CHAPTER 1 INTRODUCTION

1.1 INTRODUCTION

Cloud Computing provides us a means by which we can access the applications as utilities, over the Internet. It allows us to create, configure, and customize applications online. Cloud Computing refers to manipulating, configuring, and accessing the applications online. It offers online data storage, infrastructure and application. Cloud computing has been a familiar topic all over the world for quite a lot of time now. It basically acts as a medium using which the users are able to access different pieces of information using a network or a web browser. Therefore, it eliminates the requirement for evolving and maintaining expensive facilities required for computing. The features of a general cloud system are: access of on-demand services, elasticity, drastic reduction of cost, minimum effort required for management, scalability, and independence of the device or the location. As gradually there is an increase in the deployment and adoption of computing in the cloud, it is very essential to evaluate the performance of different cloud environments. Simulation and modeling technologies are generally used for measuring various issues in security and performance.

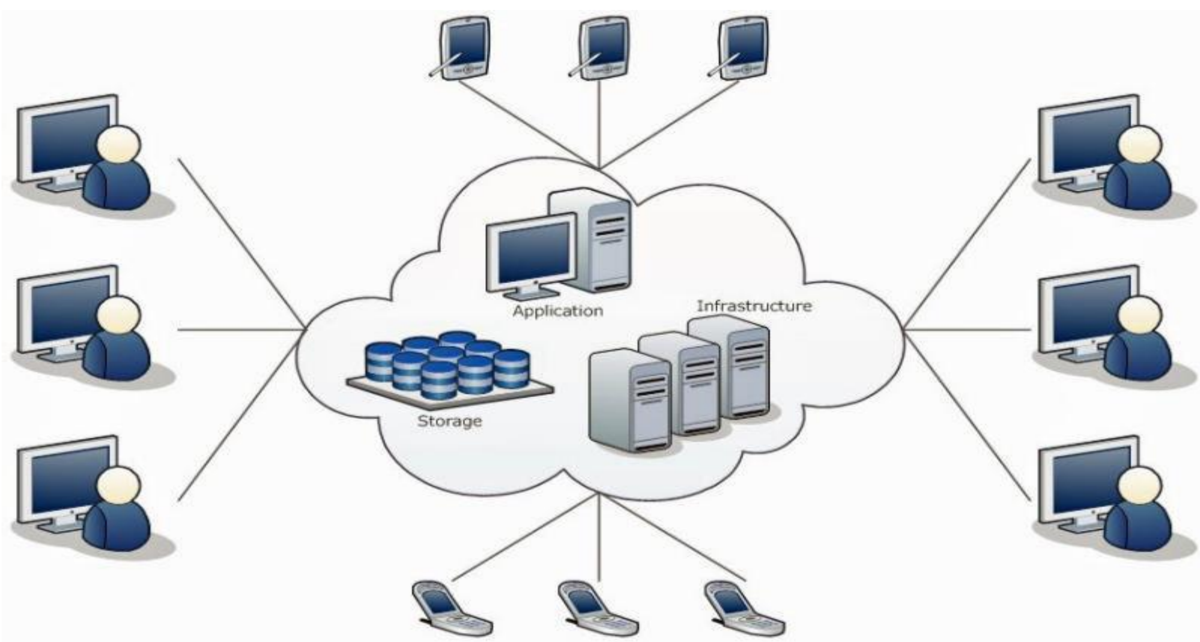


Figure 1. Cloud Computing

DEPLOYMENT MODELS

Deployment models define the type of access to the cloud, i.e., how the cloud is located? Cloud can have any of the four types of access: Public, Private, Hybrid and Community

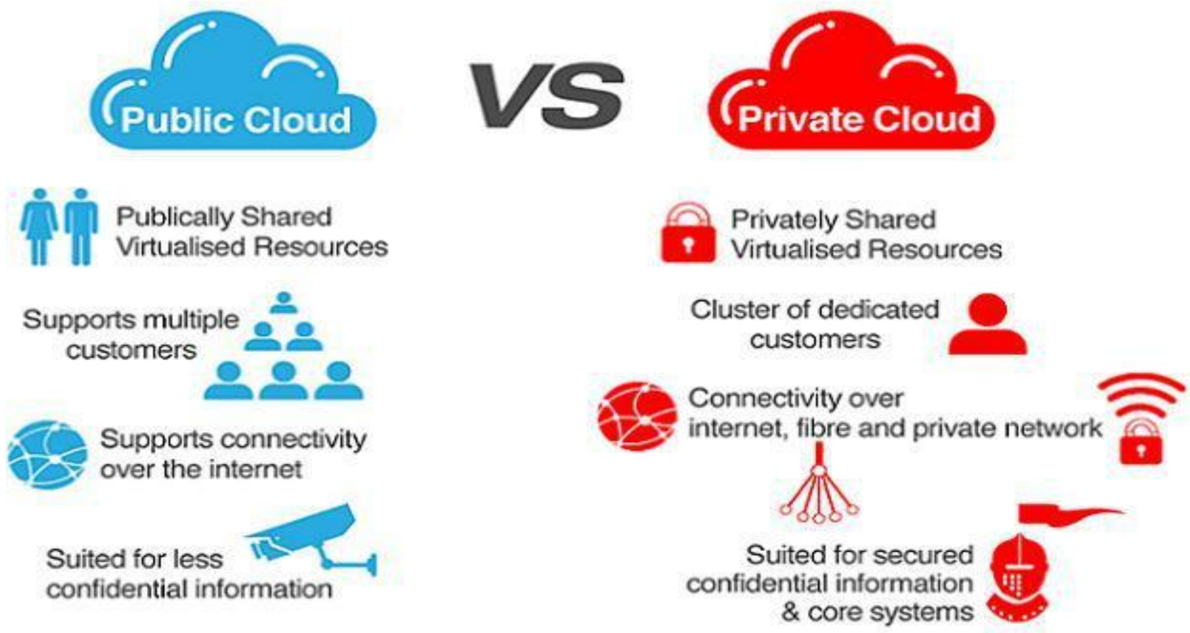


Figure 2. Public cloud Vs Private cloud

(i) PUBLIC CLOUD

Public cloud or external cloud describes cloud computing in the traditional mainstream sense, whereby resources are dynamically provisioned on a fine-grained, selfservice basis over the Internet, via web applications/web services from an off-site third-party provider who bills on a fine-grained utility computing basis. The cloud infrastructure is made available to the general public or a large industry group, and is owned by an organization selling cloud services. Examples: Amazon Elastic-Compute-Cloud, IBM's BlueCloud, Sun Cloud, Google AppEngine.

(ii) P R I V A T E C L O U D

A private cloud is a particular model of cloud computing that involves a distinct and secure cloud based environment in which only the specified client can operate. As with other cloud models, private clouds will provide computing power as a service within a virtualized environment using an underlying pool of physical computing resource. However, under the private cloud model, the cloud (the pool of resource) is only accessible by a single organization providing that organization with greater control and privacy.

(iii) C O M M UN I T Y C L O U D

Community cloud: A community cloud may be established where several organizations have similar requirements and seek to share infrastructure so as to realize some of the benefits of cloud computing. With the costs spread over fewer users than a public cloud (but more than a single tenant) this option is more expensive but may offer a higher level of privacy, security and/or policy compliance. Examples of community cloud include Google's "Gov Cloud".

(iv) H Y B R I D C L O U D

The term "Hybrid Cloud" has been used to mean either two separate clouds joined together (public, private, internal or external), or a combination of virtualized cloud server instances used together with real physical hardware. The most correct definition of the term "Hybrid Cloud" is probably the use of physical hardware and virtualized cloud server instances together to provide a single common service. Two clouds that have been joined together are more correctly called a "combined cloud". A hybrid storage cloud uses a combination of public and private storage clouds. Hybrid storage clouds are often useful for archiving and backup functions, allowing local data to be replicated to a public cloud.

SERVICE MODELS

Service Models are the reference models on which the Cloud Computing is based. These can be categorized into three basic service models as listed below:

1. Infrastructure as a Service (IaaS)
2. Platform as a Service (PaaS)
3. Software as a Service (SaaS)

There are many other service models all of which can take the form like XaaS, i.e., anything as a Service. This can be Network as a Service, Business as a Service, Identity as a Service, Database as a Service or Strategy as a Service. The Infrastructure as a Service (IaaS) is the most basic level of service. Each of the service models make use of the underlying service model, i.e., each inherits the security and management mechanism from the underlying model.

Figure 3. Service Model

(i) Software as a Service (SaaS)

The traditional model of software distribution, in which software is purchased for and installed on personal computers, is sometimes referred to as Software-as-a-Product. Software-as-a-Service is a software distribution model in which applications are hosted by a vendor or service provider and made available to customers over a network, typically the Internet. SaaS is becoming an increasingly prevalent delivery model as underlying technologies that support web services and service-oriented architecture (SOA) mature and new developmental approaches become popular. SaaS is also often associated with a pay-as-you-go subscription licensing model. Mean-while, broadband service has become increasingly available to support user access from more areas around the world. Examples are Google's Gmail and Apps, instant messaging from AOL, Yahoo and Google.

(ii) Platform as a Service (PaaS)

Cloud computing has evolved to include platforms for building and running custom web-based applications, a concept known as Platform-as-a-Service. PaaS is an outgrowth of the SaaS application delivery model. The PaaS model makes all of the facilities required to support the complete life cycle of building and delivering web applications and services entirely available from the Internet, all with no software downloads or installation for developers, IT managers, or end users. Examples include Microsoft's Azure and Salesforce's Force.com.

(iii) Infrastructure as a Service (IaaS)

The capability provided to the consumer is the provision of grids or clusters or virtualized servers, processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, which can include operating systems. The highest profile example is Amazon's Elastic Compute Cloud (EC2) and Simple Storage Service, but IBM and other traditional IT vendors are also offering services, as is telecom-and-more provider Verizon Business.

Figure 4. Cloud Computing

1.2 PROBLEM STATEMENT

Cloud computing has been a new trend in problem solving and providing reliable computing platform for big and high computational tasks. This technique is used for business industries like banking, trading and many e-commerce businesses to accommodate high request rate, high availability for all time without stopping system and system failure. In case of failure, the requests are migrated to different reliable servers with letting the user knowing about it, providing fault tolerant behavior of system. Other application zone includes scientific research like computing weather forecasting report, satellite imaging and many more applications which require high computation, which is now possible without creating a private infrastructure. Cloud deals with various kinds of applications which can have different request type and computation servers based on their capability i.e. hardware and software configuration. However, to computer these large requests count data centers consume high power and load over the data centers is also very high which may be storage, computational or network load. This may lead to reduce in QoS (Quality of Service) provided by a data center that may be due to deadline failure or a fault over a datacenter due to various reasons. Survey in 2006 shows power consumption of data center around 4.5 billion kWh, equivalent to 1.5% of total power consumed by USA, which will be increasing 18% yearly .

Cloud computing in general has various issues listed below:

1) With adoption of cloud computing techniques by industries and corporate users the user count is increasing rapidly with increase in number of cloud computing services. This increases the datacenter count and power consumption.

2) Resource allocation among data centers i.e. allocation of virtual machine on datacenters

to provide high quality resource keeping in mind the behavior and characteristics of datacenters to provide high QoS at resource level.

3) Current task allocation algorithms focus on static load balancing algorithm balance the load when request load increases but does not take into consideration previous behavior of the datacenter under high load which leads us to design efficient learning based algorithms.

4) High loaded data centers has high failure probability to compute requests and with increase in load over data center request completion time increases which is not good for user as well cloud provider. This leads to SLA (Service Level Agreement) failure promised to the client.

5) Some request are need to be computed with QoS but due to high load and fault rate they may the QoS promised which is not appropriate to user and will be a critical issue.

Cloud computing is a reliable computing platform for large computational intensive or data intensive tasks. This has been accepted by many industrial giants of software industry for their software solutions, companies like Microsoft, Accenture, Ericson etc has adopted cloud computing as their first choice for cheap and reliable computing. But which increase in number of clients adopting this there is requirement of much more cost efficient and high performance computing for more trust and reliability among the client and the service provide to guarantee cheap and more efficient solutions. So the tasks in cloud need to be allocated in an efficient manner to provide high resource utilization and least execution time for high performance, at the same time provide least computational cost. Many resource algorithms are been proposed to improve the performance, but are not cost efficient at same time. Algorithms like genetic, particle swarm and ant colony algorithm are efficient solutions but not cost efficient. So to overcome these issues, we require a learning based cost efficient algorithm for cloud Infrastructure.

The problem lies in the scheduling and managing resources in cloud IaaS posed by migration of computation to cloud platform. So performance depends on scheduling method and the space of tasks is complex. Therefore the need of task scheduling algorithm like ACO comes into picture as I provides the most efficient solution in terms of least execution cost as compared to other task scheduling algorithms like PSO,ROUND ROBIN etc.Task scheduling and resource allocation are major problems involved in cloud as well as grid computing. From the study of related work, we concluded that the existing scheduling strategies in clouds are based on the approaches developed in related areas such as distributed systems and Grids.

1.3 OBJECTIVE

Proposed algorithm aims to find least cost and least execution time to finish the task with least finish time (Time taken to complete a set of requests) and at the same time providing the global i.e. the least finish time a system can achieve. To reduce time and increase efficiency of computational performance in cloud computing. Overall to build cloud computing simulators and develop learning based cost efficient scheduling algorithm for cloud infrastructure.

1.4 METHODOLOGY

The field of ‘ant colony algorithms’ studies models derived from the behavior of real ants and it is widely used for combinatorial optimization problems. Ant colony algorithms are developed as heuristic methods to identify efficient paths through a graph and have been applied to identify optimal solutions for service composition problems. The features of ant colony algorithms include positive feedback and local heuristics. An ant colony optimization (ACO) algorithm iteratively performs a loop containing two basic procedures. The first is how the ants construct solutions to the problem, and the second is how to update the pheromone trails [16]. Using ACO algorithms to solve combinatorial optimization problems, requires a representation of the problem and the definition of the meaning of pheromone trails, as well as the heuristic information. Because the results of many ACO applications to NP-hard combinatorial optimization problems show that the best performance is achieved when coupling ACO with local optimizers, it is necessary to implement an efficient local search algorithm.

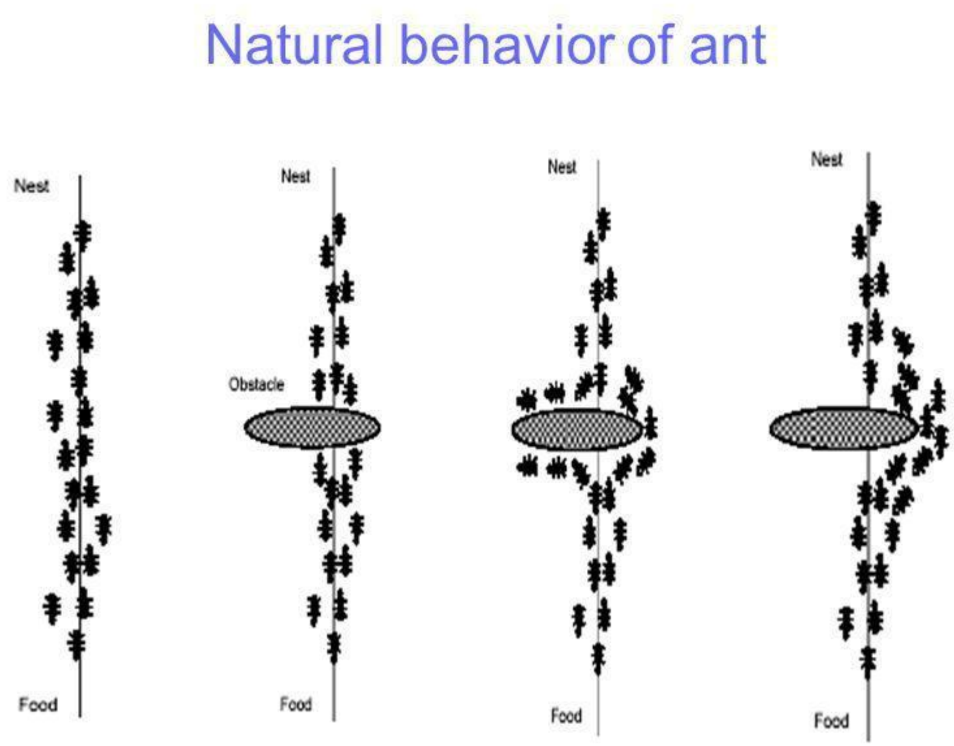


Figure 5. Natural behavior of ANT

There are a variety of options when it comes to using a cloud simulator each with its specifications, qualities, advantages and disadvantages.

Following Table 1 presents the evaluation and analysis of the various cloud simulators that are based on different software or hardware, underlying platforms or developing languages

Simulator	Underlying Platform	Programming Language	Software/Hardware
CloudSim	GridSim	Java	Software
CloudAnalyst	CloudSim	Java	Software
GreenCloud	Ns2	C++, OTcl	Software
NetworkCloudSim	CloudSim	Java	Software
EMUSIM	AEF, CloudSim	Java	Software
SPECI	SimKit	Java	Software
GroudSim	-	Java	Software
DCSim	-	Java	Software

Table 1. Cloud Simulator

In this project, I will be using CloudSim as our cloud simulator mainly due to the following advantages:

- Using it, the cloud users can evaluate particular system problems, without even considering the details which are usually low level related to the cloud-based infrastructures and services.
- It allows seamless experimentation, modeling and simulation of the application services of the cloud computing environment
- It is also a better alternative in the market than other simulators available mainly due to the fact that the existing simulators in the distributed systems aren't compatible with the cloud computing environment.

1.5 ORGANISATION

Figure shows the layered implementation of the CloudSim software framework and architectural components The top-most layer in the simulation stack is the User Code that exposes configuration related functionalities for hosts (number of machines, their specification and so on), applications (number of tasks and their requirements), VMs, number of users and their application types, and broker scheduling policies. A Cloud application developer can generate a mix of user request distributions, application configurations, and Cloud availability scenarios at this layer and perform robust tests based on the custom Cloud configurations already supported within the CloudSim.

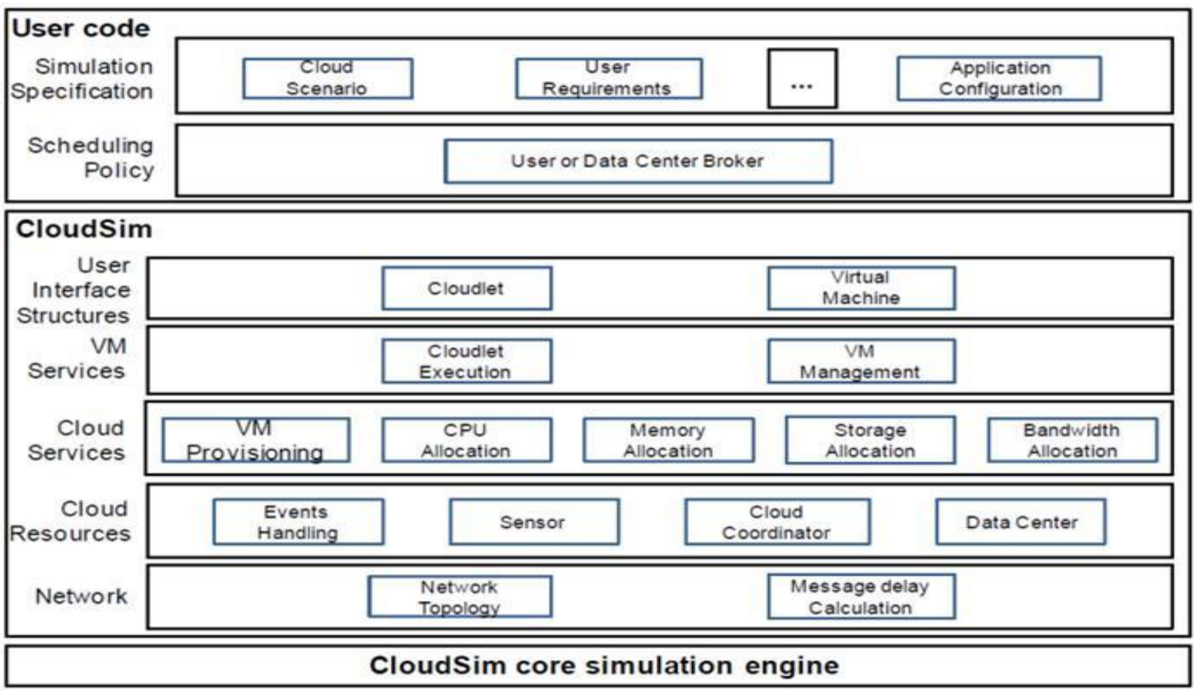


Figure 6. Layered Cloud Computing Architecture

As Cloud computing is a rapidly evolving research area, there is a severe lack of defined standards, tools and methods that can efficiently tackle the infrastructure and application level complexities. Hence in the near future there would be a number of research efforts both in academia and industry towards defining core algorithms, policies, application benchmarking based on execution contexts. By extending the basic functionalities already exposed by CloudSim, researchers would be able to perform tests based on specific scenarios and that may compromise the validity of the simulation have been already detected and fixed. By reusing these long validated frameworks, we were able to focus on critical aspects of the system that are relevant to Cloud computing, while taking advantage of the reliability of components that are not directly related to Clouds.

Data centre characteristics: The data centre characteristics model pieces of information related to the configurations of the data centre resource.

Data centres: The data centres model the services of infrastructure provisioned by different providers of cloud services. The data centres also encapsulate a set of servers or computing hosts that are usually either homogeneous or heterogeneous in their nature and also depend on hardware configurations of their own.

Hosts: Hosts are used to model physical resources which include storage or compute.

Cloudlet : The cloudlet specifies the user instructions. It consists of the i/p as well as o/p files, the ID of application, the respective size of the execution commands for request and the respective name of the user foundation which is also the originator where the replies are to be sent back. It schedules the application services based on the cloud. The CloudSim simulator recognises the application complexity in the terms of its respective computational requirements. Each and every application service must have a data transfer overhead that is required to be carried out during its life cycle as well as an instruction length which is pre-assigned.

VMM allocation policy: The VMM allocation policy provides policies on how to provisionally distribute VM’s to various hosts.

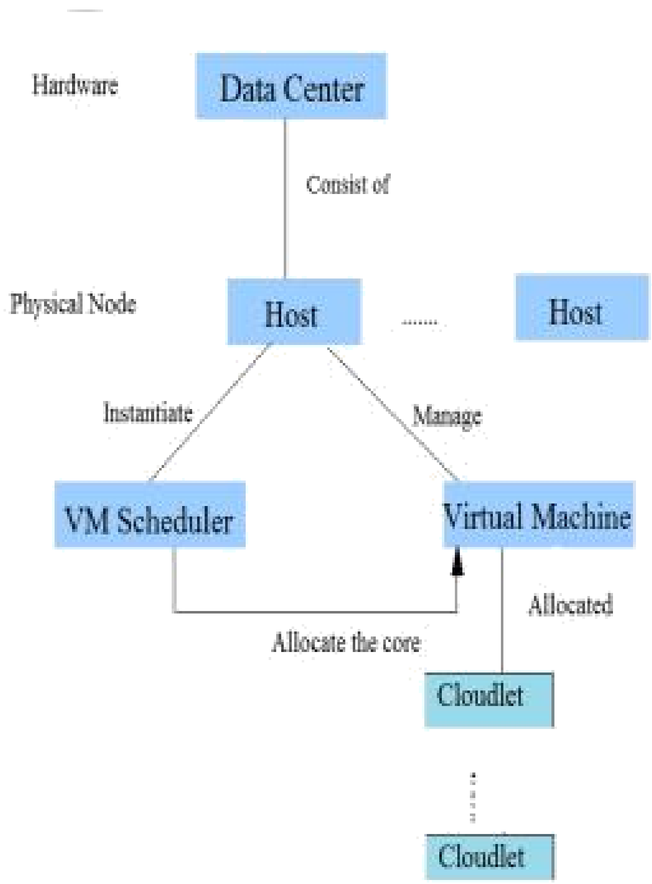


Figure 7. Data center Flowchart

CHAPTER 2 LITERATURE SURVEY

2.1

Marco Dorigo, Mauro Birattari, Thomas Stutzle

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 more researchers and a relatively large amount of 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.

2.2 BIOLOGICAL INSPIRATION

In the forties and fifties of the twentieth century, the French entomologist Pierre-Paul Grass´e observed that some species of termites react to what he called "significant stimuli". 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. Grass´e used the term stigmergy to describe this particular type of communication in which the "workers are stimulated by the performance they have achieved".

The two main characteristics of stigmergy that differentiate it from other forms of communication are the following.

• Stigmergy is an indirect, non-symbolic form of communication mediated by the environment: insects exchange information by modifying their environment.
• Stigmergic information is local: it can only be accessed by those insects that visit the locus in which it was released (or its immediate neighborhood).

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 perceive the presence of pheromone and tend to follow paths where pheromone concentration is higher. Through this mechanism, ants are able to transport food to their nest in a remarkably effective way.

2.3 What is ANT COLONY OPTIMISATION?

Ant colony optimization is derived from the biological behavior of ants. Ant drops pheromones on ground to mark the favorable path that will be followed by other ant members of colony. ACO uses this mechanism to solve optimization problems.
Ant colony optimization (ACO) has been formalized into a metaheuristic for combinatorial optimization. Metaheuristic is a set of algorithmic concepts that can be used to define heuristic methods applicable to a wide set of different problems. In other words, a metaheuristic is a general-purpose algorithmic framework that can be applied to different optimization problems with relatively few modifications.

Where ANT COLONY OPTIMISATION is used?

- Routing: Traveling salesman, Vehicle Routing, Sequential ordering.
- Assignment: Quadratic Assignment, Course timetabling, Graph colouring.
- Scheduling Project Scheduling, Open Shop, and Total Weighted tardiness.
- Other: Bayesian Networks, Classification rules.

2.4 Why ANT COLONY OPTIMISATION is used?
Daniel Merkle and Martin Middendorf

Applications to NP-hard problems

The usual approach to show the usefulness of a new metaheuristic technique is to apply it to a number of different problems and to compare its performance with that of already available techniques. In the case of ACO, this research initially consisted of testing the algorithms on TSP. Subsequently, other NP-hard problems were also considered. So far, ACO has been tested on probably more than one hundred different NP-hard problems. Many of the tackled problems can be considered as falling into one of the following categories: routing problems as they arise, for example, in the distribution of goods; assignment problems, where a set of items (objects, activities, etc.) has to be assigned to a given number of resources (locations, agents, etc.) subject to some constraints; scheduling problems, which–in the widest sense–are concerned with the allocation of scarce resources to tasks over time; and subset problems, where a solution to a problem is considered to be a selection of a subset of available items. In addition, ACO has been successfully applied to other problems emerging in fields such as machine learning and bioinformatics.

Applications to telecommunication networks

ACO algorithms have shown to be a very effective approach for routing problems in telecommunication networks where the properties of the system, such as the cost of using links or the availability of nodes, varies over time.

Ant-based algorithms have given rise to several other routing algorithms, enhancing performance in a variety of wired network scenarios.

Applications to industrial problems

The success on academic problems has raised the attention of a number of companies that have started to use ACO algorithms for real-world applications. Among the first to exploit algorithms based on the ACO metaheuristic is Euro- Bios (www.eurobios.com). They have applied ACO to a number of different scheduling problems such as a continuous two-stage flow shop problem with finite reservoirs. The problems modeled included various real-world constraints such as setup times, capacity restrictions, resource compatibilities and maintenance calendars. Another company that has played, and still plays, a very important role in promoting the real-world application of ACO is AntOptima (www.antoptima.com). AntOptima's researchers have developed a set of tools for the solution of vehicle routing problems whose optimization algorithms are based on ACO. Particularly successful products based on these tools are (i) DYVOIL, for the management and optimization of heating oil distribution with a nonhomogeneous fleet of trucks, used for the first time by Pina Petroli in Switzerland, and (ii) AntRoute, for the routing of hundreds of vehicles of companies such as Migros, the main Swiss supermarket chain, or Barilla, the main Italian pasta maker. Still another vehicle routing application was developed by BiosGroup for the French company Air Liquide. Other interesting real-world applications are those by Gravel, Price and Gagn´e , who have applied ACO to an industrial scheduling problem in an aluminum casting center, and by Bautista and Pereira , who successfully applied ACO to solve an assembly line balancing problem with multi-objective function and constraints between tasks for a bike assembly line.

Problems in ANT COLONY OPTIMISATION

- Dynamic optimization problems
- Stochastic optimization problems
- Multi-objective optimization
- Parallel implementation
- Continuous optimization

In ACO, a number of artificial ants build solutions to an optimization problem and exchange information on their quality via a communication scheme that is reminiscent of the one adopted by real ants.

2.5 Particle Swarm Optimization

James Kennedy and Russell Eberhart

This paper introduces a method for optimization of continuous nonlinear functions. The method was discovered through simulation of a simplified social model; thus the social metaphor is discussed, though the algorithm stands without metaphorical support. This paper describes the particle swarm optimization concept in terms of its precursors, briefly reviewing the stages of its development from social simulation to optimizer.

A number of scientists have created computer simulations of various interpretations of the movement of organisms in a bird flock or fish school. Notably, Reynolds and Heppner and Grenander presented simulations of bird flocking. Reynolds was intrigued by the aesthetics of bird flocking choreography, and Heppner, a zoologist, was interested in discovering the underlying rules that enabled large numbers of birds to flock synchronously, often changing direction suddenly, scattering and regrouping, etc. Both of these scientists had the insight that local processes, such as those modeled by

cellular automata, might underlie the unpredictable group dynamics of bird social behavior. Both models relied heavily on manipulation of inter-individual distances; that is, the synchrony of flocking behavior was thought to be a function of birds' efforts to maintain an optimum distance between themselves and their neighbors.

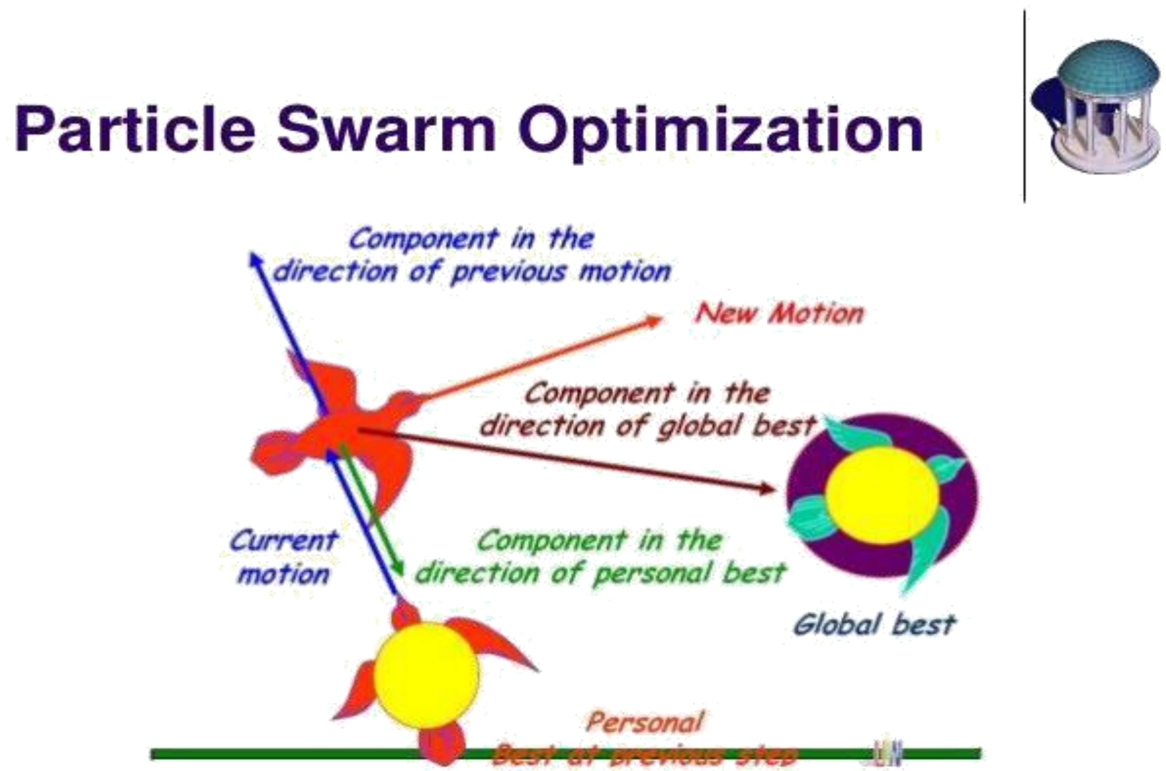


Figure 8. Pso

2.6 BEE COLONY OPTIMIZATION PART I: THE ALGORITHM OVERVIEW

Yugoslav Journal of Operations Research

This paper is an extensive survey of the Bee Colony Optimization (BCO) algorithm, proposed for the first time in 2001. BCO and its numerous variants belong to a class of nature-inspired meta–heuristic methods, based on the foraging habits of honeybees.

The nature-inspired algorithms are motivated by a variety of biological and natural processes. Their popularity is based primarily on the ability of biological systems to efficiently adapt to frequently changeable environments. Evolutionary computation, neural networks, ant colony optimization, particle swarm optimization, artificial immune systems, and bacteria foraging algorithm are among the algorithms and concepts that were motivated by nature. Swarm behavior is one of the main features of different colonies of social insects (bees, wasps, ants, termites). This type of behavior is principally characterized by autonomy, distributed functioning, and self-organizing. Swarm Intelligence [5, 6] is the area of Artificial Intelligence based on studying actions of individuals in various decentralized systems. When creating Swarm Intelligence models and techniques, researchers apply some principles of the natural swarm intelligence.

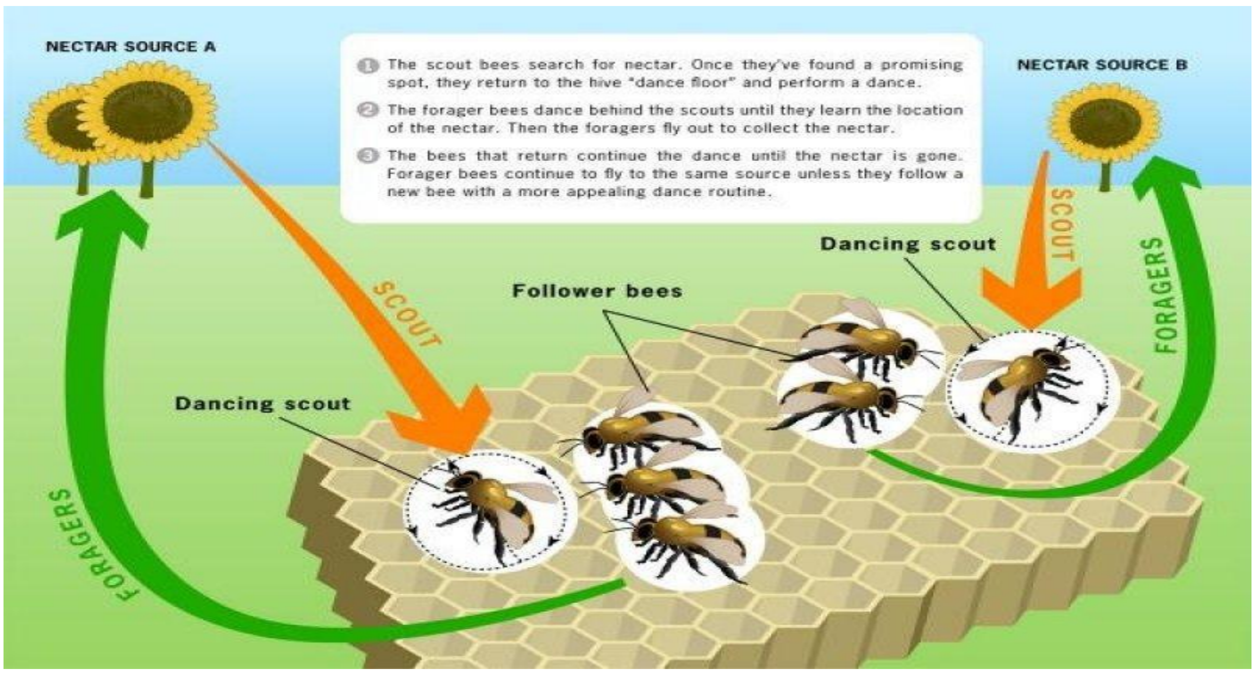


Figure 9 Honey Bee Optimization

CHAPTER 3 SYSTEM DEVELOPMENT

3.1 ALGORITHMS

Several ACO algorithms have been proposed in the literature. Here we present the original Ant System. The original ant colony optimization algorithm is known as Ant System and was proposed in the early nineties. Since then, several other ACO algorithms have been proposed.

The Ant Colony Optimization Metaheuristic

Set parameters, initialize pheromone trails

While termination condition not met **do**

ConstructAntSolutions

ApplyLocalSearch (optional)

UpdatePheromones

End while

Ant System Algorithm

Pheromone values are updated by all the ants that have completed the tour.

$$\tau_{ij} \leftarrow (1 - \rho) \cdot \tau_{ij} + \sum_{k=1}^{m} \Delta\tau_{ij}^{k}$$

Where

ρ is the evaporation rate

m is the number of ants

$\Delta\tau_{ij}^{k}$ is pheromone quantity laid on edge (i, j) by the kth ant

$$\Delta\tau_{ij}^{k} = \begin{cases} \dfrac{1}{L_k}, & \text{if ant k travels on edge i, j} \\ 0, & \text{otherwise} \end{cases}$$

Where $L_k$is the tour length of the k^{th} ant.

In the construction of a solution, ants select the following city to be visited through a stochastic mechanism. When ant k is in city i and has so far constructed the partial solution sp, the probability of going to city j is given by:

$$P_{ij}^{k} = \begin{cases} \dfrac{\tau_{ij}^{\alpha} \cdot \eta_{ij}^{\beta}}{\sum_{c_{il} \in N(s^p)} \tau_{ij}^{\alpha} \cdot \eta_{ij}^{\beta}}, & if\, c_{ij} \in N\,(s^p) \\ 0, & otherwise \end{cases}$$

Where N(sp) is the set of feasible components; that is, edges (i, l) where l is a city not yet visited by the ant k. The parameters α and β control the relative importance of the pheromone versus the heuristic information η_{ij} , which is given by:

$$\eta_{ij} = \frac{1}{d_{ij}}$$

Where d_{ij} is the distance between cities j and i.

3.2 Flow Chart

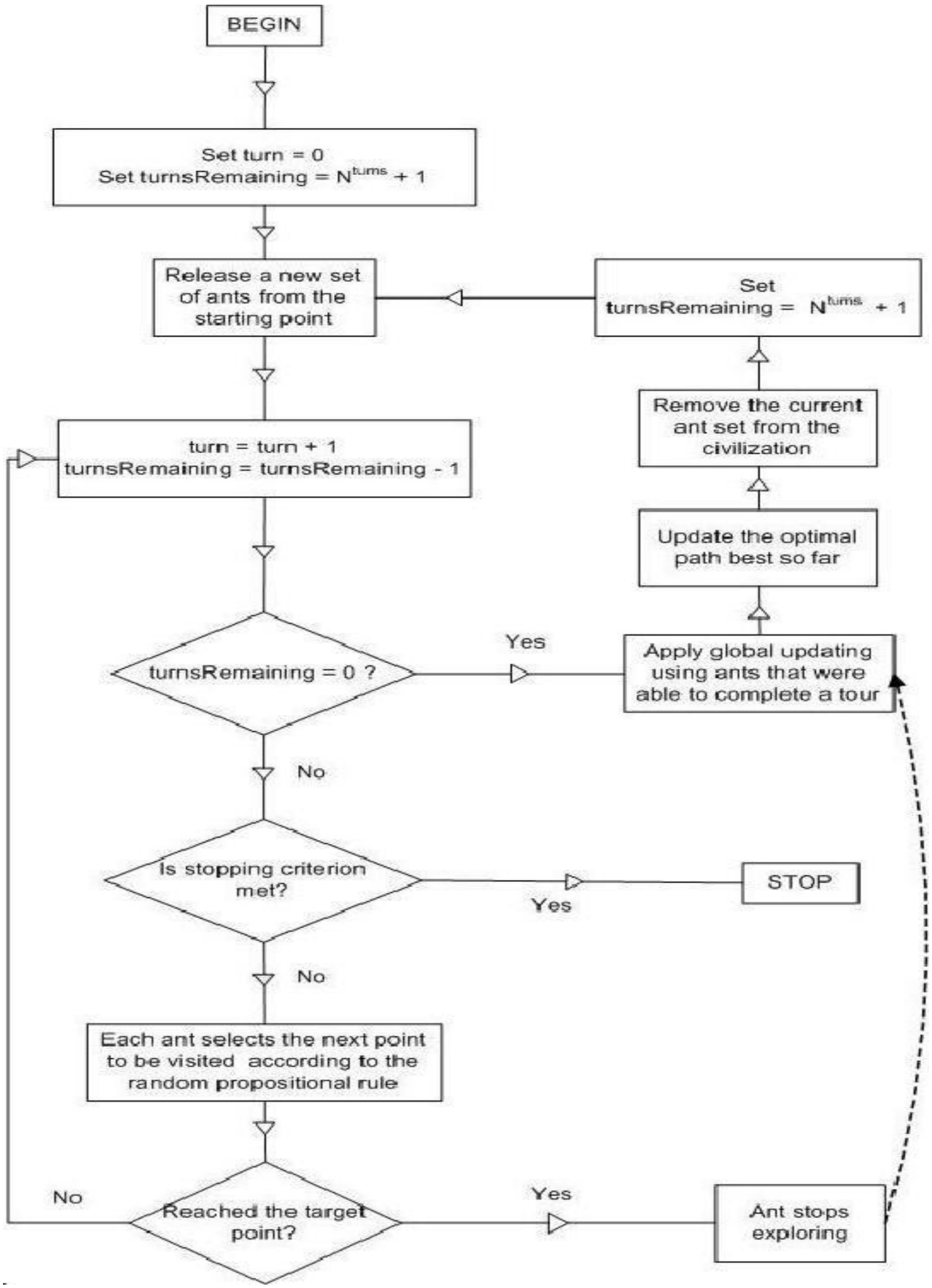


Figure 9. Flowchart aco

3.3 PSEUDO CODE

Input: ProblemSize, $Population_{size}$, m, ρ, β, σ, q0

Output: P_{best}

P_{best} ← CreateHeuristicSolution (ProblemSize)

P_{best} ← Cost (S_h)

$$Pheromone_{init} \leftarrow \frac{1.0}{ProblemSize\ *\ Pbest_{cost}}$$

Pheromone InitializePheromone ($Pheromone_{init}$)

While (¬StopCondition ())

For ($= 1$ **To** $\boldsymbol{m}$)

S_i ← ConstructSolution(Pheromone, ProblemSize, β,q0)

Si_{cost} ← Cost(S_i)

If ($Si_{cost} \leq Pbest_{cost}$)

$$Pbest_{cost} \leftarrow Si_{cost}$$

$P_{best} \leftarrow S_i$

End

LocalUpdateAndDecayPheromone(Pheromone, S_i , Si_{cost},σ)

End

GlobalUpdateAndDecayPheromone(Pheromone, P_{best} , $Pbest_{cost}$,ρ)

End

Return (P_{best})

3.4 SOFTWARE REQUIREMENTS

The software requirements of the project include:

- JAVA
- Cloudsim

Java

Cloud computing explains the latest delivery model, supplement as well as consumption for computing services depending upon the Internet network. Java has now been the latest programming language for quite a long time which provisions the structure for applications related to Web, and recently development of Java has even reached the applications of cloud.

The rapid growth in development of web services has been a popular trend that has completely been changing the field of playing, as adoption movement and cloud conversion from Java EE 7 to Java EE 8 are growing. Some self-describing "late adopters," and firms, have informed stability and security issues as critical risks in transporting the platform of their development. To tackle the fear of this kind, there has been an era of pouring out of positive comments from the Java community after the release of EE 8, which highly increases features of code simplification by using Lambda Expressions. The recent release of the the open source tool NetBeans 8.0.2 as well as official Java IDE are revamping faster the discussions of migration. With even more easier machine learning, about 30% less programming code required with Java 8 (no need to describe more effective code), and lessened code complexity for Java and Spark programmers who might not have specific professionalism in Big Data earlier, may ultimately create machine learning in approximately half of the code lines with almost identical programs of Hadoop.

CloudSim

CloudSim is the tool for simulation which facilitates the cloud program developers to check the operation of all their policies for provisioning in a controllable and repeatable surrounding, for free of any expenditure. It also helps keep a check on the bottlenecks before deployment in real time for the real world. Its a simulator; therefore, it does not compile and then run any original software. In nutshell, it may be described as 'compiling and running a module of a surrounding in a module of hardware', where technical details are encapsulated.

CloudSim is basically a library for various cloud scenario simulations. Moreover, it provides critical classes for defining the computational resources, data centres, applications, users, virtual machines, and schemes for the handling of different constituents of the system including provisioning and scheduling. Running these constituents, it is easier to analyse latest introduced strategies monitoring the working of clouds, as also considering policies of load balancing, schemes, scheduling algorithms, ,etc. It can further be used for assessing the complexity of different strategies from typical perspectives including execution time of application, cost, etc. It even helps in the evaluation process of the Green IT schemes. It may also be used to work as a fundamental base for cloud environment that has to be simulated and may even add latest schemes for new scenarios, scheduling, and load balancing. It is scalable to work as a library which facilitates the user for adding a scenario desired by programming in JAVA. Using CloudSim, firms, industry-based programmers and R&D departments may also check the operation of a latest programmed application in an easy set-up as well as controlled environment.

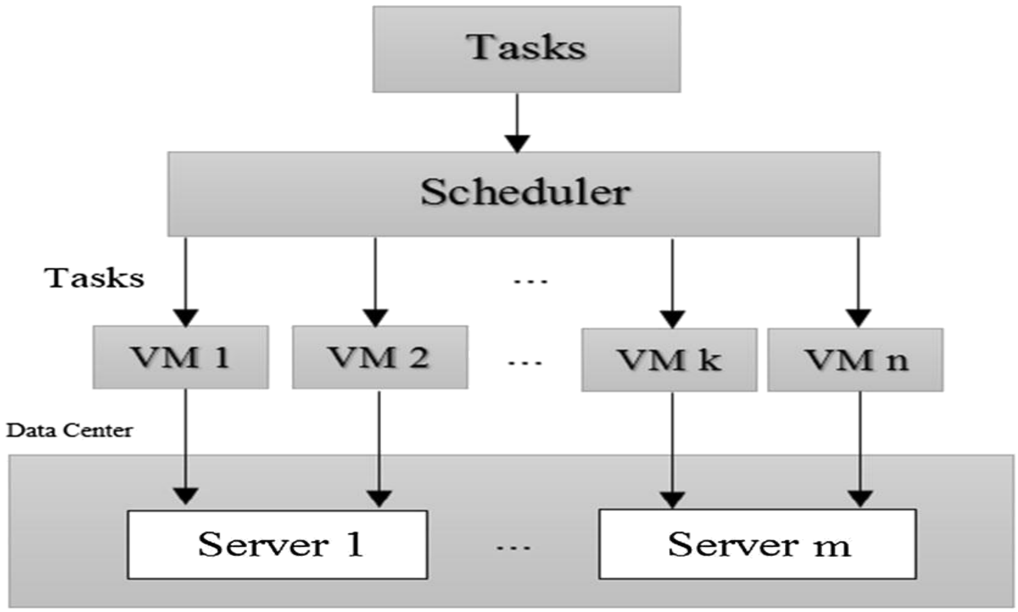


Fig.10: Task scheduling in cloud environment

CHAPTER 4 PERFORMANCE ANALYSIS

Initial task is task scheduling in local environment which has been done using algorithms, namely Round Robin, Existing ACO and Proposed ACO These algorithms are run on the CloudSim simulator having 4 datacenter.

4.1 Test Plan

1.With the use of **cloudsim 3.0.3** we are here doing comparative study of some algorithms used in scheduling such as:

1. Existing ACO

2. Round Robin

3. Proposed ACO

2. **Number of virtual machines**

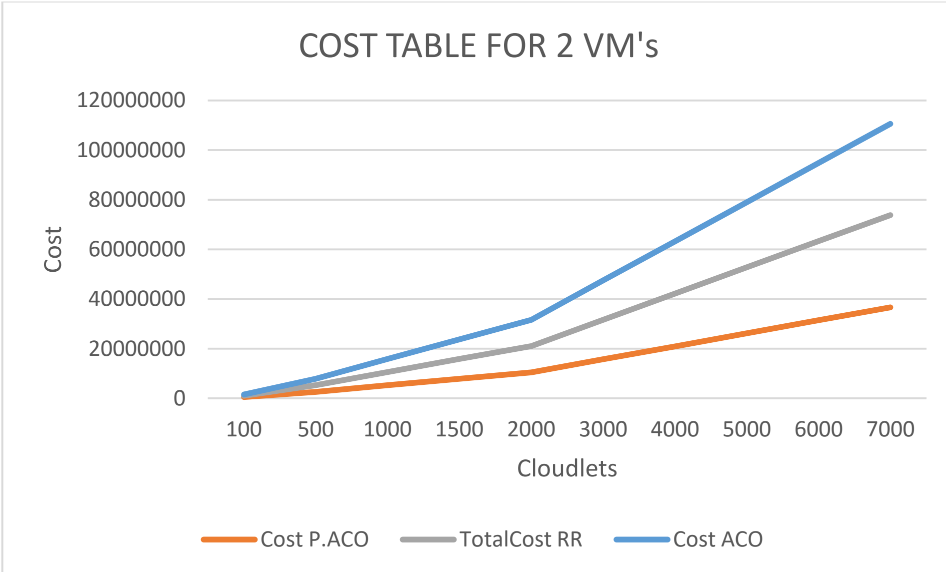
The number of virtual machines was set to 2 for datacenter 2 and 10 to datacenter 4.

3**. Cloudlets**: Range between 200 to 7000.

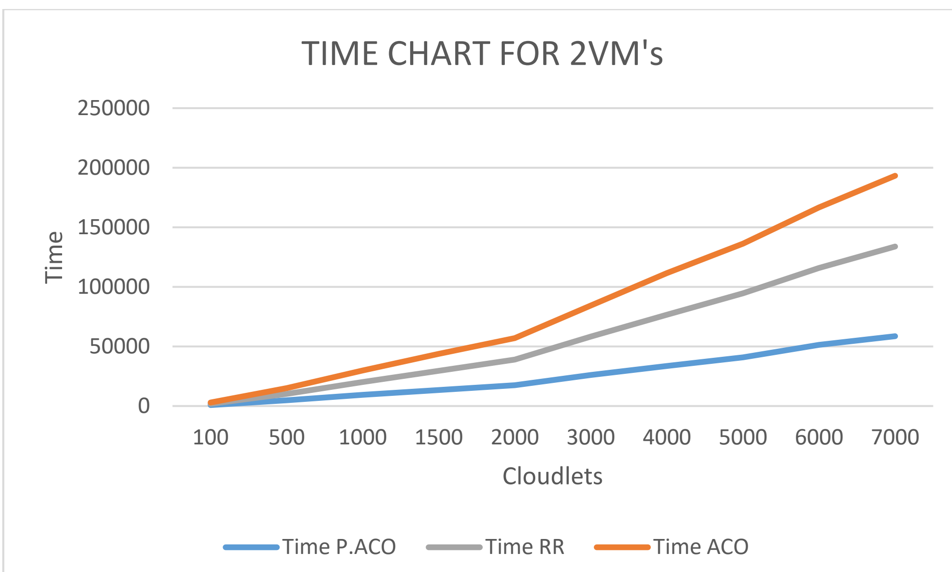
(i) Comparison of ACO , Proposed ACO , Round Robin when Vms are 2, datacenter are 2.

	ACO			Proposed ACO			ROUNDROBIN	
Cloudlets	**VM =2**			**VM =2**			**VM =2**	
	Cost ACO	Time ACO		Cost P.ACO	Time P.ACO		Cost RR	Time RR
100	527300.32	939.1		527300.32	939.1		530956	1075.1
500	2641286.24	4873.1		2641205.6	4870.1		2654780	5375.1
1000	5279427.519	9629.1		5275825.6	9495.1		5309559.9	10750.1
1500	7910821.9	14134.1		7.89E+06	13513.1		7964340	16125.1
2000	1.05E+07	17939.1		1.05E+07	17650.1		1.06E+07	21500.1
3000	1.58E+07	26269.1		1.58E+07	26028.1		1.59E+07	32250.1
4000	2.10E+07	34827.1		2.10E+07	33713.1		2.12E+07	43000.1
5000	2.62E+07	41585.1		2.62E+07	41016.1		2.65E+07	53750.1
6000	3.15E+07	50936.1		3.15E+07	51338.1		3.19E+07	64500.1
7000	3.67E+07	59486.1		3.67E+07	58644.1		3.72E+07	75250.1

Table 2. Comparison of algo with 2 vms



Graph 1. Cost using 2 vms

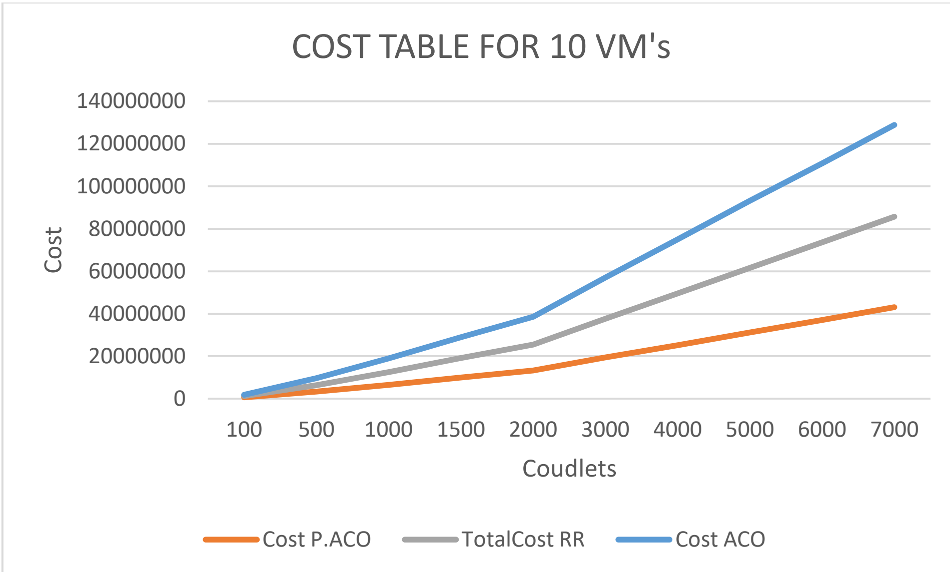


Graph 2. Time using 2 vms

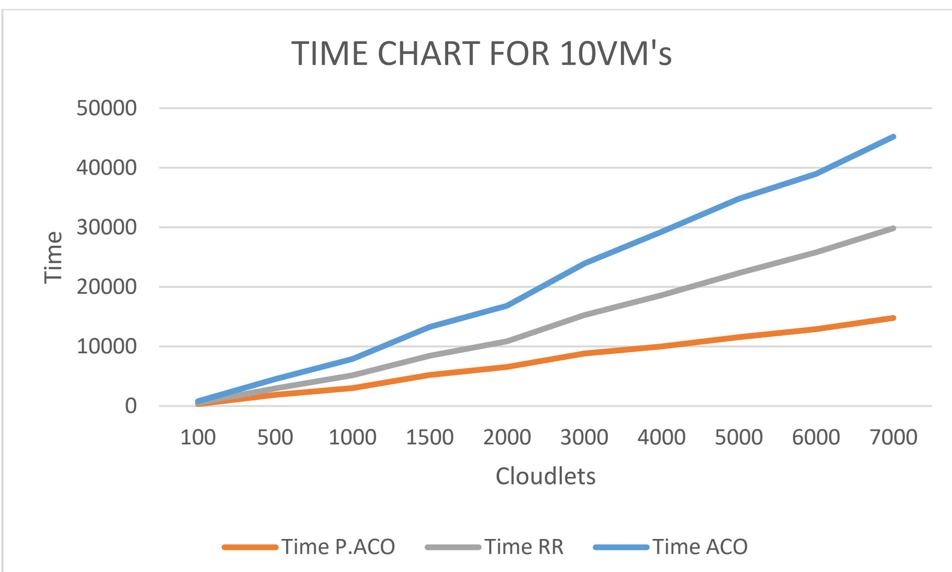
(ii) Comparison of ACO , Proposed ACO , Round Robin when Vms are 10, datacenter are 4.

	ACO			**RoundRobin**			**P.ACO**	
Cloudlets	**VM =10**			**VM =10**			**VM =10**	
	Cost ACO	Time ACO		Cost P.ACO	Time P.ACO		TotalCost RR	Time RR
100	649916.32	277.2		674161.86	336.2		608541.86	215.2
500	3268058.026	1527.2		3386259.94	1923.2		3042709.33	1075.2
1000	6434084.9	2742.2		6522212.31	3024.2		6085418.66	2150.22
1500	9838212.79	4821.2		1.00E+07	5251.2		9128127.99	3225.22
2000	1.31E+07	5938.2		1.33E+07	6585.2		1.22E+07	4300.25
3000	1.94E+07	8654.2		1.95E+07	8819.2		1.83E+07	6450.2
4000	2.54E+07	10639.2		2.52E+07	10037.2		2.43E+07	8600.2
5000	3.15E+07	12420.2		3.12E+07	11599.2		3.04E+07	10750.2
6000	3.72E+07	13158.2		3.71E+07	12929.2		3.65E+07	12900.2
7000	4.33E+07	15374.2		4.30E+07	14784.2		4.26E+07	15050.2

Table3. Comparison of algo with 10 vms



Graph 3. Cost using 10 vms



Graph 4. Cost using 10 vms

CHAPTER 5 CONCLUSION

The design of an ACO scheduling algorithm to deals with the Task-Resource assignment in such a way that the total cost & time is minimized Simulation results demonstrate that ACO algorithm is out performing Existing ACO & RoundRobin.

Cloud computing is a big shift from the traditional way businesses think about IT resources. Cloud computing eliminates the capital expense of buying hardware and software and setting up and running on-site datacenters—the racks of servers, the round-the-clock electricity for power and cooling, the IT experts for managing the infrastructure. It adds up fast. Most cloud computing services are provided self service and on demand, so even vast amounts of computing resources can be provisioned in minutes, typically with just a few mouse clicks, giving businesses a lot of flexibility and taking the pressure off capacity planning.

we have evolved the CloudSim toolkit for simulating and modeling extensible Clouds. The research papers that are published by different authors relating to the Cloud are read thoroughly and different log files available on the internet of various sizes are downloaded and incorporated in the project. A graphical user interface is designed wherein freedom to input the number of instructions he/she wants to work upon is given.

5.2 FUTURE SCOPE

In this Study based on the results, it can be also concluded that there is not a single scheduling algorithm that provides superior performance with respect to various types of quality services. This is because job scheduling algorithms needs to be selected based on its capability to ensure good aspect of services with reasonable cost and maintain fairness by fairly distribute the available resources among all the jobs and respond to the constraints of the users. Existing scheduling algorithm gives high throughput and cost effective but they do not consider reliability and availability. It involves the development of proposing a method based on Ant Colony optimization to resolve the problem of load balancing in cloud environment. In future work will propose a new algorithm for resource scheduling and comparative with existing algorithms. The efficiency of the user request first may be optimized the processor and execute the request.

In the future the project can be enhanced by implementing a GUI or a graphical user interface in which the user may input the number of instructions he/she wants to execute from the log file at a particular time rather than executing the whole lot of instructions. Furthermore, when the project is completed using the CloudSim toolkit it is tested on Google dataset i.e. the real time dataset. Rather than finding dummy results on dummy datasets we work on the real time data set.

REFERENCES

1. M. Dorigo. Optimization, Learning and Natural Algorithms. PhD thesis, Politecnico di Milano, Italy, 1992.
2. M. Dorigo and Thomas Stutzle, "Ant Colony Optimization", MIT Press Cambridge, 2004.
3. M.Dorigo, T.Stutzle, “The Ant Colony Optimization Metaheuristic: Algorithms, Applications , and Advancers”, Handbook of Metaheuristics, 2002.
4. Anirudh Shekhawat, Pratik Poddar, Dinesh Boswal “Ant colony Optimization Algorithms: Introduction and Beyond”, IIT Bombay.
5. Saad Ghaleb Yaseen, Nada M.A.AL – Slamy “Ant Colony Optimization” IJCSNS International Journal of Computer Science and Network Security, VOL.8 No.6, June 2008.
6. Marco Dorigo, Luca Maria Gambardella “Ant Colony System:A Cooperative Learning Approach to theTraveling Salesman Problem”, Belgium, TR/IRIDIA/1996-5
7. Computing Services And Deployment Models Ch Chakradhara Rao , Mogasala Leelarani , Y Ramesh Kumar.
8. Enhanced ant colony algorithm for cost-aware dataintensive service provision Lijuan Wang, Jun Shen ,Ghassan Beydoun.
9. Bhoi U, Ramanuj PN. Enhanced max-min task scheduling algorithm in cloud computing. International Journal of Application or Innovation in Engineering and Management. 2013 Apr;2(4):259-64.
10. Chen H, Wang F, Helian N, Akanmu G. User-priority guided Min-Min scheduling algorithm for load balancing in cloud computing. InParallel Computing Technologies (PARCOMPTECH), 2013 National Conference on 2013 Feb 21 (pp. 1-8). IEEE.
11. Bhoi, U., & Ramanuj, P. N. (2013). Enhanced max-min task scheduling algorithm in cloud computing. International Journal of Application or Innovation in Engineering and Management, 2(4), 259-264.

12. Gupta, P., & Ghrera, S. P. (2015). Load and Fault Aware Honey Bee Scheduling Algorithm for Cloud Infrastructure. In Proceedings of the 3rd International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA) 2014 (pp. 135-143). Springer International Publishing.

13. Gao, Y., Guan, H., Qi, Z., Hou, Y., & Liu, L. (2013). A multi-objective ant colony system algorithm for virtual machine placement in cloud computing.Journal of Computer and System Sciences, 79(8), 1230-1242.

14. Jena, R. K. (2015). Multi Objective Task Scheduling in Cloud Environment Using Nested PSO Framework. Procedia Computer Science, 57, 1219-1227.