CHAPTER 4
PROPOSED ALGORITHM

This chapter provides the details of the optimization process, the proposed algorithm, its flow chart and the behaviour of Ant Algorithm. The purpose of optimization is to achieve the best design relative to a set of prioritized criteria or constraints. These include maximizing factors such as efficiency, reliability, productivity etc. Ant colony optimization is one of the most recent techniques for approximate optimization. The inspiring sources of ACO algorithm are real ant colonies. Most specifically, ACO is inspired by the ants foraging behaviour. At the core of this behaviour is the indirect communication between the ants by means of chemical pheromone trails, which enables them to find short path between their nest and food source.

4.1 Ant Process (AP):
It was the first Ant colony optimization algorithm. Three different types of AP had been defined. These were known as ant-density, Quantity of an Ant, or Ant-cycle. Whilst in density and quantity of an ant, the ants update the pheromone immediately after a transfer from a one place to another, in ant-cycle, the pheromone modified was once handiest finished after that all ants had constituted the route and quantity of pheromone submitted by using every ant was set to be the performance of route quality. Considering the fact that ant-cycle completed significantly superior than other two editions, here an ant-cycle algorithm, is presented, referred to it as Ant process with in following discussion. In ant process system each and every of (artificial) ants create alternate (tour) of TSP, as mentioned earlier than.

ACO ideas are situated on ordinary nature of ants. In daily life, probably most duties ants must perform are to seek for meals in neighbourhood of their nest. Whilst moving in such a quest, ants submit a chemical material referred to as Pheromone within floor. That is performed with two basic objects. It allows for ants to seek out their route again to the nest, comparable to Gretel and Hansel in fairy-tale and however, it makes it possible for different ants to know about their path, so that other ants can follow that path.
The curiosity is that, in view that huge amount of ants or countless numbers of ants have this type of nature, if one might see pheromone laid within floor as a variety of sunshine, the bottom could be a tremendous community with one of the arcs brighter than others and inside the paths generated with the aid of those arcs is the shortest path among nest and the food resource. This type of nature may also be noticeable as a style of interaction among the different ants. If route has a bigger awareness of pheromone that is most like a result of its smaller size that has permitted ants to travel leading to a bigger number of travel through the route for this reason as more ants submitting pheromone on it.

This behaviour discovered in ants, i.e. Path-laying and path-following is the inspiring supply of ACO. One of the first and most exciting experiences with real ants, in seeking for a motive for the choice ants make between paths to follow. On this expertise, there have been two bridges connecting the nest of an Argentine ant species and a meals source, one bridge being longer than the opposite. The experiences have been running with a collection of extraordinary ratios between the bridges dimension. The outcome showed that although in the beginning, the ants looked as if it would pick at random the path to follow, after a while they all converged to the shorter path. The investigators determined that this was due to the amount of pheromone present in the route.

If the quantity of moving in a particular direction is high, then pheromone quantity as a way to be deposited in that route can also be excessive, which incentives other ants to follow it. One interesting observation used to be made when probably the most paths were twice as long as the other. Even though nearly all of the ants adopted the shorter course, there was perpetually a small percentage that used the longer direction. This truth was interpreted as a variety of exploration behaviour.

Some researches ready to exhibit that foraging ants may to find shorter route among their meal resource and nest, by means of a chemical substance referred to as pheromone that they deposit whilst moving. After these types of researches, authors planned a stochastic mannequin to explain what that they had located. This was the innovative step towards an optimization algorithm based on ant’s behaviour. This
algorithm was called Ant procedure and used to be to begin with proposed to remedy the travelling salesman trouble.

### 4.2 Ant Colony Optimization:

In the proposed model, ACO algorithm is used to decrease the time duration to migrate the data from source to other destination. Ant Colony optimization algorithm describes that if data will travel through shortest path, then less time will be consumed and ultimately cost will also decreased. In this problem, each ant is behaviourally as much unsophisticated insect. Basically these ants have the small memory and exhibit an individual behaviour that appears to have large random component. As a crowd however, ants can handle to perform a variety of difficult problems with great consistency and reliability.

Despite the fact, this is almost essentially self-organizing as the learning ants have to take care of a phenomenon that looks like perfect training in reinforcement learning approaches. The problematic social behaviours of ants were so much studied by means of science, now a days scientist at moment are in process of research of discovering that these types of behaviour patterns can provide model for solving different types of optimization related issues. An Attempt to generate algorithms encouraged by one aspect of ant nature, capability to search out what scientists would call smallest paths, has become the field of ant colony optimization, the most positive and generally recognized algorithmic technique based on ant behaviour.
Ants are very little insects. These ants have an ability to find the shortest path. Ants leave some chemical material which is referred to as pheromone while moving. Basically movement of an ant and deposition of pheromone have the equal speed at equal rate. Chemical substances attract different ants to follow same route. Pheromone fades away after some instances of time. When huge number of ants travels on the same route, Intensity of pheromone enhances rapidly. The two factors which affect the intensity of pheromone are:

- Quality of meal.
- Distance of meals.

In process of Cloud Computing the “meals” is taken as different services of web. The route that has high amount of pheromone is mostly chosen by other ants, the reason behind this is that the shorter path will receive a greater amount of pheromone than longer path because the ants choosing the shorter path will reach to food early and return to the nest by same path due to un-evaporated of pheromone. So, smallest route has extra quantity of pheromone and ultimately all ants choose the smallest path. According to ACO state Transition Rule, the probability with which ant $k$ currently at stage $i$ choosing to go to stage $j$ is given by
Migration of Data From One Cloud Server to another Cloud Server using the TCP-IP Protocol

\[ P_{ij}^k(t) = \frac{[\tau_{ij}(t)]^r[\eta_{ij}(t)]^q[d_{\text{load}}]^{d_{\text{load}}}}{\sum_{l=1}^{n_{ij}}[\tau_{lj}(t)]^r[\eta_{lj}(t)]^q} \]  

.............................. Eq. (4.1)

Where,

- \( P_{ij} \) = Pheromone trail
- \( ij \) = Heuristic value
- \( r \) = Parameter that find relative influence of pheromone trail.
- \( d \) = Decision state according to load.

**Recommend Algorithm is elaborated as mentioned**

Step 1: arbitrarily choose a Job Scheduler.

Step 2: Job Scheduler schedule a job to other services of web.

- While Job is not in schedule,
- then
- Do again step 3 and 4.

Step 3: Job check its nearby region for availability of web services with probability,

\[ P_{ij}^k(t) = \frac{[\tau_{ij}(t)]^r[\eta_{ij}(t)]^q[d_{\text{load}}]^{d_{\text{load}}}}{\sum_{l=1}^{n_{ij}}[\tau_{lj}(t)]^r[\eta_{lj}(t)]^q} \]  

.............................. Eq. (4.2)

Step 4: if

- Server is accessible
  - After that
  - Get server
- Else

  Switch to step 3

Step 5: back to Job Scheduler.

Step 6: terminate job after completion.

Step 7: End

The Ant Colony Optimization algorithm is a metaheuristic that has a mixture of autocatalysis, distributed computation, and optimistic materialism to search out an ideal resolution for combinatory optimization issues. This type of Algorithm aims to mimic ant’s nature in real world. Nowadays, an ACO algorithm is in consideration...
and has implemented in many development and optimization issues, such as network paths, quadratic project, travelling salesman and resource distributed issues.

The Ant Colony Optimization algorithm has been propounded by researches run by an author “Goss et. al.” making use of a real ant’s colony. They discovered that real ants were competent to decide shorter path among nests and meal source, by existence of alternate routes among two. This search is made viable with the aid of an indirect communication often called stigmatize among ants. While moving, ants submit a chemical material, referred to as pheromone, on bottom. Once they reach at a point where they can take decision, they can generate probabilistic alternative, partial by means of intensity of pheromone. This nature has an auto catalytic effect due to fact that the path that an ant deciding would develop probability that alternative route would probably be selecting once more by means of different ants in further. When ants return again, probability of choosing the identical route is larger. Fresh pheromone can be launched on selected route that makes it much attractive for further ants. In short, all ants would choose the shorter route.

Figure 4.2 (b) shows that lots of ants select upon the shortest path/route in principle have the capability of making a solution (i.e., of finding a path among nest and food resource), it is just colony of ants that represents the “shortest path finding” nature. In a sense, this nature is an emergent property of an ant colony.

![Figure 4.2: Double bridge test.](image)
Figure 4.2 (a)Shows the process how Ants begin exploring double bridge.

In Figure 4.2, different nature of ants in double bridge test is shown. In the specific case, by considering the equal pheromone laying mechanism the shortest path is much typically selected. Primary ants to reach at meals resource are those which have taken two short streams. On that time ants begin their return route; extra pheromone is released on short branch than Long Branch. This will likely stimulation successive ants to decide upon the shorter branch. This nature was once formed as system of Ant by author “Dorigo et al.” based on AS algorithm, ACO algorithm was innovated [1].

In Ant Colony Optimization algorithm, optimization issue is formed as a graphic representation $G = (C; L)$; here $C$ is group of component of issue, and $L$ is group of feasible transition between factors of $C$. This solution can be expressed as possible routes on graphic representation $G$, with respect to a set of certain parameters. The ants mutually resolve issue by using graph representation. Although every ant is able of discovering a (more commonly terrible) solution, good first-rate solutions can emerge due to combined interaction between ants. Pheromone trails encode a long memory about whole ant search technique. Its price based upon situation described and optimization function.

Even as (termination criterion no longer fulfilled)
Generation of ant and exercise ();
Pheromone
Evaporation ();
Daemon procedures (); “non-compulsory”
Finish at the same time
End Algorithm

The nature of ants in ACO algorithm can also be summarized as mentioned. An ant colony simultaneously moves via adjoining states of concern through relocating of local nodes of $G$. They transfer with the aid of submitting a stochastic local selection coverage that makes utilization of information stores in nearby point and routing.
table of ant’s. When solution is being prepared, each ant describes the solution and get knowledge about its kindness on pheromone trails of connection used. This pheromone information will direct the search of future ants, unless a viable resolution is obtained [8].

Ant Colony Optimization as late planned Meta heuristic technique for fixing tough 1 optimization issues. Inspirational source of ACO is pheromone following nature of precise ants that utilize pheromones as a medium of interaction. As a biological e.g., ACO is centred on oblique interaction of a colony of simple dealers, called (artificial) ants, intervene with support of pheromone trails. Pheromone trails in ACO functionally distributed, arithmetical capabilities that ants utilize to probabilistically gather options to crisis being resolved and that ants adjust throughout the algorithm’s execution to their research experience.

Artificial ants in ACO execution a randomized development heuristic which made it rare selections as a function of man-made pheromone trails and in most cases available heuristic information headquartered on input knowledge of drawback to be resolved. Like this, ACO may also be interpreted as an extension of ordinary building heuristics which can be comfortably to be had for a lot of combinatorial optimization problems. However, an essential differentiation with progress heuristics is difference of pheromone trails in algorithm execution to take into account cumulated search advantage.

The ACO that is explained in Algorithm is manufactured from common recommendations for the development of ant algorithms to clear up optimization problems.

There are four Algorithms of Pseudo-code for ACO.

1. Initialization parameters
2. Initialization of pheromone trails
3. Generation of ants
4. Whilst ending criterion isn’t achieved do
5. Let ants build their solution
6. Update pheromone trails
7. Permit Daemon moves
8. End While

Predominant alternate from common constitution of AS algorithm is introduction of a Daemon. Operations which may also be carried out with the support of daemon provide world expertise of solutions, hence having an awfully predominant and energetic performance in algorithm, in contrast to AS algorithm a place every ant was speculated to give pheromone in its resolution regardless of what the reverse choices were like. It is a mission that does not equivalence inside the nature. Daemon may, for illustration, manipulate feasibility of every decision or provide an additional pheromone extensive variety to the first-class answer found from starting of the run of the algorithm, or even to the excellent solution in existing iteration. These type of operations have been recounted in earlier algorithm but never attributing its responsibility to a main entity in colons.

One more primary characteristic, ordinarily utilized by ant colony optimization algorithm is utilization of local search process which follows the generation of solution. It is most effective feature that has found to be important in research discipline close to higher solutions, leading by and large to raise performances of ACO.

In this section, a short historical perspective of the innovative idea is presented that result in the development of ant algorithms. The number of investigators working with and the number of ant algorithms were growing exponentially, especially following the definition of the ACO Meta heuristic. It would accordingly, be impossible to give a correct account of all of them.

**Ants in ACO algorithm have the following features:**

With a view to get extra perception on algorithm, the same ant colony optimization will also be carried out in cloud system retrieval, in different steps. In this process a cloud agent will also behave as an ant, that is supposed for data recovery and search,
results will also be saved within the storage area and after successful completion of process, stored data may be viewed as output from cloud.

1. Each cloud agent searches for the data which is taken for the retrieval
2. Cloud agent has its own memory which enable to store the searched data in it for latter, based on the memory capacity different types of data can be searched out and stored for the future retrieval purpose.
3. A cloud agent C may be allocated a start state SC or retrieving storage buffer can be set as EC
4. Cloud agent start from an initial state and move over all to the feasible data locations, making solutions in an incremental path. Procedure terminate whilst at least 1 queried data has found.
5. The Cloud agent locates a data in a node ‘f’ may transfer to node ‘g’ selected in a possible zone ‘Nc’ throughout probabilistic assessment rules. This may be formed as mentioned: A cloud agent ‘c’ into state $sr =< Sr - 1; f >$ may transfer to some node in its possible zone ‘Nc’.
6. A probabilistic regulation is task of following
   (a). Data contained in a local node, data structures $A_f = [A, fg]$ known as ant route table obtains from. Pheromone trails and heuristic values
   (b). Ant’s self-memory from preceding iteration, and issue constraints.
7. When transforming from node ‘f’ to local node ‘g’, agent modified pheromone trails $t(fg)$ on edge (f, g).
8. As soon as the information is retrieved from the cloud, the agent may redesign same route backward, update pheromone trails and end the operation. A cloud agent retrieves all forms of data from cloud storage.

4.3 The Building Blocks of Ant Algorithm:
The following features must be specified, when describing an ant algorithm:
- Pheromone update rule,
- probability function and Transition rule,
- Parameters constraints, and
- Approach chosen to build the solution,
- Termination situation.
• Heuristic information

It becomes noticeable that the unique procedures that may be developed, when combined with each other, result in a variety of ant colony algorithm, which are sufficient to deal with different types of issues. As per literature in the field, distinctive plans will also be specified to strengthen previous results or without problems to clear up another kind of issues in present scenario. In the earlier sections the process of detail, the AS is discussed that was once used to resolve the TSP. Assessed enormously, number of approaches proposed earlier can be on the grounds that it is not possible to monitor whole work which has been accomplished since the previous level of an algorithm. Nevertheless, reader is normally referred to worse as a way to be mentioned in this part, for additional information, or references there in.

{Start}
Start $\tau_\psi$ and $\eta_\psi, \forall(\psi)$
1. {Create}
   Every ant (k) (existing within state i) do
   repeat
   Select in possibility state to maneuver.
   Attach selected shifted to k-th ant's group tabuk unless ant “k” have finished its solution.
   End for
2. {Trails updating}
   each ant move(\psi) do
   calculate $\Delta \tau \psi$
   modified trail matrix.
   End for
3. {Terminated Condition}
   If not (terminate test) switch to step 2

4.4 Ant colony algorithm Flow chart:
In these modules, the ants implies as jobs and food implies as server. The ants first select the server, if the server is available the ants perform in that server and
evaluate the best result as shown in figure 4.3. If the server is not available to do the jobs the ants choose next server by using the transition rule. The server already have different amount of load. The load increases or decreases according to the jobs. By using number of ants the response time will slightly increases.

![Ant Colony Algorithm Flow Chart](image)

**Figure 4.3: Ant Colony Algorithm Flow Chart**

4.5 **Ant Behaviour:**

1. Ant behaviour is stochastic
2. The behaviour is induced by indirect communication (Pheromone paths) - Stigmergy
3. Ants explore the search space
4. Limited ability to sense local environment
5. Act concurrently and independently
6. High quality solutions emerge via global cooperation
Migration of Data From One Cloud Server to another Cloud Server using the TCP-IP Protocol

Figure 4.4: Ant Colony Optimization

Formal Description:
When new source ‘m’ is added to the cluster of cloud, the number ‘k’ of processors could offer the capability of every processor ‘p’ and bandwidth of resource ‘b’, kind of every processor, size of memory etc. The native pheromone’s depth ‘q_m’ and source ‘m’ starting in line.

\[ q_m = \sum_{i=1}^{k} p_i + b \]  \hspace{1cm} \text{Eq. (4.3)}

At the start of algorithm, pheromone intensity \( w_m(t) \) is initialized in step with the native pheromone intensity of every resource:

\[ w_m(0) = q_m \]  \hspace{1cm} \text{Eq. (4.4)}

Each ant selects their duty and resources to perform this. The ant selects a useful source for a task with probability in step according to (Eq.4.5).

\[ p_m^i(t) = \left\{ \frac{[w_m(t)]^c [q_m]^d}{\sum_d [w_m(t)]^c [q_m]^d} \right\} \]  \hspace{1cm} \text{Eq. (4.5)}

Where (1)\( w_m(t) \) = actual-time pheromone intensity of useful source ‘m’ is time \( t \); (Eq.4.4) \( q_m \) = native efficiency variety of useful source ‘m’; ‘c’ and ‘d’ = dynamic
parameters and may be modified based on conditions. \( C \) = Presents actual-time pheromone intensity, \( d \) = represents the importance measure of the innate pheromone intensity.

When a useful source is chosen, pheromone intensity of it would be up-to-date according to (Eq.4.6).

\[
\begin{align*}
\omega_m(t) &= \omega_m(t) - \theta l \quad \text{Eq. (4.6)}
\end{align*}
\]

\( l \) =the length of the assigned task onto resource mat time \( t \), \( \theta = (0<\theta<1) \) variable parameter.

Finally duties are be assigned, compute the whole execution time and determine whether or not it's the shortest execution time in present. If so, the change of the real-time pheromone according to equation 4 can be cancelled, after that updates the time pheromone intensity in step with (Eq.4.7).

\[
\begin{align*}
\omega_m(t) &= (1 - \rho)\omega_m(t) + \rho\Delta\omega_m \quad \text{Eq. (4.7)}
\end{align*}
\]

\( \Delta\omega_m \) is entire length of all tasks assigning to resource \( m \) this instance; and \( \rho(0 < \rho < 1) = \) different dynamic parameter. ‘1-\( \rho \)’ = signify pheromone residue factor.

**4.6 Proposed Algorithm:**

**Step 1:** Initialize the Connection String of Primary Server in the Middle Server.

**Step 2:** Check if (IsPagePostback == False)

Clear DropDownList.

Else

Create Index for Table in Database. (Index Shows the Number of Table)

**Step 3:** If (Table.Index == 0)

Message: No Schema is in Primary Sever or Schema is Already Migrated.

Exit.

**Step 4:** If (WholeDatabase ==1)

Repeat While Index != Null

Generate Architecture with Data Till Index Not Equal to Null.

Index - -

Else
Goto Step 7:

Step 5: Initialize Connection String of Secondary Server

Step 6: Run Migration Protocol

Step 7: If(Selected Index = 1)
   Generate Architecture and Data for Selected Index.
   Run Migration Protocol.
   Else
   Message: Please Select Table from the Selected Index for Migration.

Step 8: Stop.