**ANT COLONY OPTIMIZATION**

**SEMINAR REPORT**

***Submitted by:***

1. **GEETHA SRI**

**188T1A0544**

**to**

**the JNTU, Kakinada**

**In partial fulfillment of the requirements for the**

**Award of the degree of**

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE & ENGINEERING**

A close-up of a coin

Description automatically generated with low confidence

**DHANEKULA INSTITUTE OF ENGINEERING & TECHNOLOGY GANGURU, A.P. (INDIA) – 521139**

**(AFFILIATED TO JNTUK, KAKINADA, ANDHRA PRADESH (INDIA), JULY 2011)**

**Department of Computer Science & Engineering**

**Dhanekula Institute Of Engineering & Technology, Ganguru**

A close-up of a coin

Description automatically generated with low confidence

**CERTIFICATE**

Certified that this report entitled ‘Ant Colony Optimization’ is the report of seminar presented by **Pamarthi . Geetha Sri, 188t1a0544** during **2021-2022** in partial fulfillment of the requirements for the award of the Degree of Bachelor of Technology in Computer Science and Engineering of the JNTU, Kakinada.

(Dr. Suresh)

Head of the Department

Dept. Of CSE,

**DECLARATION**

I hereby declare that the seminar entitled “**Ant Colony Optimization**” was carried out and written by me under the guidance of Professor Dr. K. Srinivasa Rao, Department of Computer Science & Engineering, Dhanekula Institute of Engineering and Technology, Ganguru. This work has not been previously formed the basis for the award of any degree or certificate nor has been submitted elsewhere for the award of any degree.

**ABSTRACT**

Swarm intelligence is a relatively new approach to problem solving that takes inspiration from the social behaviors of insects and of other animals. In particular, ants have inspired a number of methods and techniques among which the most studied and the most successful is the general purpose optimization technique known as ant colony optimization. Ant colony optimization (ACO) takes inspiration from the foraging behavior of some ant species. These ants deposit pheromone on the ground in order to mark some favorable path that should be followed by other members of the colony. Ant colony optimization exploits a similar mechanism for solving optimization problems. From the early nineties, when the first ant colony optimization algorithm was proposed, ACO attracted the attention of increasing numbers of researchers and many successful applications are now available. Moreover, a substantial corpus of theoretical results is becoming available that provides useful guidelines to researchers and practitioners in further applications of ACO. The goal of this article is to introduce ant colony optimization and to survey its most notable applications.

**Vision-Mission-PEO**

|  |  |
| --- | --- |
| Institute Vision | Pioneering Professional Education through Quality |
| Institute Mission | Providing Quality Education through state-of-art infrastructure, laboratories and committed staff.  Moulding Students as proficient, competent, and socially responsible engineering personnel with ingenious intellect.  Involving faculty members and student in research and development works for betterment of society. |
| Department Vision | To empower students of Computer Science and Engineering Department to be technologically adept, innovative, global citizens possessing human values. |
| Department Mission | To Encourage students to become self-motivated and problem-solving individuals.  To prepare student for professional career with academic excellence and leadership skills.  To Create Centre’s of excellence in Computer Science and Engineering |
| Program Educational Objectives(PEO’s) | Graduates of Computer Science and Engineering will:  PEO1: Excel in Professional career through knowledge in mathematics and engineering principles.  PEO2: Able to pursue higher education and research.  PEO3: Communicate effectively, recognize and incorporate societal needs in their professional endeavours.  PEO4: Adapt to technological advancements by continuous learning. |

**PROGRAM OUTCOMES**

|  |  |
| --- | --- |
| 1 | **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals and an engineering specialization to the solution of complex engineering problems. |
| 2 | **Problem Analysis:** Identify, formulate, review research literature, and analyse complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences. |
| 3 | **Design /Development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations. |
| 4 | **Conduct investigations of complex problems:** Use research-based knowledge and research method including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions. |
| 5 | **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering an IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations. |
| 6 | **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues, and the consequent responsibilities relevant to the professional engineering practices. |
| 7 | **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts and demonstrate the knowledge of and need for sustainable development. |
| 8 | **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice. |
| 9 | **Individual and teamwork:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings. |
| 10 | **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective report and design documentation, make effective presentations, and give and receive clear instructions. |
| 11 | **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one’s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments. |
| 12 | **Life-long learning:** Recognize the need for and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change. |

**PROGRAM SPECIFIC OUTCOMES**

**PSO1:** Have expertise in algorithms, networking, web applications and software engineering for efficient for efficient design of computer – based systems of varying complexity.

**PSO2:** Qualify in national international level competitive examinations for successful higher studies and employment.

**Project Mappings**

|  |  |
| --- | --- |
| **Batch No:** | 07 |
| **Project Title** | PREDICTION OF BIGMART SALES USING DIFFERENT MACHINE LEARNING ALGORITHMS |
| **Project Domain** | Machine Learning |
| **Type of the Project** | Research |
| **Guide Name** | Ms. S. Maha Lakshmi |
| **Students Roll No:** | **Students Name:** |
| 188T1A0560 | Y. Jothsna |
| 188T1A0555 | T. Gopi Krishna |
| 188T1A0544 | P. Geetha Sri |
| 188T1A0510 | B. Sravani |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Project Title** | **P**  **O**  **1** | **P**  **O**  **2** | **P**  **O**  **3** | **P**  **O**  **4** | **P**  **O**  **5** | **P**  **O**  **6** | **P**  **O**  **7** | **P**  **O**  **8** | **P**  **O**  **9** | **P**  **O**  **1**  **0** | **P**  **O**  **1**  **1** | **P**  **O**  **1**  **2** | **P**  **S**  **O**  **1** | **P**  **S**  **O**  **2** |
| ANT COLONY OPTIMIZATION | 3 | 3 | 3 | 3 | 3 | 1 | 3 | 1 | 3 | 3 | 3 | 3 | 3 | 3 |

|  |  |
| --- | --- |
| **Mapping Level** | **Mapping Description** |
| 1 | Low Level Mapping with PO and PSO |
| 2 | Moderate Mapping with PO and PSO |
| 3 | High Level Mapping with PO and PSO |

**Mapping Justifications:**

**PO1:** Apply the gained domain knowledge in this project.

**PO2:** Compare and contrast the several existing solutions for research challenge.

**PO3:** Demonstrates the performance of the design.

**PO4:** Assesses solution by formulating proper methodology.

**PO5:** Apply appropriate techniques and modern engineering hardware and software tools.

**PO6:** Give reasoning and assess societal, health, legal and cultural issues with competency in professional engineering practice.

**PO7:** Understand the impact of the professional engineering solutions in societal and environmental contexts.

**PO8:** Apply ethical principles and commit to professional ethics and responsibility.

**PO9:** Function effectively as an individual and as a member or leader in project team and reports and presents the finding of the study.

**PO10:** Makes effective presentation and communicate effectively.

**PO11:** Formulate and propose a plane for creating a solution for the project identified.

**PO12:** Demonstrate the understanding of the engineering and management principles in multidisciplinary environments to engage in lifelong learning in the broadest context of technological change.

**Table of Contents**

1. **INTRODUCTION** 
   1. Swarm Intelligence
   2. Ant Colony
   3. Real Ant Behaviour
2. **ANT COLONY OPTIMIZATION**
3. **APPLICATIONS OF ACO**
   1. Travelling Sales man
   2. Quadratic Assignment Problem(QAP)
   3. Network Model
   4. Vehicle Routing Problem with Time Windows
4. **ADVANTAGES OF ACO**
5. **DISADVANTAGES OF ACO**
6. **CONCLUSION**
7. **REFERENCES**

**INTRODUCTION**

**1 Introduction**

**1.1 Swarm Intelligence:**

Swarm intelligence (SI) describes the collective behaviour of decentralized, self

Organized systems, natural or artificial. The concept is employed in work on artificial

intelligence. The expression was introduced by Gerardo Beni and Jing Wang in 1989, in the

context of cellular robotic systems.

Swarm intelligence is the discipline that deals with natural and artificial systems composed

of many individuals that coordinate using decentralized control and self-organization. In

particular, the discipline focuses on the collective behaviours that result from the local

interactions of the individuals with each other and with their environment. Examples of

systems studied by swarm intelligence are colonies of ants and termites, schools of fish,

flocks of birds, herds of land animals. Some human artefacts also fall into the domain of

swarm intelligence, notably some multi-robot systems, and also certain computer programs

that are written to tackle optimization and data analysis problems.

Emphasis is given to such topics as the modelling and analysis of collective biological

systems; application of biological swarm intelligence models to real-world problems; and

theoretical and empirical research in ant colony optimization, particle swarm optimization,

swarm robotics, and other swarm intelligence algorithms. Articles often combine

experimental and theoretical work.



**1.2 Ant Colony:**

The complex social behaviours of ants have been much studied by science, and computer

scientists are now finding that these behaviour patterns can provide models for solving

difficult combinatorial optimization problems. The attempt to develop algorithms inspired by

one aspect of ant behaviour, the ability to find what computer scientists would call shortest

paths, has become the field of ant colony optimization (ACO), the most successful and

widely recognized algorithmic technique based on ant behaviour. This book presents an

overview of this rapidly growing field, from its theoretical inception to practical applications,

including descriptions of many available ACO algorithms and their uses.

The book first describes the translation of observed ant behaviour into working optimization

algorithms. The ant colony metaheuristic is then introduced and viewed in the general context

of combinatorial optimization. This is followed by a detailed description and guide to all

major ACO algorithms and a report on current theoretical findings. The book surveys ACO

applications now in use, including routing, assignment, scheduling, subset, machine learning,

and bioinformatics problems. AntNet, an ACO algorithm designed for the network routing

problem, is described in detail. The authors conclude by summarizing the progress in the field

and outlining future research directions. Each chapter ends with bibliographic material, bullet

points setting out important ideas covered in the chapter, and exercises. *Ant Colony*

*Optimization* will be of interest to academic and industry researchers, graduate students, and

practitioners who wish to learn how to implement ACO algorithms.



**1.3 Real Ant Behavior :**

Natural behaviour of ants have inspired scientists to mimic insect operational methods to

solve real-life complex problems such as Travelling sales man problem, Quadratic

assignment problem, Network model, Vehicle routing. By observing ant behaviour, scientists

have begun to understand their means of communication

Ants communicate with each other through tapping with the antennae and smell. They are

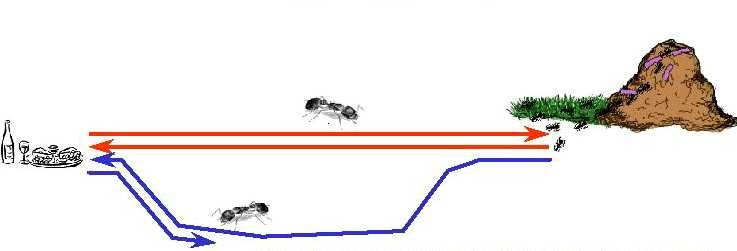
considered, together with the bees, as one of the most socialized animals. They have a perfect

social organization, and each type of individual specializes in a specific activity within the

colony. They are thought by many as having a collective intelligence, and each ant is

considered then as an individual cell of a bigger organism. Ants wander randomly & on

finding food return to their colony while laying “PHEROMONE TRIALS”.



If other Ants find such paths they do not travel randomly but follow the Pheromone trail.

Ants secrete pheromone while travelling from the nest to food and vice versa in order to

communicate with each other to find shortest path.

Pheromone is a highly volatile substance which starts to evaporate, more the time taken by

the ant to travel to and fro more time the pheromone have to evaporate. A shortest path gets

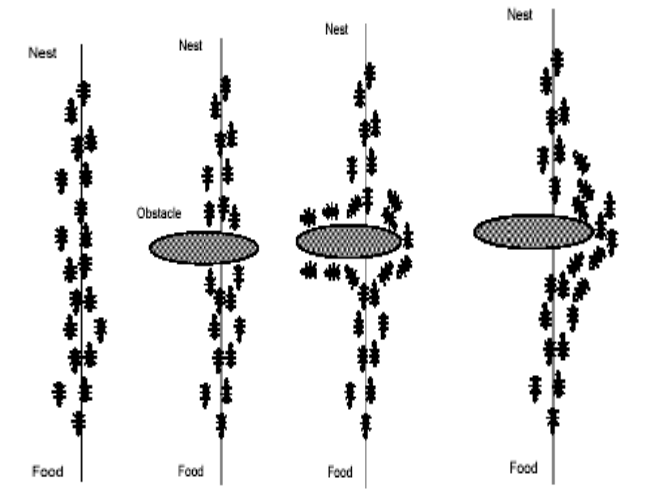
marched over faster and thus the pheromone density remains high. If one of the ant finds the

shortest path from colony to food source other ants are more likely to follow the same path.

**Behaviour of ant in presence of an obstacle:**

Ants are forced to decide whether they should go left or right. The choice that is made is a

random one. Pheromone accumulation is Faster on shortest path.



**Ants:**

One of the first researchers to investigate the social behaviour of insects was the French

entomologist Pierre-Paul Grasse. In the forties and fifties of the 20-th century, he was

observing the behaviour of termites { in particular, the *Bellicositermes natalensis* and

*Cubitermes* species. He discovered [26] that these insects are capable to react to what he

called \significant stimuli", signals that activate a genetically encoded reaction. He observed

that the effects of these reactions can act as new significant stimuli for both the insect that

produced them and for the other insects in the colony. Grasse used the term *stigmergy* to

describe this particular type of indirect communication in which the \workers are stimulated

by the performance they have achieved". The two main characteristics of stigmergy that

di®erentiate it from other means of communication are:

• the physical, non-symbolic nature of the information released by the communicating

insects, which corresponds to a modification of physical environmental states visited

by the insects and

• the local nature of the released information, which can only be accessed by those

insects that visit the place where it was released (or its immediate neighbourhood).

Examples of stigmergy can be observed in colonies of ants. In many ant species, ants walking

to, and from, a food source deposit on the ground a substance called *pheromone*. Other ants

are able to smell this pheromone, and its presence influences the choice of their path i.e., they

tend to follow strong pheromone.

5**2 Ant Colony Optimization(ACO):**

Ant Colony Optimization (ACO) is a paradigm for designing metaheuristic algorithms for

combinatorial optimization problems. The first algorithm which can be classified within this

framework was presented in 1991 and, since then, many diverse variants of the basic principle

have been reported in the literature.

The essential trait of ACO algorithms is the combination of a priori information about the

structure of a promising solution with posterior information about the structure of previously

obtained good solutions. Metaheuristic algorithms are algorithms which, in order to escape

from local optima, drive some basic heuristic: either a constructive heuristic starting from a

null solution and adding elements to build a good complete one, or a local search heuristic

starting from a complete solution and iteratively modifying some of its elements in order to

achieve a better one.

The metaheuristic part permits the low-level heuristic to obtain solutions better than those it

could have achieved alone, even if iterated. Usually, the controlling mechanism is achieved

either by constraining or by randomizing the set of local neighbour solutions to consider in

local search (as is the case of simulated annealing [ or tabu ), or by combining elements taken

by different solutions (as is the case of evolution strategies and genetic or bionomic

algorithms).

The characteristic of ACO algorithms is their explicit use of elements of previous solutions.

In fact, they drive a constructive low-level solution, as GRASP does, but including it in a

population framework and randomizing

the construction in a Monte Carlo way.

A Monte Carlo combination of different

solution elements is suggested also by

Genetic Algorithms , but in the case of

ACO the probability distribution is

explicitly defined by previously obtained

solution components. The particular way

of defining components and associated

probabilities is problem- specific, and

can be designed in different ways, facing a trade-off between the specificity of the

information used for the conditioning and the number of solutions which need to be

constructed before effectively biasing the probability dis- tribution to favour the emergence

6of good solutions. Different applications have favoured either the use of conditioning at the

level of decision variables, thus requiring a huge number of iterations before getting a precise

distribution, or the computational efficiency, thus using very coarse conditioning information.

The chapter is structured as follows. Section 2 describes the common elements of the

heuristics following the ACO paradigm and outlines some of the variants proposed. Section 3

presents the application of ACO algorithms to a number of different combinatorial

optimization problems and it ends with a wider overview of the problem attacked by means

of ACO up to now. Section 4 outlines the most significant theoretical results so far published

about convergence properties of ACO variants.

ACO is a class of algorithms, whose first member, called Ant System, was initially proposed

by Colorni, Dorigo and Maniezzo . The main underlying idea, loosely inspired by the

behaviour of real ants, is that of a parallel search over several constructive computational

threads based on local problem data and on a dynamic memory structure containing

information on the quality of previously obtained result. The collective behaviour emerging

7from the interaction of the different search threads has proved effective in solving

combinatorial optimization (CO) problems.

we use the following notation. A combinatorial optimization problem is a problem defined

over a set **C** = c*1*, ... , c*n* of basic *components*. A subset **S** of components represents a

*solution* of the problem; **F** ⊆ 2***C*** is the subset of *feasible solutions*, thus a solution **S** is

feasible if and only if **S** ∈ **F**. A *cost function z* is defined over the solution domain, z : 2***C*** ?

**R**, the objective being to find a minimum cost feasible solution S\*, i.e., to find S\*: S\* ∈ **F**

and z(S\*) ≤ z(**S**),

∀**S**∈**F**. Given this, the functioning of an ACO algorithm can be summarized as follows (see

also [27]). A set of computational concurrent and asynchronous agents (a colony of ants)

moves through states of the problem corresponding to partial solutions of the problem to

solve. They move by applying a stochastic local decision policy based on two parameters,

called *trails* and *attractiveness*. By moving, each ant incrementally constructs a solution to

the problem. When an ant completes a solution, or during the construction phase, the ant

evaluates the solution and modifies the trail value on the components used in its solution.

This pheromone information will direct the search of the future ants.

Furthermore, an ACO algorithm includes two more mechanisms*: trail evaporation* and,

optionally, *daemon actions*. Trail evaporation decreases all trail values over time, in order to

avoid unlimited accumulation of trails over some component. Daemon actions can be used to

implement centralized actions which cannot be performed by single ants, such as the

invocation of a local optimization procedure, or the update of global information to be used to

decide whether to bias the search process from a non-local perspective .More specifically, an

*ant* is a simple computational agent, which iteratively constructs a solution for the instance to

solve. Partial problem solutions are seen as *states*. At the core of the ACO algorithm lies a

loop, where at each iteration, each ant *moves* (performs a *step*) from a state ι to another one ψ,

corresponding to a more complete partial solution. That is, at each step σ, each ant *k*

computes a set **A***k* σ(ι) of feasible expansions to its current state, and moves to one of these in

probability. The probability distribution is specified as follows. For ant *k*, the probability

pιψ*k* of moving from state ι to state ψ depends on the combination of two values:

• the *attractiveness* ηιψ of the move, as computed by some heuristic indicating the *a priori*

desirability of that move.

• the *trail level* τιψ of the move, indicating how proficient it has been in the past to make that

particular move: it represents therefore an *a posterior* indication of the desirability of that

move.

8Trails are *updated* usually when all ants have completed their solution, increasing or

decreasing the level of trails corresponding to moves that were part of "good" or "bad"

solutions, respectively. The general framework just presented has been specified in different

ways by the authors working on the ACO approach. The remainder of Section 2 will outline

some of these contributions.

The ant system simply iterates a main loop where *m* ants construct in parallel their solutions,

thereafter updating the trail levels. The performance of the algorithm depends on the correct

tuning of several parameters, namely: α, β, relative importance of trail and attractiveness, ρ,

trail persistence, τ*ij*(0), initial trail level, *m*, number of ants, and Q, used for defining to be of

high quality solutions with low cost. The algorithm is the following.

**1. {Initialization}**

**Initialize** τιψ **and** ηιψ**,** ∀**(**ιψ**).**

**2. {Construction}**

**For each ant *k* (currently in state** ι**) do**

**repeat**

**choose in probability the state to move into.**

**append the chosen move to the *k*-th ant's set tabu***k***.**

**until ant *k* has completed its solution.**

**end for**

**3. {Trail update}**

**For each ant move (**ιψ**) do**

**compute** Δτιψ

**update the trail matrix.**

**end for**

**4. {Terminating condition}**

**If not(end test) go to step 2**

9**3 Applications OF ACO:**

**3.1 Travelling sales man:**

The TSP is a very important problem in the context of Ant Colony Optimization because it is

the problem to which the original AS was first applied, and it has later often been used as a

benchmark to test a new idea and algorithmic variants.

**OBJECTIVE:**

Given a set of *n* cities, the Traveling Salesman Problem requires a salesman to find the

shortest route between the given cities and return to the starting city, while keeping in mind

that each city can be visited only once

10The TSP was chosen for many reasons:

•

It is a problem to which the ant colony metaphor

•

It is one of the most studied NP-hard problems in the combinatorial optimization

•

it is very easily to explain. So that the algorithm behavior is not obscured by too

many technicalities.

Since the route B is shorter, the ants on this path will complete the travel more times and

thereby lay more pheromone over it.

The pheromone concentration on trail B will increase at a higher rate than on A, and soon the

ants on route A will choose to follow route B

Since most ants will no longer travel on route A, and since the pheromone is volatile, trail A

will start evaporating Only the shortest route will remain

**WHY TSP IS DIFFICULT TO SOLVE**

Finding best solution may entail an exhaustive search for all combination of cities, this can be

prohibitive as “N” gets large. Heuristic like greedy methods doesn’t guarantee optimal

solutions.

11**Ant colony optimization in TSP:**

The meta heuristic Ant Colony Optimization (ACO) is an optimization algorithm

successfully used to solve many NP hard optimization problems introduced in ACO . ACO

algorithms are a very interesting approach to find minimum cost paths in graphs especially

when the connection costs in the graphs can change over time, i.e. when the problems are

dynamic. The artificial ants have been successfully used to solve the (conventional) Traveling

Salesman Problem (TSP) , as well as other NP hard optimization problems, including

applications in quadratic assignment or vehicle routing.

The algorithm’s based on the fact that ants are always able to find the shortest path between

the nest and the food sources, using information of the pheromones previously laid on the

ground by other ants in the colony. When an ant is searching for the nearest food source and

arrives at several possible trails, it tends to choose the trail with the largest concentration of

pheromones, with a certain probability p. After choosing the trail, it deposits another

pheromone, increasing the concentration of pheromones in this trail. The ants return to the

nest using always the same path, depositing another portion of pheromone in the way back.

Imagine then, that two ants at the same location choose two different trails at the same time.

The pheromone concentration on the shortest way will increase faster than the other: the ant

that chooses this way, will deposit more pheromone in a smaller period of time, because it

returns earlier. If a whole colony of thousands of ants follows this behaviour, soon the

concentration of pheromone on the shortest path will be much higher than the concentration

in other paths. Then the probability of choosing any other way will be very small and only

very few ants among the colony will fail to follow the shortest path. There is another

phenomenon related with the pheromone concentration since it is a chemical substance, it

tends to evaporate, so the concentration of pheromones vanishes along the time. In this way,

the concentration of the less used paths will be much lower than that of the most used ones,

not only because the concentration increases on the other paths, but also because its own

concentration decreases.

12**Flow chart for TSP using ACO**

13**Ant Colonies Optimization Algorithm:**

**procedure** Ant colony algorithm

Set for every pair (i, j): Tij = Tmax

Place the g ants

**For** i = 1 to N:

*Build a complete tour*

**For** j = 1 to m

**For** k = 1 to g

Choose the next node using pk ij in (2)

Update the **tabu list** T

**End**

**End**

*Analyze solutions*

**For** k = 1 to g

Compute performance index fk

Update globally Tij(t + m × g) using (3)

**End**

**End**

Euclidean distance between two locations dij is used as heuristic. However, within a city, the

traveling time tij between two machines is more relevant than distance, due to traffic reasons.

Therefore, the heuristic function is given by \_ = (tij − tmin)/(tmax − tmin), where tij is the

estimated traveling time between location i and location j and tmin = min tij and tmax = max

tij are the minimum and maximum travelling times considered. In this way, the heuristic

matrix \_ entries are always restricted to the interval [0, 1]. The objective function to

minimize, fk(t), is simply the sum of travelling time between all the visited locations:

14*Jik*

**Rules for Transition Probability**

1. Whether or not a city has been visited

Use of a **memory**(tabu list): : set of all cities that are to be visited

2. =

**visibility**: Heuristic desirability of choosing city j when in city i.

3**.**Pheromone trail: This is a global type of information

Transition probability for ant k to go from city i to city j while building its route.

a = 0: closest cities are selected

Trial visibility is η ij = 1/dij

The intensity in the probabilistic transition is α

The visibility of the trial segment is β

The trail persistence or evaporation rate is given as ρ

Trail intensity is given by value of τ ij

15

*ij*

*N*

*dij*

1

*Tij* (*t*)

o t h e r w i s e

0

j a l l o w e d

i f

k















∈

∑

=

∈

*k*

*k a l l o w e d*

*i k*

*i k*

*i j*

*i j*

*k*

*i j*

*[ ]*

*(t ) ]*

*[*

*[ ]*

*[ (t ) ]*

*(t )*

*p*

α β

α β

η

τ

η

τ**TSP Applications**

• Lots of practical applications

• Routing such as in trucking, delivery, UAVs

• Manufacturing routing such as movement of parts along

manufacturing floor or the amount of solder on circuit board

• Network design such as determining the amount of cabling required

• Two main types

– Symmetric

– Asymmetric

**TSP Heuristics**

• Variety of heuristics used to solve the TSP

• The TSP is not only theoretically difficult it is also difficult in practical

application since the tour breaking constraints get quite numerous

• As a result there have been a variety of methods proposed for the TSP

• Nearest Neighbour is a typical greedy approach

**Advantages:**

• Positive Feedback accounts for rapid discovery

of good solutions

• Distributed computation avoids premature

convergence

• The greedy heuristic helps find acceptable

solution in the early solution in the early stages

of the search process.

• The collective interaction of a population of

agents