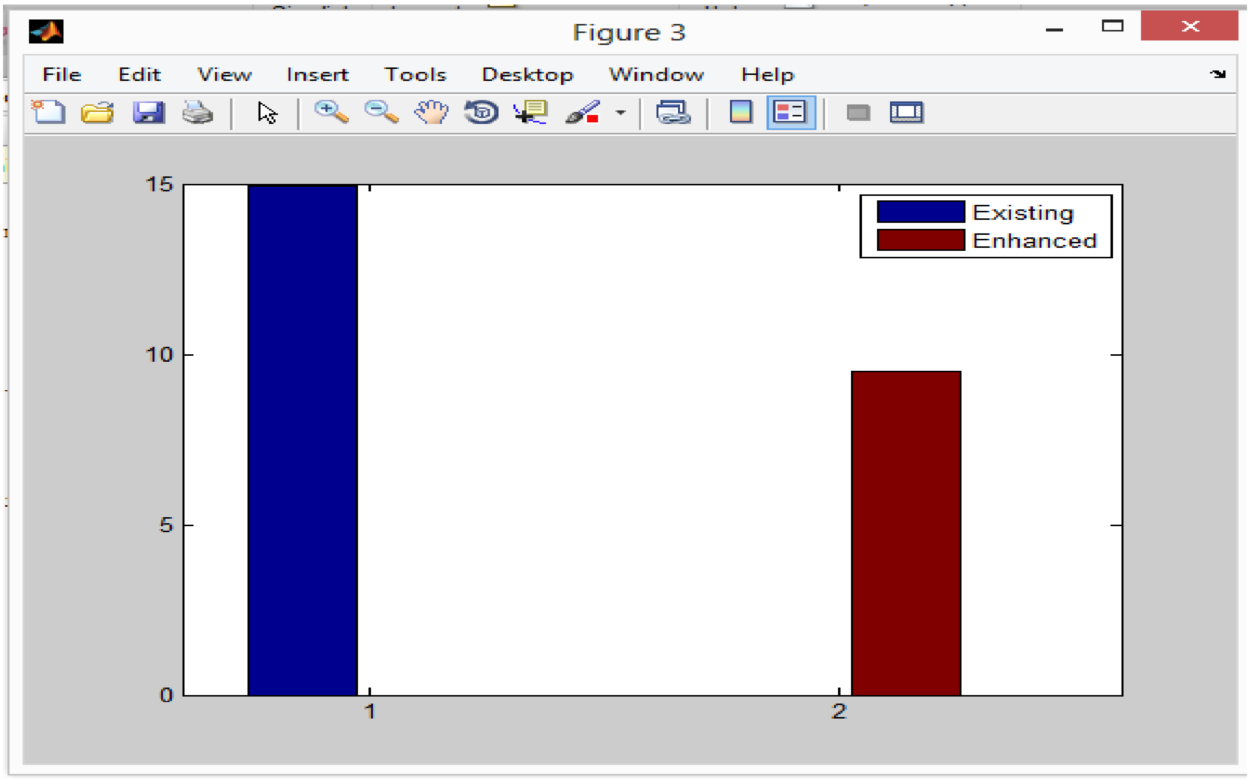


Fig7.6- Bar chart of time taken by ACO and Hybrid algorithm to travel 15 cities

4. In the fourth experiment we have taken 20 cities. The distance between each pair of cities is given following by a symmetric matrix of 20*20:

```
D=[ 0 18 3 12 10 7 10 8 9 10 10 12 16 16 10 9 5 3 13 12
18 0 10 5 17 12 7 12 13 19 9 18 5 11 7 11 11 15 9 11
3 10 0 9 10 9 14 16 12 6 10 11 10 7 7 18 13 13 12 14
12 5 9 0 12 10 13 9 5 11 11 12 10 12 5 11 10 17 8 14
10 17 10 12 0 15 7 13 10 4 2 3 8 11 19 7 3 17 7 10
7 12 9 10 15 0 7 14 10 7 8 6 12 6 3 6 6 11 16 9
10 7 14 13 7 7 0 8 9 9 14 14 7 15 11 4 12 12 9 14
8 12 16 9 13 14 8 0 9 6 10 10 6 12 13 7 12 6 10 13
9 13 12 5 10 10 9 9 0 9 11 10 8 14 5 10 6 10 7 8
10 19 6 11 4 7 9 6 9 0 11 8 14 12 8 8 6 6 11 5
10 9 10 11 2 8 14 10 11 11 0 11 1 5 6 8 5 4 11 14
12 18 11 12 3 6 14 10 10 8 11 0 15 6 11 9 3 10 10 8
16 5 10 10 8 12 7 6 8 14 1 15 0 9 12 9 9 8 17 12
16 11 7 12 11 6 15 12 14 12 5 6 9 0 3 13 13 11 5 6
10 7 7 5 19 3 11 13 5 8 6 11 12 3 0 6 6 7 12 6
9 11 18 11 7 6 4 7 10 8 8 9 9 13 6 0 14 8 7 8
5 11 13 10 3 6 12 12 6 6 5 3 9 13 6 14 0 12 8 10
3 15 13 17 17 11 12 6 10 6 4 10 8 11 7 8 12 0 2 12
13 9 12 8 7 16 9 10 7 11 11 10 17 5 12 7 8 2 0 19
12 11 14 14 10 9 14 13 8 5 14 8 12 6 6 8 10 12 19 0]
```

Time taken by ACO is 30.4802 second and by proposed algorithm is 17.9556 second. The snapshot of result is shown in fig 7.7 and fig 7.8.

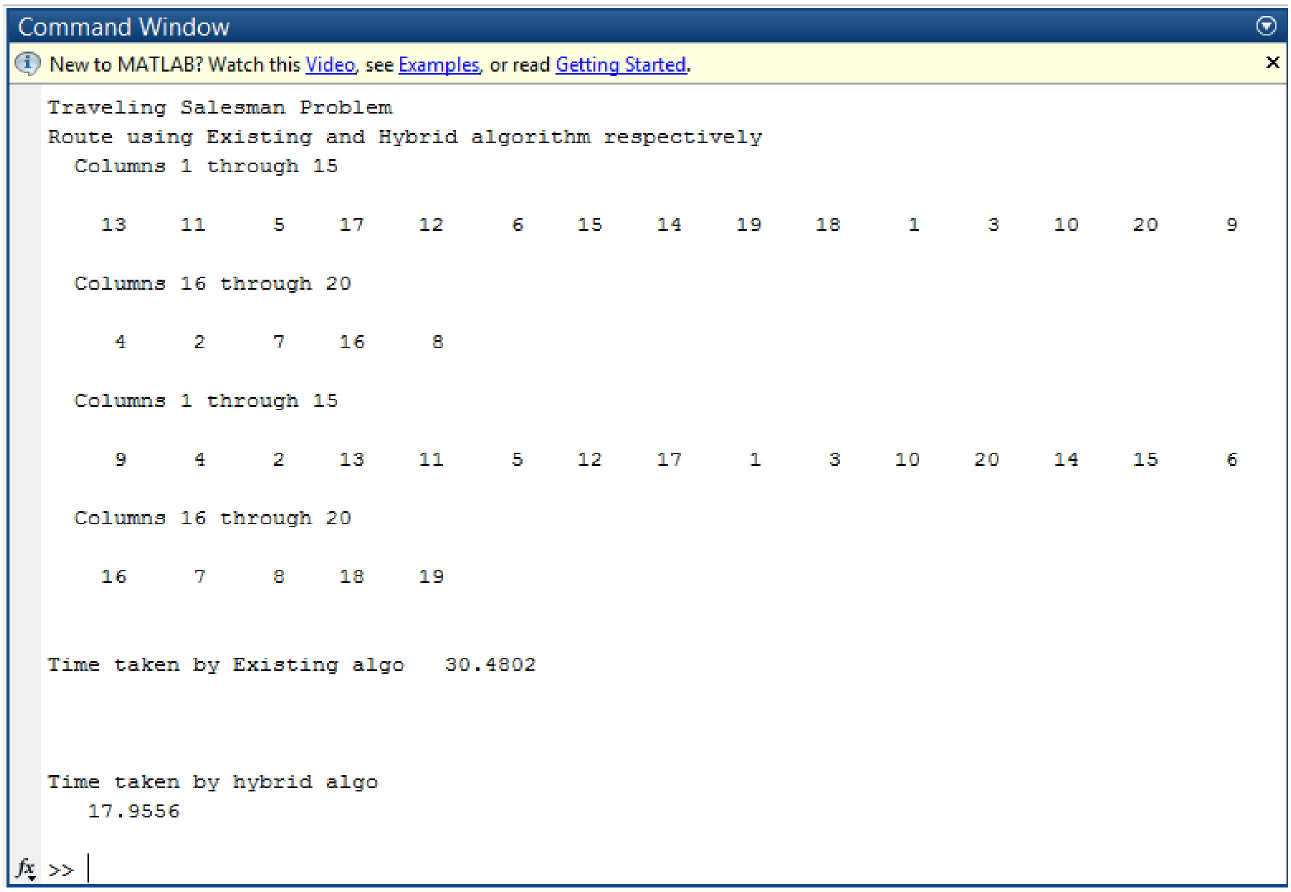


Fig 7.7- Time taken by ACO and Hybrid algorithm to travel 20 cities

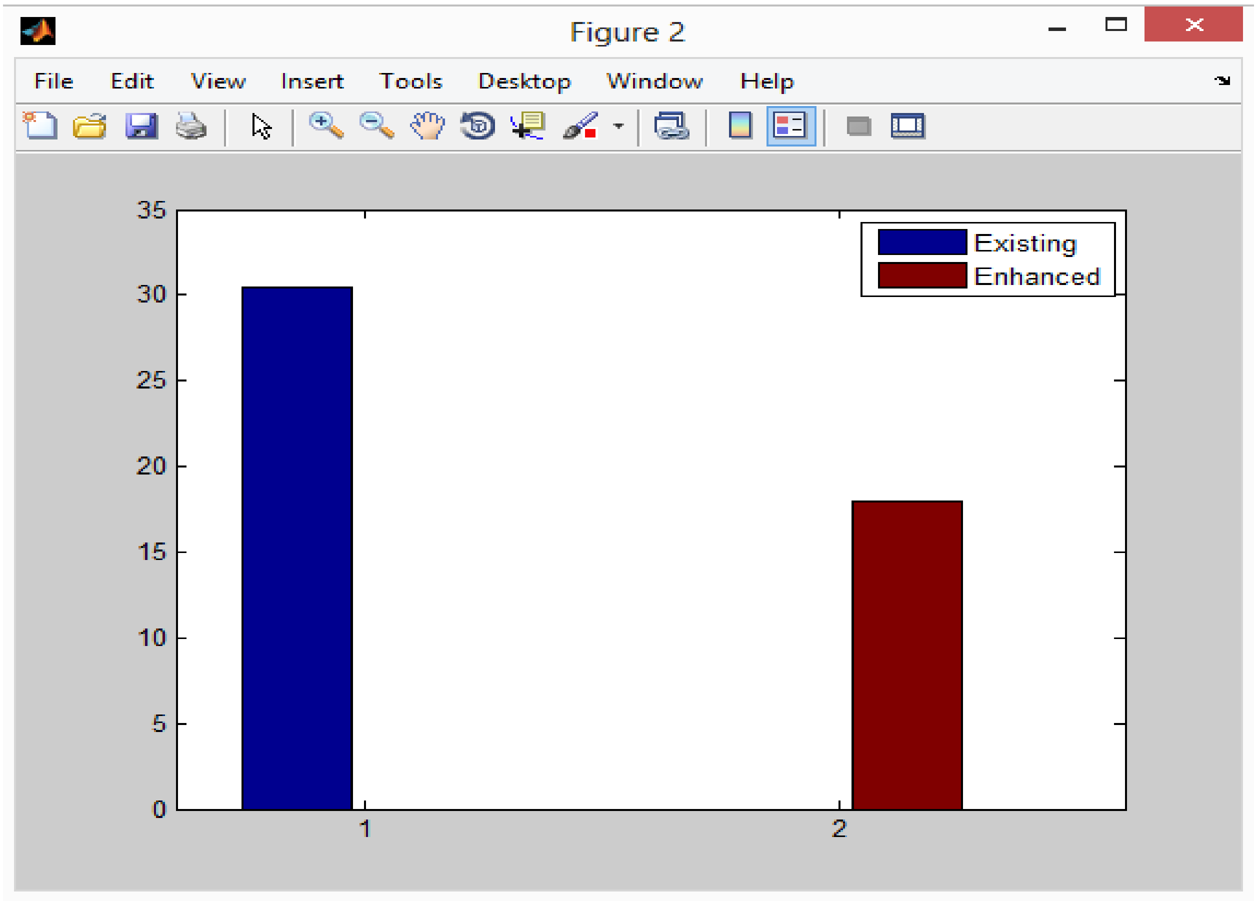


Fig 7.8- Bar chart of time taken by ACO and Hybrid algorithm to travel 20 cities

5. In the fourth experiment we have taken 25 cities. The distance between each pair of cities is given following by a symmetric matrix of 25*25:

D=[0 18 3 12 10 7 10 8 9 10 10 12 16 16 10 9 5 3 13 12 11 12 5 13 6
18 0 10 5 17 12 7 12 13 19 9 18 5 11 7 11 11 15 9 11 13 7 8 9 14
3 10 0 9 10 9 14 16 12 6 10 11 10 7 7 18 13 13 12 14 14 10 11 13 17
12 5 9 0 12 10 13 9 5 11 11 12 10 12 5 11 10 17 8 14 9 13 16 9 9
10 17 10 12 0 15 7 13 10 4 2 3 8 11 19 7 3 17 7 10 6 10 16 3 3
7 12 9 10 15 0 7 14 10 7 8 6 12 6 3 6 6 11 16 9 9 13 11 18 14
10 7 14 13 7 7 0 8 9 9 14 14 7 15 11 4 12 12 9 14 5 13 14 16 6
8 12 16 9 13 14 8 0 9 6 10 10 6 12 13 7 12 6 10 13 15 16 9 6 9
9 13 12 5 10 10 9 9 0 9 11 10 8 14 5 10 6 10 7 8 15 10 16 9 8
10 19 6 11 4 7 9 6 9 0 11 8 14 12 8 8 6 6 11 5 12 8 11 6 17
10 9 10 11 2 8 14 10 11 11 0 11 1 5 6 8 5 4 11 14 12 19 11 10 11
12 18 11 12 3 6 14 10 10 8 11 0 15 6 11 9 3 10 10 8 11 3 10 7 17
16 5 10 10 8 12 7 6 8 14 1 15 0 9 12 9 9 8 17 12 4 12 9 18 11
16 11 7 12 11 6 15 12 14 12 5 6 9 0 3 13 13 11 5 6 18 7 10 8 13
10 7 7 5 19 3 11 13 5 8 6 11 12 3 0 6 6 7 12 6 7 1 12 12 9

9 11 18 11 7 6 4 7 10 8 8 9 9 13 6 0 14 8 7 8 14 3 4 13 7

5 11 13 10 3 6 12 12 6 6 5 3 9 13 6 14 0 12 8 10 17 12 13 3 9

3 15 13 17 17 11 12 6 10 6 4 10 8 11 7 8 12 0 2 12 1 1 7 6 6

13 9 12 8 7 16 9 10 7 11 11 10 17 5 12 7 8 2 0 19 5 10 12 2 4

12 11 14 14 10 9 14 13 8 5 14 8 12 6 6 8 10 12 19 0 9 8 11 13 4

11 13 14 9 6 9 5 15 15 12 12 11 4 18 7 14 17 1 5 9 0 5 8 10 10

12 7 10 13 10 13 13 16 10 8 19 3 12 7 1 3 12 1 10 8 5 0 3 16 10

5 8 11 16 16 11 14 9 16 11 11 10 9 10 12 4 13 7 12 11 8 3 0 11 12

13 9 13 9 3 18 16 6 9 6 10 7 18 8 12 13 3 6 2 13 10 16 11 0 8

6 14 17 9 3 14 6 9 8 17 11 17 11 13 9 7 9 6 4 4 10 10 12 8 0]

Time taken by ACO is 53.6349 second and by proposed algorithm is 27.7136 second. The snapshot of result is shown in fig 7.9 and fig 7.10.

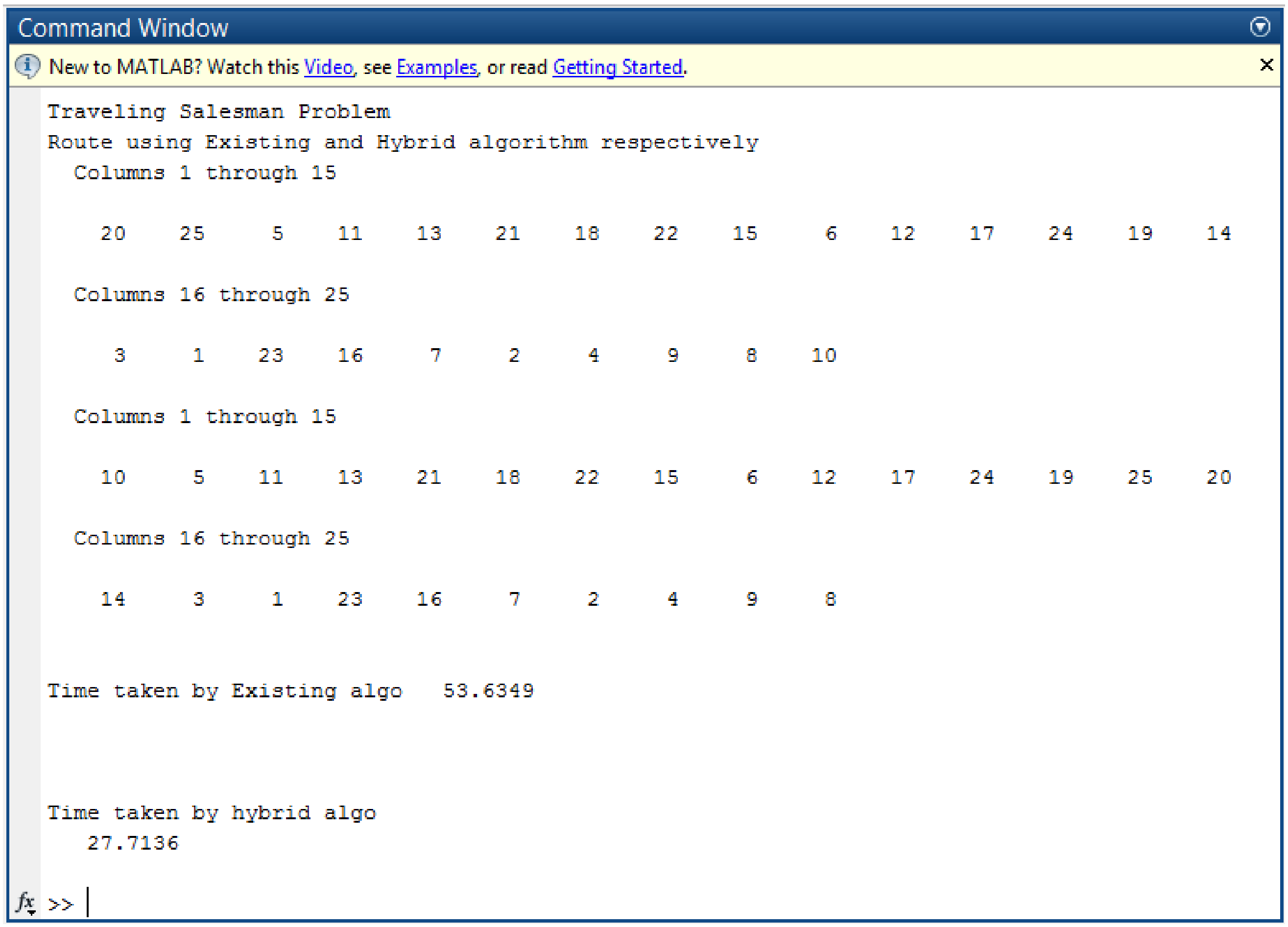


Fig 7.9- Time taken by ACO and Hybrid algorithm to travel 25 cities

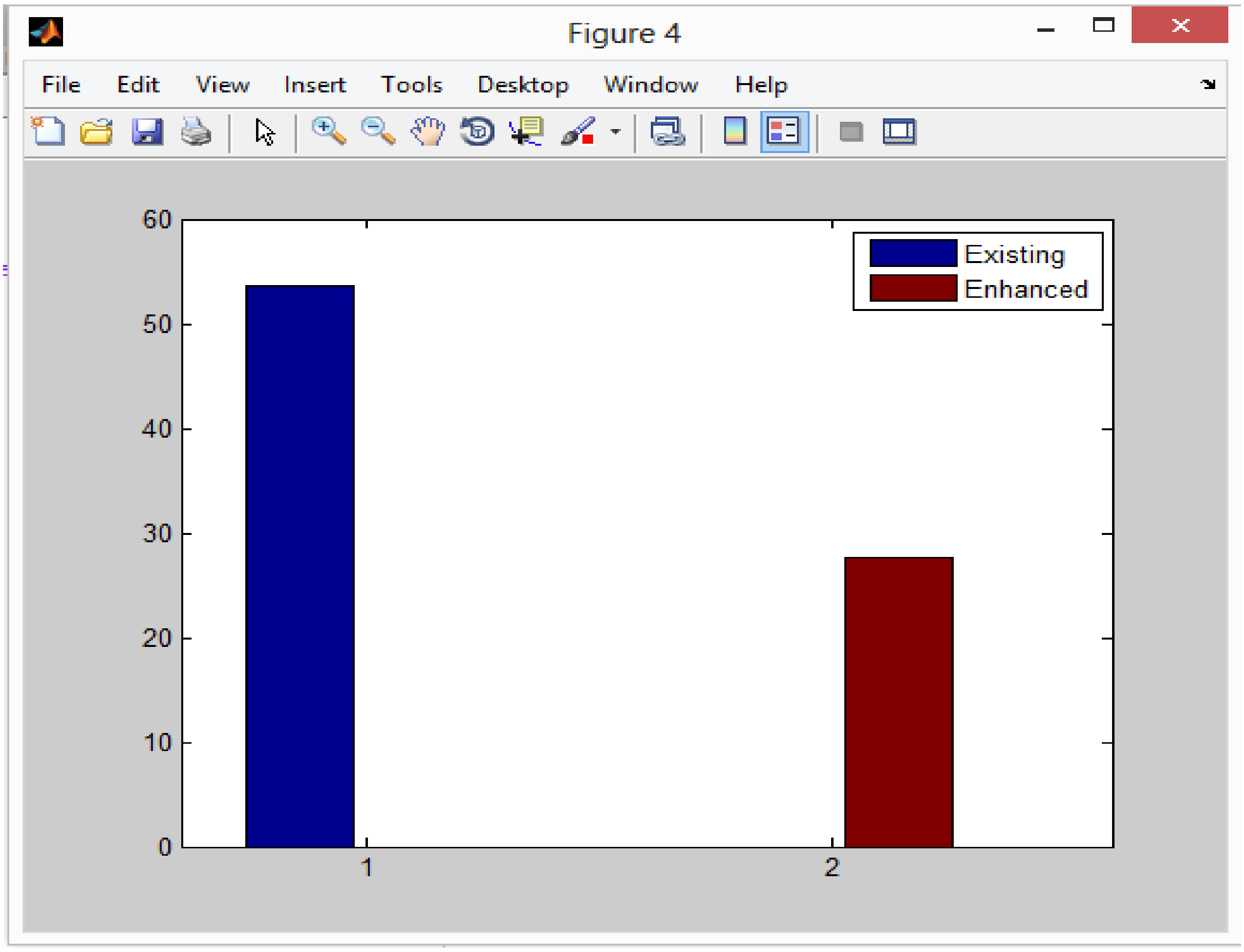


Fig7.10- Bar chart of time taken by ACO and Hybrid algorithm to travel 25 cities

6. In the fourth experiment we have taken 30 cities. The distance between each pair of cities is given following by a symmetric matrix of 30*30:

[0 18 3 12 10 7 10 8 9 10 10 12 16 16 10 9 5 3 13 12 11 12 5 13 6 9 11 3 16 5

18 0 10 5 17 12 7 12 13 19 9 18 5 11 7 11 11 15 9 11 13 7 8 9 14 4 9 10 11 7

3 10 0 9 10 9 14 16 12 6 10 11 10 7 7 18 13 13 12 14 14 10 11 13 17 8 12 6 15 13

12 5 9 0 12 10 13 9 5 11 11 12 10 12 5 11 10 17 8 14 9 13 16 9 9 6 14 3 7 9

10 17 10 12 0 15 7 13 10 4 2 3 8 11 19 7 3 17 7 10 6 10 16 3 3 11 17 4 13 5

7 12 9 10 15 0 7 14 10 7 8 6 12 6 3 6 6 11 16 9 9 13 11 18 14 9 12 8 18 11

10 7 14 13 7 7 0 8 9 9 14 14 7 15 11 4 12 12 9 14 5 13 14 16 6 5 11 17 14 3

8 12 16 9 13 14 8 0 9 6 10 10 6 12 13 7 12 6 10 13 15 16 9 6 9 2 9 12 15 4

9 13 12 5 10 10 9 9 0 9 11 10 8 14 5 10 6 10 7 8 15 10 16 9 8 16 4 6 5 17

10 19 6 11 4 7 9 6 9 0 11 8 14 12 8 8 6 6 11 5 12 8 11 6 17 7 15 9 11 14

10 9 10 11 2 8 14 10 11 11 0 11 1 5 6 8 5 4 11 14 12 19 11 10 11 8 12 15 6 11

12 18 11 12 3 6 14 10 10 8 11 0 15 6 11 9 3 10 10 8 11 3 10 7 17 4 13 9 7 14

16 5 10 10 8 12 7 6 8 14 1 15 0 9 12 9 9 8 17 12 4 12 9 18 11 9 6 3 4 12

16 11 7 12 11 6 15 12 14 12 5 6 9 0 3 13 13 11 5 6 18 7 10 8 13 3 11 14 9 10

10 7 7 5 19 3 11 13 5 8 6 11 12 3 0 6 6 7 12 6 7 1 12 12 9 4 15 9 11 19

9 11 18 11 7 6 4 7 10 8 8 9 9 13 6 0 14 8 7 8 14 3 4 13 7 8 7 6 18 6

5 11 13 10 3 6 12 12 6 6 5 3 9 13 6 14 0 12 8 10 17 12 13 3 9 17 13 9 11 11

3 15 13 17 17 11 12 6 10 6 4 10 8 11 7 8 12 0 2 12 1 1 7 6 6 12 18 5 7 4

13 9 12 8 7 16 9 10 7 11 11 10 17 5 12 7 8 2 0 19 5 10 12 2 4 6 14 15 6 7

12 11 14 14 10 9 14 13 8 5 14 8 12 6 6 8 10 12 19 0 9 8 11 13 4 9 15 15 4 9

11 13 14 9 6 9 5 15 15 12 12 11 4 18 7 14 17 1 5 9 0 5 8 10 10 15 5 12 7 6

12 7 10 13 10 13 13 16 10 8 19 3 12 7 1 3 12 1 10 8 5 0 3 16 10 9 11 12 10 9

5 8 11 16 16 11 14 9 16 11 11 10 9 10 12 4 13 7 12 11 8 3 0 11 12 3 6 11 8 0

13 9 13 9 3 18 16 6 9 6 10 7 18 8 12 13 3 6 2 13 10 16 11 0 8 11 7 4 8 7

6 14 17 9 3 14 6 9 8 17 11 17 11 13 9 7 9 6 4 4 10 10 12 8 0 13 4 11 9 12

9 4 8 6 11 9 5 2 16 7 8 4 9 3 4 8 17 12 6 9 15 9 3 11 13 0 5 14 6 13

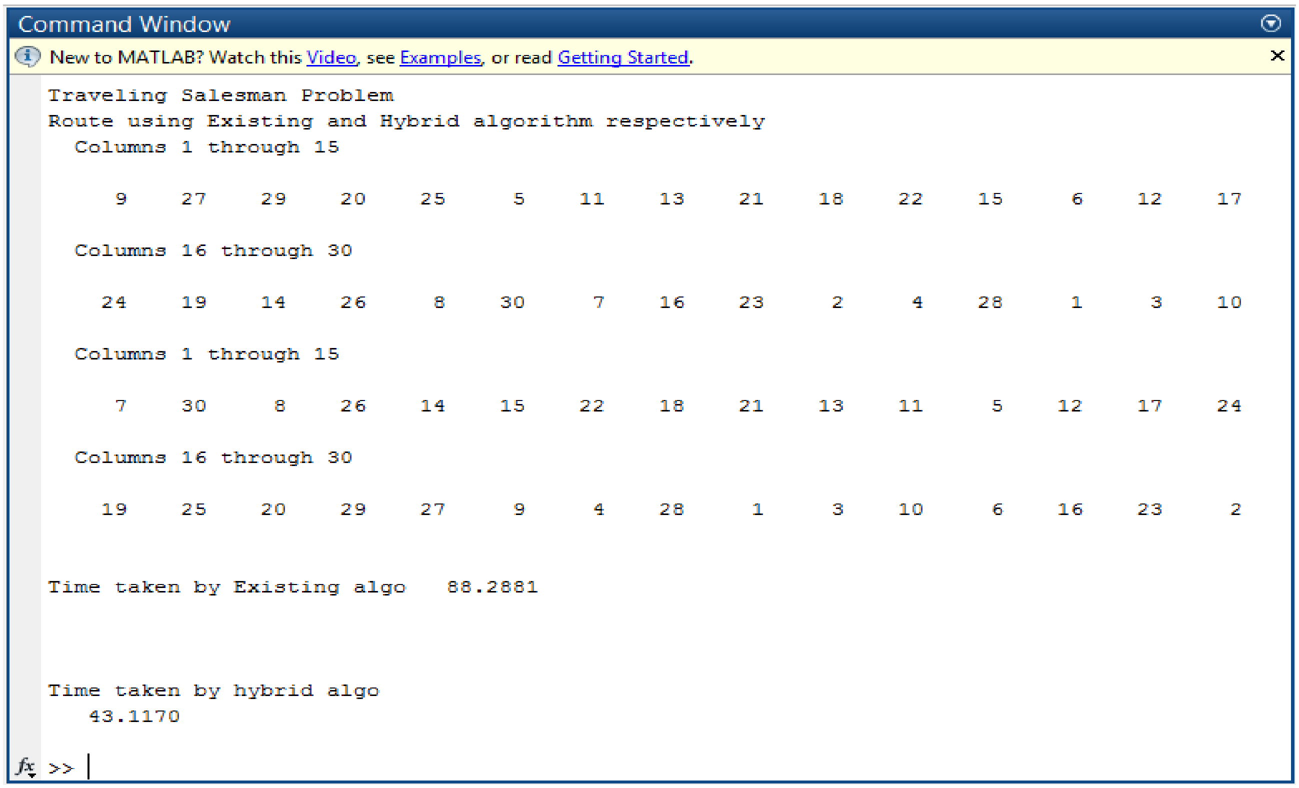
11 9 12 14 17 12 11 9 4 15 12 13 6 11 15 7 13 18 14 15 5 11 6 7 4 5 0 8 4 15

3 10 6 3 4 8 17 12 6 9 15 9 3 14 9 6 9 5 15 15 12 12 11 4 11 14 8 0 9 11

16 11 15 7 13 18 14 15 5 11 6 7 4 9 11 18 11 7 6 4 7 10 8 8 9 6 4 9 0 17

5 7 13 9 5 11 3 4 17 14 11 14 12 10 19 6 11 4 7 9 6 9 0 7 12 13 15 11 17 0];

Time taken by ACO is 88.2881 second and by proposed algorithm is 43.1170 second. The snapshot of result is shown in fig 7.11 and fig 7.12 .



```
Command Window
New to MATLAB? Watch this Video, see Examples, or read Getting Started.
Traveling Salesman Problem
Route using Existing and Hybrid algorithm respectively
  Columns 1 through 15

     9    27    29    20    25     5    11    13    21    18    22    15     6    12    17

  Columns 16 through 30

    24    19    14    26     8    30     7    16    23     2     4    28     1     3    10

  Columns 1 through 15

     7    30     8    26    14    15    22    18    21    13    11     5    12    17    24

  Columns 16 through 30

    19    25    20    29    27     9     4    28     1     3    10     6    16    23     2


Time taken by Existing algo   88.2881



Time taken by hybrid algo
   43.1170

fx >>
```

Fig 7.11- Time taken by ACO and Hybrid algorithm to travel 30 cities

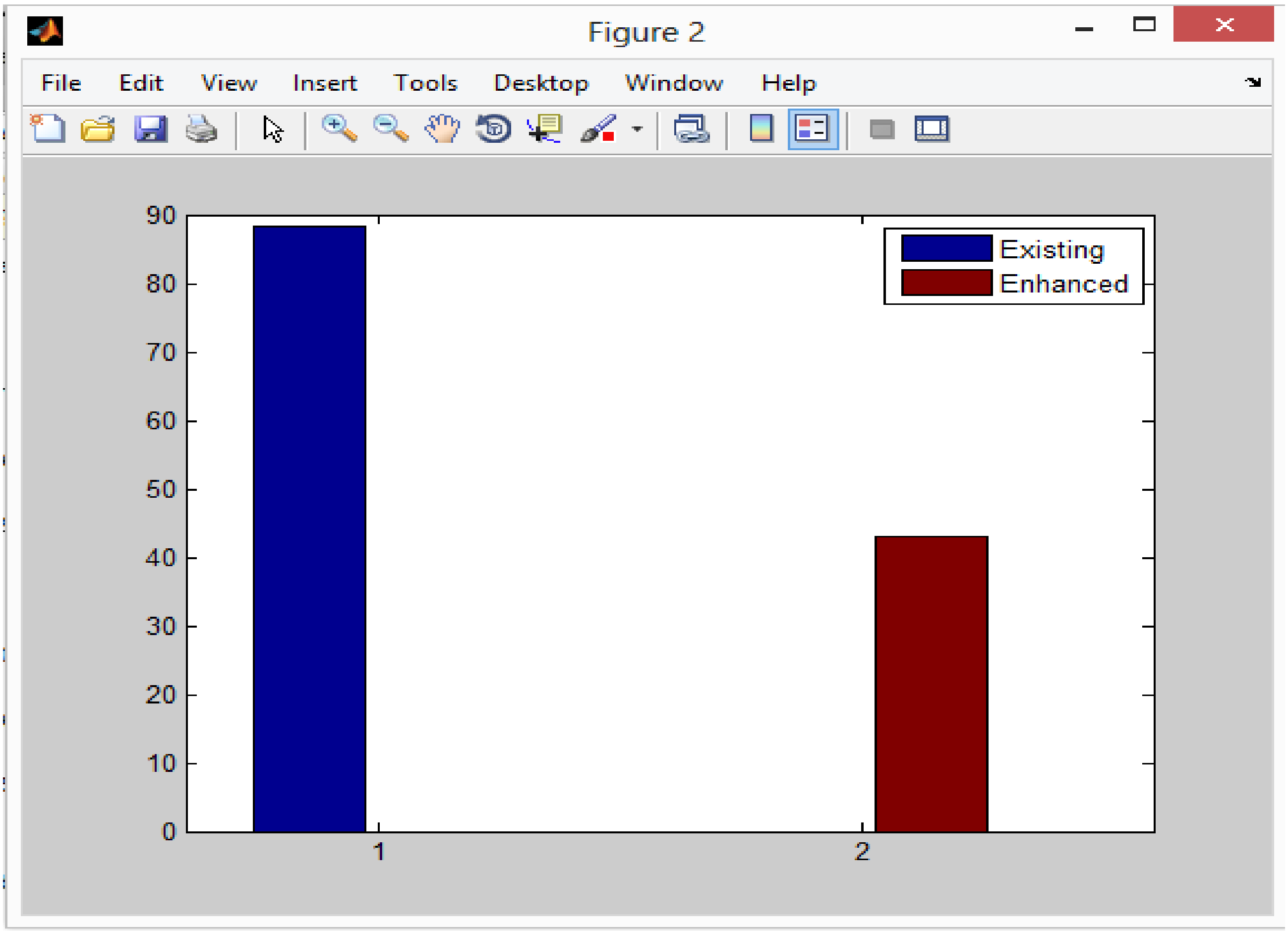


Fig7.12- Bar chart of time taken by ACO and Hybrid algorithm to travel 30 cities

The proposed hybrid algorithm has produced better results than ACO algorithm each time. Complete comparision is shown in following bar chart:

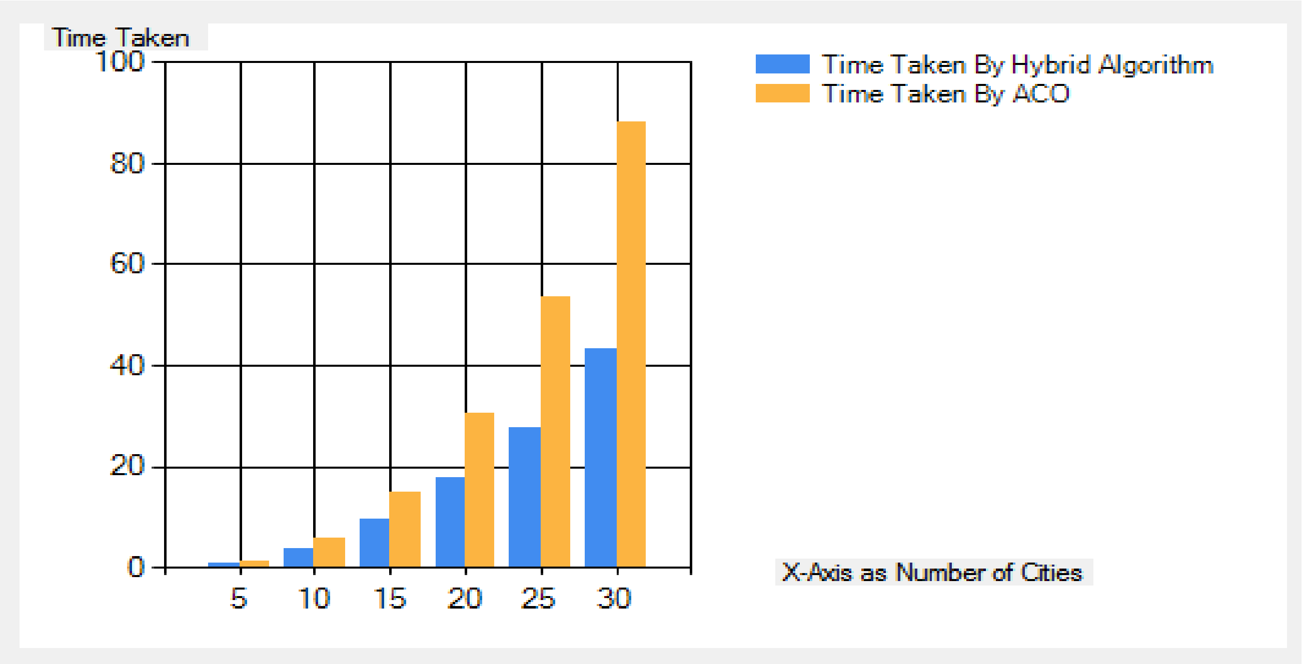


Fig 7.13- Comparision of Hybrid Algorithm and ACO for different no. of cities

After analyzing the comparative results of our proposed hybrid algorithm and existing ACO algorithm we can say that our proposed algorithm has overcome the problem of ACO algorithm. It is performing local search faster than ACO algorithm.

Chapter 8

SUMMARY AND CONCLUSION

Through the study of above research papers we can conclude that the advantage of distributed computing can be get using ACO algorithm. We can easily hybrid this algorithm with any metaheuristic algorithm. It is flexible and robust. ACO algorithm is able to adapt changes easily in real time and can be run continuously. At the same time ACO has some limitations as well which are following:

1. Obviously ACO algorithm can efficiently and easily solve different optimization problem but at the same time we can't prove the convergence of ACO algorithm.

2. Although ACO is good metaheuristic algorithm but drawback of this algorithm is that the ant will pass through that path where pheromone density is high which makes whole process slow. Because of this reason the local search performed by ACO algorithm is not much faster.

So to overcome this problem we have applied Cuckoo Search with ACO algorithm. It has carried out the problem of local search of ACO. The main advantage of Cuckoo search is that it takes only one parameter p_a (probability function) with population size. Very less number of parameters makes Cuckoo Search algorithm so efficient and simple than other metaheuristic algorithm. By mingling ACO with Cuckoo Search it will not depend on pheromone density to choose the next city which will make the process faster.

After applying the cuckoo search algorithm with ACO the problem of local search has been removed. The results of hybrid algorithm are better than simple ACO algorithm.

REFERENCES

Research Papers

1. Nada M. A. Al Salami, “Ant Colony Optimization Algorithm, UbiCC Journal”, Volume 4, Number 3, August 2009.
2. Utkarsh Jaiswal and Shweta Aggarwal, “Ant Colony Optimization”, In: International Journal of Scientific & Engineering Research Volume 2, Issue 7, ISSN 2229-5518, July-2011.
3. John E. Bell and Patrick R. McMullen, “Ant colony optimization techniques for the vehicle routing problem”, In: Advanced Engineering Informatic, vol 18 pp. 41–48, Elsevier 2004.
4. M. Duran Toksari, “Ant colony optimization for finding the global minimum”, In:Applied Mathematics and Computation, vol 176 pp. 308–316, Elsevier 2006.
5. Majid Yousefikhoshbakht, Farzad Didehvar and Farhad Rahmati, “Modification of the Ant Colony Optimization for Solving the Multiple Traveling Salesman Problem”, In: Romanian Journal Of Information Science and Technology. Volume 16, Number 1, pp.65–80, 2013.
6. Hui Yu, “Optimized Ant Colony Algorithm by Local Pheromone Update”, in: TELKOMNIKA Indonesian Journal of Electrical Engineering, Vol.12, No.2, pp. 984 - 990, February 2014.
7. Isa Maleki and Seyyed Reza Khaze, “A New Solution for Dynamic Travelling Salesman Problem with Hybrid Ant Colony Optimization Algorithm and Chaos Theory”, in: International Journal of Advanced Research in Computer Science, Volume 3, No. 7, ISSN No. 0976-5697, Nov-Dec 2012.
8. Ivan Brezina Jr. Zuzana Cickova, “Solving the Travelling Salesman Problem Using the Ant Colony Optimization, Management Information Systems”,Vol No. 4, pp. 010-014, 2011.
9. Zar Chi Su Su Hlaing and May Aye Khine, “An Ant Colony Optimization Algorithm for Solving Traveling Salesman Problem” in: International Conference on Information Communication and Management IPCSIT vol.16, 2011.
10. Xin-She Yang and Suash Deb, “Cuckoo Search via Levy Flights”, in: IEEE, 978-1-4244-5612-3, 2009.
11. Aziz Ouaarab,Belaid Ahiod and Xin-She Yang, “Discrete cuckoo search algorithm for the travelling salesman Problem”, in: Springer-Verlag London, vol 24, pp. 1659–1669, 2014.
12. Ramin Rajabioun, “Cuckoo Optimization Algorithm”, in: Applied Soft Computing , pp.5508–5518, Elsevier 2011.
13. Li-Pei Wong, Malcolm Yoke Hean Low and Chin Soon Chong, “ Bee Colony Optimization with Local Search for Traveling Salesman Problem”, In: Applied Mathematics and Computation, vol. 189 pp.312–319, Elsevier 2008.

14. Sangita Roy, Sheli Sinha Chaudhuri “Cuckoo Search Algorithm using Lèvy Flight: A Review” in I.J. Modern Education and Computer Science, 2013, 12, 10-15.
15. J.B. Jona and N. Nagaveni, “Ant-cuckoo Colony Optimization in feature selection ”, in: Pakistan Journal of Biological Sciences, vol. 2, ISSN-1028-8880,pp. 266-271, 2014
16. Belaid Ahiod, Aziz Ouaaraba and Xin Yang “Isolated cuckoo search procedure for the travelling salesman Problem”, in: Springer-Verlag London, vol 24, pp. 1659–1669, 2014.
17. Aziz Ouaarab, Belaid Ahiod and Xin-She Yang, “ Random-key cuckoo search for the travelling salesman problem”, in: Springer-Verlag Berlin Heidelberg, vol. 24, pp. 1826–1829, 2014.
18. Byung-In Kim, Jae-Ik Shim and Min Zhang, “Comparison of TSP Algorithms ”, in : Project for Models in Facilities Planning and Materials Handling, Vol 1, pp. 289-293, 1998.
19. Ala'a Abu-Srhan and Essam Al Daoud” A Hybrid Algorithm Using a Genetic Algorithm and Cuckoo Search Algorithm to Solve the Traveling Salesman Problem and its Application to Multiple Sequence Alignment”,in International Journal of Advanced Science and Technology Vol.61, (2013), pp.29-38.
20. M. Z. Rashad, A.E. Keshk, M. A. El-Dosuky “Genetic Cuckoo Optimization Algorithm” in International Journal of Computer Applications (0975 – 8887) Volume 90 – No.3, March 2014.
21. David S. Johnson, Lyle A. McGeoch , “The Traveling Salesman Problem: A Case Study in Local Optimization” 1995.

Websites

- http://en.wikipedia.org/wiki/Ant_colony_optimization_algorithms.
- http://en.wikipedia.org/wiki/Cuckoo_search.
- http://en.wikipedia.org/wiki/TSP.
- http://en.wikipedia.org/wiki/NP-complete.

APPENDIX

A-

Ant Colony Optimization (ACO)

B-

Brood parasite birds

C-

Combinatorial problem

Continuous optimization problem

Convergence rate

Cuckoo Search (CS)

D-

Discrete optimization problem

E-

Edge

Egg laying region

F-

Foraging behaviour

Fitness function

H-

Habitat

Heuristic

Hybrid

L-

Local Search

M-

Metaheuristic algorithms

N-

Node

NP- Complete Problem

O-

Optimization Algorithm

P-

Pheromone

Pheromone evaporation coefficient (ρ)

Pheromone deposit factor (Q)

Population

Probability function

R-

Roundtrip distance

T-

Travelling Salesman Problem