Chapter 1
Introduction

This chapter commences with an overview of cloud computing comprising of its definitions, historical view, prominent characteristics, underpinning technologies that led to the advancement of cloud computing, various service models and deployment models. The chapter also covers the cloud reference architecture suggested by National Institute of Standards and Technology (NIST) focusing on the role played by each cloud entity. Next it identifies the major obstacles such as resource scheduling and security which are inhibiting the cloud to realize its full potential. Further it addresses the issues pertaining to scheduling workflow applications in clouds followed by the basic concept of Intelligent Water Drops (IWD) algorithm. The chapter culminates with thesis organization and contributions.

1.1 Overview of Cloud Computing

Cloud Computing has emerged as a revolutionary technology that provides massive IT resources following ‘pay-per-use’ strategy. It relieves the consumers from the overhead of capacity planning and allows them to acquire or release the resources on the fly as per their fluctuating needs. The resources are delivered as services through Internet [1] which can be accessed from anywhere and at any time. The services are hosted in advanced data centers that exploit virtualization technology to boost the utilization of resources [2]. The provisioning of ready to use infrastructure supports quick deployment of applications without worrying about the orchestration of resources. The additional benefit is immense cost savings with respect to capital and operational expenditures.

The following definition proposed by NIST reflects these aspects of cloud computing:

“Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction” [3].
Another significant perspective of cloud technology is Quality of Service (QoS). As the users’ applications are executed on the infrastructure of external cloud service providers and major aspects of service such as availability and quality are beyond their control, it necessitated guarantees on service delivery. Thus a contract termed as Service Level Agreement (SLA) is negotiated amid cloud vendors and cloud consumers. It describes the responsibilities of cloud service providers and QoS parameters for providing the service. Another definition given by Buyya et al. highlights this notion of cloud computing:

“A Cloud is a type of parallel and distributed system consisting of a collection of interconnected and virtualized computers that are dynamically provisioned and presented as one or more unified computing resources based on service-level agreements established through negotiation between the service provider and consumer” [4].

John McCarthy perceived the idea of cloud computing in 1960s when he envisioned computing capabilities being provided as a public utility alike water, telephone and electricity. But it actually came into existence with the advent of grid computing and utility computing. Other key factors that contributed in its emergence are advancement in virtualization technology and availability of high-speed Internet connectivity. The following section explores cloud computing from a historical perspective.

1.1.1 Cloud Computing – A Historical Perspective

The origin of cloud computing could be traced back to 1950s when several users used to access the computing resources of mainframes through dummy terminals [5]. Due to unaffordable cost of a mainframe, organizations could not provide it individually to each employee. Offering shared access to a mainframe computer was the only feasible solution at that time. In 1960s, the idea behind cloud technology was first suggested by John McCarthy [6]. However due to technological limitations, it could not take its present shape.

In 1969, “ARPANET (Advanced Research Projects Agency Network)”, the antecedent of Internet was proposed by J.C.R Licklider who intended to interconnect scientific users of various institutions [7]. The 1970s witnessed the invention of virtualization technology. IBM launched the VM operating system which supported the simultaneous operation of multiple virtual machines isolated from each other on a single physical machine. These machines could run different operating systems while sharing the computing resources of the host machine. It was around 1990 that telecom companies began
to provide VPN connections which further aided in offering multiple users with shared access to given infrastructure. Also Internet availability and server virtualization became more common. Prof. Ramnath Chellappa is attributed to coin the term “Cloud Computing” in 1997 and explained it as “computing paradigm where the boundaries of computing will be determined by economic rationale rather than technical limits alone” [8].

The arrival of grid computing in 1999 is often believed as the precursor of cloud technology [5]. Grids utilize the computing resources of interconnected servers worldwide to solve computation-intensive problems. Cloud computing actually came into existence in 2006 with the inception of Amazon’s Elastic Compute Cloud (EC2) enabling the users to lease computing infrastructure for running their applications [6]. By 2009, other enterprises like Google, Microsoft and Oracle also joined the Cloud market and launched various cloud services. Figure 1.1 depicts the gradual evolution of cloud computing from mainframe computing.

**Figure 1.1 Evolution of Cloud Computing from Mainframe Computing**
The growth of clouds has drastically transformed the way organizations accomplish their computing requirements and manage IT assets. It has emerged as a disruptive innovation [9], substituting traditional software licensing and locally managed infrastructure for delivering software, platform and infrastructure as services. Elasticity of resources and usage based pricing are the key business drivers of this model which alleviates the burden of consumers to manage scaling up and down of in-house infrastructure and enables the development of applications in a cost-efficient manner.

1.1.2 Salient Characteristics of Cloud

The five salient characteristics of cloud which distinguish it from other distributed technologies are [3]:

On-demand Self-Service: The computing capabilities like CPU time, storage etc. can be acquired by users as per need without any human interaction with the cloud vendor. It is generally performed individually using web-based portal.

Broad Network Access: The users can gain access to the computing capabilities over the Internet through a number of client devices such as laptops, desktops and mobile phones.

Resource Pooling: The computing capabilities are shared among users using virtualization or multi-tenancy model with resource provisioning as per demand of the users. Users remain oblivious of the precise location of the acquired resources.

Rapid Elasticity: A user can acquire and release resources on the fly as per fluctuating needs. It gives a feeling of apparently unlimited resources which can be provisioned in appropriate quantity at any time. It frees the users from the burden of provisioning plan.

Measured Service: Cloud service providers monitor and measure the utilization of resources using automated tools which is further reported to the users to keep transparency. They also optimize the usage of resources employing scheduling strategies. The users are billed for the usage of resources as per pricing model.

1.1.3 Service Models

Service Models describe the kind of service that the service provider is offering. According
to the abstraction level of the resources, these can be categorized as follows and also depicted in Figure 1.2.

a) **Infrastructure as a Service (IaaS):** This model enables the availability of high performance computing infrastructure (servers, networking technology and storage) on rental basis. It exploits virtualization technology to manage the infrastructure resources. Users are offered various options for operating systems and are allowed to run arbitrary applications. Presently, the most recognized IaaS provider is Amazon whose EC2 service offers virtual machines that can be customized by installing required software [10]. Other examples in this category are Rackspace, Microsoft, Google and Gogrid.

![Figure 1.2 Cloud Service Models](image)

b) **Platform as a Service (PaaS):** It renders development environments to users for creating and deploying their applications [10]. The users need not have to be concerned about the orchestration of hardware as it is managed and controlled by service provider. One of the prominent PaaS offering is Google AppEngine, which enables the users to implement web applications in programming languages like Java.
or Python. Microsoft Azure also offers programming environments for creating applications.

c) **Software as a Service (SaaS):** It offers software applications on subscription basis that can be used via web browser. Cloud provider manages underlying cloud infrastructure and application platform. Users are also free from the burden of software installation and maintenance. Moreover, a user has a flexibility to assess a new software by paying rent and then continue its usage if it seems appropriate for him [12]. Salesforce.com delivers Customer Relationship Management (CRM) applications based on SaaS model which can be customized as required by users. Other major SaaS applications are Google Docs, Google Maps, Microsoft Windows Live.

### 1.1.4 Deployment Models

Based on location and management of infrastructure, there can be four deployment models of cloud which are defined below [1]. The user can opt for any of these models depending upon his need and Figure 1.3 depicts the pictorial view of cloud deployment models.

- **a) Public Cloud:** This model provides computing facilities to a single user or an organization as services on subscription basis. These are cost-efficient as well as provide highly scalable infrastructure but have security concerns and provides limited customization of applications.

- **b) Private cloud:** It is developed for a single organization and provides unified access to its resources from various locations or divisions of the organization. It offers a lot of flexibility and has a tremendous value from security point of view. It is known as on-site private cloud in case it is located on-premises and governed by the organization itself. If a third party accommodates the resources, it is termed as outsourced private cloud.

- **c) Community Cloud:** It allows the cloud setup to be collaboratively utilized by a set of industries having similar needs and considerations (e.g. compliance and security concerns). It can be possessed and governed by one or more industries in the group or an external service provider. Former is defined as on-site community cloud and latter is named as outsourced community cloud.

- **d) Hybrid Cloud:** It is an integration of diverse cloud models (private, public or community) that are interconnected to allow data and application portability. For
example, an organization could use its private cloud to store sensitive data and may opt public cloud for archive data.

Figure 1.3 Cloud Deployment Models [13]

1.1.5 Underpinning Technologies

The key technologies that propelled the growth of cloud computing (see Figure 1.4) are virtualization, service-oriented architecture, grid technology, utility computing and autonomic computing. This section discusses development of these underpinning technologies and their contribution in the inception of cloud computing.

1.1.5.1 Virtualization

Virtualization allows the creation of multiple virtual instances on a single server. Each virtual instance acts similar to a physical server having its own operating system (OS) known as guest OS and a software stack. A software layer called hypervisor enables the coordination between guest OS and physical hardware [10] as represented in Figure 1.5.
Virtualization is aimed to enhance utilization of servers by running multiple VMs over a single server. It is termed as server consolidation. This helps in minimizing the total number of servers required and thus optimizing floor space and costs. Additional advantages include better security and reliability. VMs run in complete isolation from each other which results in increased security. Reliability is improved as failures inside one VM do not have an impact on other VMs.

![Diagram](image)

*Figure 1.4 Technologies that propelled the growth of cloud computing [10]*

The state of a VM RAM, disks along with configuration files at a moment can be saved in the form of an image [14], which can later be restored. It allows for moving and copying the virtual machines from one server to another, termed as VM migration. It has its applications in load balancing, hardware maintenance and disaster recovery.

Virtualization also supports quick provisioning and deployment of VMs either by means of a preconfigured VM image or by cloning a running VM. These hefty benefits of
virtualization promote its use in cloud data centers and in overcoming related operational issues.

![Architecture of a Virtualized Server](image)

**Figure 1.5 Architecture of a Virtualized Server** [10]

### 1.1.5.2 Service-Oriented Architecture

Service-Oriented Architecture (SOA) is a technique of designing software which supports development of components as services [15]. A component is a well-defined and independent module that offers required business functionality. They are loosely coupled and can interact with each other through messages. Existing services could be reused to build complex applications that enables the businesses to respond rapidly to market opportunities while saving both time and money.

SOA is generally implemented using web services. Web services consists of standards for defining the format of messages and for implementing service interfaces. Simple Object Access Protocol (SOAP) /Web Services Description Language (WSDL) and REpresentational State Transfer (REST) are two well-known approaches for describing web services. WSDL is used to define web services and the methods to access them. SOAP is a protocol for enabling transfer of information in a distributed environment. However, REST exploits HTTP semantics for implementing web services.

Cloud Computing follows service-oriented architecture so that infrastructure, platform and software can be provided as modular services with well-defined interfaces to be used by anyone. Thus the consumer need not bother about the details of the service and
can consume it like a black box service. Further they can combine the services from multiple providers to build composite applications. For instance, services such as user authentication, google maps, gmail could act as building blocks for creating new applications if existing ones do not satisfy all the requirements.

1.1.5.3 Grid Technology

Grid computing realizes the sharing of compute and storage resources that pertain to several administrative domains in a transparent manner. They are generally placed at remote geographical locations connected with a computer network. Its main aim is to support complicated applications requiring high computation power and data storage. Standard Web-based protocols are utilized for managing and provisioning of the distributed resources in a grid. SETI@Home [16] is a research project that utilizes grid computing to detect radio signals looking for evidence of intelligent life beyond earth. Here the grid is formed via the unused computing power of millions of computers connected through Internet whose users voluntarily participated in this project from worldwide.

The advances in grid technology have enabled it to deliver computing resources through Internet as per demand of the user. However, the major barrier is assuring QoS due to its inability to isolate performance of different users [10]. One more obstacle is access to resources which are having different software configurations that gives birth to interoperability problems [14]. Virtualization is the key to all these concerns of the grids and can aid in realizing the vision of offering service-oriented computing environments.

1.1.5.4 Utility Computing

It is a business paradigm under which users are offered computing capabilities as per their need and are required to pay based on resource consumption. The service provider manages infrastructure and is liable for enforcing QoS constraints while providing services. The consumers are free from initial investment of buying the infrastructure and afterwards alleviated from the burden of its maintenance and upgradation. The main aim of this model is to make available computing resources as conventional utilities.

1.1.5.5 Autonomic Computing

Autonomic computing can be described as the capability of a computing system to manage
itself with very low or no manual intervention. The system keeps on monitoring its state for an unpredictable change and takes corrective action based on stated policies which are defined by technical personnel controlling the system. This is why its architecture is also termed as Monitor-Analyze-Plan-Execute (MAPE). IBM introduced this concept around 2001 with the goal of developing self-managing computer systems so that the problems pertaining to system management can be avoided. IBM has proposed four significant features for these systems: self-configuration, self-optimization, self-healing and self-protection [17], [18].

The automated management of huge data centers of clouds is motivated from the idea of autonomic computing. VM provisioning and migration, failure management and capacity planning in data centers are some of these tasks.

1.2 Cloud Reference Architecture

NIST published the reference architecture [19] for cloud in September 2011 after analyzing various prevailing cloud models recommended by researchers and cloud service providers. The architecture addresses the operational perspective of cloud computing. Figure 1.6 displays the reference architecture of cloud ecosystem that is composed of five main entities: cloud consumer, cloud provider, cloud auditor, cloud broker and cloud carrier.

Following are the details of the specific role played by each actor which could be a person or an industry.

Cloud Consumer: A consumer can request services from the cloud provider and has to pay in accordance with consumption of services. The type of service invoked decides the control provided to the consumer. SaaS users have limited control over application configuration settings. PaaS user has complete control of deployed applications and partial control of hosting environment. IaaS users are permissible to manage OS, storage and applications at VM level as well as host firewalls but cannot directly access physical servers.

Cloud Provider: A service provider procures and oversees the required infrastructure and makes it available to consumers. SaaS providers administer infrastructure, application platform as well as software applications. PaaS providers are accountable for hosting environment along with infrastructure setup. IaaS providers are liable for managing
hardware and software required for provisioning of infrastructure such as hypervisors for creating VMs. There are mainly five activities of providers as discussed below:

- **Service Deployment:** Cloud Providers offer four kinds of service deployment models: Private, Public, Community and Hybrid which are categorized on the basis on access boundary.

- **Service Orchestration:** It is meant to aid organization and synchronization of computing resources and is divided into three layers. Service layer is the uppermost layer that provides interfaces for accessing various kinds of cloud services (IaaS, PaaS and SaaS). Intermediate layer is resource abstraction and control layer. It addresses the abstraction of hardware resources using virtualization software. It is also liable for provisioning and monitoring the utilization of resources. The bottom layer is physical resource layer that comprises of hardware i.e. servers, storage and networking components, along with facilities i.e. electricity, ventilation and cooling etc.
• Service Management: Service management module provides business support (such as managing customer accounts), provisioning and configuration of requested capabilities as well as means to facilitate data portability and service interoperability.
• Security Management: It deals with the security requirements of the cloud ecosystem namely authentication, confidentiality, data integrity and incident response etc.
• Privacy Management: Privacy management is concerned with non-disclosure of consumer’s sensitive information.

*Cloud Broker:* A cloud broker enacts as an intermediary between cloud consumer and cloud provider. When a consumer desires the services from several cloud vendors, the cloud broker hides the complexity of data integration and assures secure delivery of services.

*Cloud Auditor:* Cloud provider and consumer mutually agree upon some policies and norms with respect to security, privacy and performance of cloud applications. A cloud auditor is an external entity who evaluates the cloud services based on these regulations and ensures the fulfilment of legal requirements.

*Cloud Carrier:* Cloud carrier is responsible for providing secure network connection between cloud consumer and cloud vendor to access the cloud services. Cloud vendor negotiates SLAs with cloud carrier pertaining to service availability which should comply with SLAs signed with consumers.

### 1.3 Major Impediments to Cloud Adoption

Although cloud technology has gained wide acceptance among individual users as well as organizations, yet there are some issues that require attention to relish its long-term benefits. The major impediments towards development and success of cloud computing are:

• **Resource Scheduling:** Resource scheduling facilitates optimal provisioning of resources to users in a finite time. Service providers employ various techniques for scheduling resources to satisfy their objectives as well as quality of service requirements specified by consumers. Researchers are studying scheduling
algorithms catering to stated requirements. For instance, deadline and budget constrained scheduling strategies assign resources to user jobs such that execution time and cost remain within user defined deadline and budget respectively.

- **Security and Privacy**: Data security and privacy are of grave concern for consumers while transition to the cloud environment. This is due to the fact they remain unaware of the location of their data as well as the cloud infrastructure is virtualized and shared among multiple consumers. Hence it is essential that data should be protected from unauthorized access, modification, deletion or insertion [20]. Privacy of consumer should be ensured while accessing sensitive data by thwarting intruders from deducing his behavior by analyzing his visit pattern [21]. Storing data in a cloud also presents numerous legal and compliance issues. Data may be placed in any of the data centers of service provider which could be in Asia, Australia or somewhere else. This precept of cloud computing may interfere with legal necessities of some countries. For instance, laws of a country may require that sensitive data like patient medical records cannot be stored outside its territory.

- **Vendor Lock-in**: It is a state wherein a consumer utilizing services of a cloud vendor cannot migrate to another because of incompatibility issues between them. These issues arise as a consequence of use of proprietary technologies such as proprietary interfaces and APIs. In case a consumer desires to move out from a vendor who is incapable of satisfying his requirements, high cost and complexity of transferring the applications and data would inhibit his transition to some other vendor [22]. Lack of standardization also makes it problematic to integrate services from different vendors. Developing open standards for cloud is an effective approach to mitigate this problem and avoid interoperability and portability issues.

- **Service Availability**: Consumers desire the services to be highly available when they move to cloud technology. Downtimes due to power outages, faulty software updates and overloading of servers may pose a risk to their expectation and incur huge losses to the consumer. As it is not possible to completely avoid these kinds of failures, a reliable vendor will have robust resilience measures in place to prevent loss of consumer’s data. Moreover, SLA guaranteeing the availability of service
must be negotiated between both the parties and penalties in case of violations must also be mutually agreed upon [10].

- **Performance Unpredictability:** Due to implicit lack of control over resources in cloud, it appears as an opaque system to consumers where they are not entitled to inspect any aspect of the resources. Moreover, cloud follows multi-tenancy model i.e. several consumers are competing for similar resources. As a result, consumers may face various problems pertaining to performance of applications in cloud. For example, AWS EC2 is susceptible to lot of performance issues such as disk I/O performance unpredictability in EBS (Elastic Block Storage), CPU cycles stolen by VM neighbors residing on same server [23]. These concerns can be averted by using efficient techniques for resource monitoring and provisioning.

- **Data Transfer Bottlenecks:** Numerous applications require transmitting large quantity of data between consumer’s site and remote cloud data center. As the data transfers over the basic monthly payment are charged on a per GB basis, these costs can rapidly accumulate, making data transfer costs a significant concern [24]. Data transfer costs can be reduced by shipping disks containing required data. Alternative approach is to explore the possibilities of reducing the cost of network bandwidth. For this purpose, researchers are investigating for inexpensive routers as a replacement to high-end routers.

  High network latency could impact the performance of applications in terms of data transfer. Amazon is taking initiatives to tackle these issues. Its latest offering, Cloudfront [25] aids in reducing latency while transferring content to users.

- **Energy Efficiency:** There is a dramatic increase in energy consumption due to mushrooming of data centers. A typical data center consumes energy equivalent to power 25000 households and emits large carbon footprints [26]. It not only poses a serious threat to the environment but also invites huge investment pertaining to power needed for operating data centers. Thus we need to pay attention to reduce energy usage and carbon footprints before the problem exaggerates. Server consolidation [27] and Dynamic Voltage Frequency Scaling (DVFS) [28] are often
utilized for conserving energy. The challenge is to acquire a balance between minimizing energy consumption and optimizing performance of the applications.

**1.4 Workflow Scheduling in Cloud: Challenges and Motivation**

For successful implementation of any user application in cloud computing environment, one of the most substantial tasks is to generate an efficient schedule before its execution. Scheduling allows optimal allocation of resources among given tasks in a feasible time to attain user’s desired Quality of Service (QoS). Formally, scheduling problem involves tasks that must be scheduled on resources subject to some constraints to optimize some objective function. The aim is to build a schedule that specifies when and on which resource each task will be executed [29].

Scientific community is always striving for performance and cost effective solutions for executing complex scientific applications. As these complex scientific applications require a large number of resources to produce results in a realistic time, distributed computing environments such as grids and clusters have been largely opted in the past few decades. Recently cloud technology has come up as a promising computing platform which offers a cheaper substitute to clusters and a much more scalable, reliable and secure platform as compared to grids.

Workflows have been commonly used to characterize scientific applications from various domains such as bioinformatics, astronomy, physics and seismology [30]. Workflows are normally illustrated as Directed Acyclic Graphs (DAGs) where nodes resemble tasks and links denote data or control dependencies among tasks [31]. Complex workflows are composed of substantial number of tasks requiring huge storage and processing power. Clouds are heterogeneous environments composed of numerous servers, storage and network resources with varying configuration. Workflow Scheduling in cloud implies mapping of workflow tasks to appropriate resources to satisfy user-defined QoS parameters and falls into a category of problems termed as NP-hard problems owing to vast search space.

Scientists get their workflow applications executed without being aware of the underlying intricacies of the cloud ecosystem. Service providers rely on scheduling algorithms to facilitate the efficient allocation of cloud resources to workflow tasks. Since this problem is considered as NP-hard, algorithms require vast amount of time to get an
optimal solution. Moreover, users and service providers are having own requirements which have to be inculcated by the scheduling algorithm. Users may expect optimal performance of their applications in terms of execution time, execution cost, reliability, security or may impose some constraints related to deadline/budget and so on. The major optimization criteria desired by service providers are maximum utilization of their resources, minimizing energy consumption in their data centers and maximum revenue generation from the infrastructure set up etc.

Workflow Scheduling in cloud is challenging owing to optimization of multiple conflicting objectives along with compliance of constraints while mapping tasks to resources in a short period of time. Furthermore, scheduling algorithms need to be devised such that they exploit the inherent characteristics of cloud such as elasticity, apparently unlimited resources and pay per use pricing model. For instance, number of VMs allocated to an application should be scaled up/down as per its requirement. This would facilitate in enhancing application performance, resource utilization and minimizing economic cost [32].

Another related concern is the performance inconsistencies of Virtual Machines (VMs) in clouds as analyzed in [24] [33]. According to their study, the major reasons behind this performance unpredictability are heterogeneity of hardware, virtualization of resources and sharing of non-virtualized resources such as network bandwidth and disk I/O. Dejun et al. [34] have observed that even the identical VM instances show variation in their performance behavior under the similar workloads. This may significantly influence the execution time of workflows and thus possibly allow the applications to cross their deadlines. Scheduling algorithms usually require execution time of all tasks on divergent VMs so that appropriate VM can be selected for every task. Execution time depends on the computing capacity of VM and if there is a variation in compute capacity than actually stated, task would take more time to execute and thus its completion would be deferred. It will also force the child tasks to be delayed which ultimately delays the completion time of workflow. This demands the forthcoming scheduling algorithms to be “variance-aware”.

VM acquisition delay which is referred to as the time taken by a VM to deploy on a server and boot can further complicate the problem. As it is variable and service providers are uncertain about its value, it may affect the execution of a workflow.

Metaheuristic approaches have gained huge popularity in the foregoing decades as they can effectively tackle huge and complicated problems. Some of them are Ant Colony
Optimization (ACO), Genetic Algorithm (GA), Particle Swarm Optimization (PSO), League Championship Algorithm (LCA) and Bat algorithm. Intelligent Water Drops (IWD) algorithm is a latest metaheuristic technique, widely explored by researchers to achieve best solutions for complicated problems. The next section provides an overview of this contemporary algorithm.

1.5 Overview of Intelligent Water Drops Algorithm

IWD is a nature-inspired swarm-based optimization technique recommended by Shah-Hosseini in 2007 [35]. The IWD algorithm impersonates the action of water drops flowing in rivers. While moving from source to destination, water drops may affect their environment (riverbeds) and the environment may change the route of the water drops. The water drops are influenced by earth’s gravitational force which attracts everything directly towards the core of the earth. If there are no obstructions, water drops would ideally move straight away towards earth’s core. But, in fact due to the presence of different obstructions in the ideal path, the actual path is full of twists and turns and the destination is some lake or sea instead of earth’s core. A river is usually able to locate optimal path out of various paths while proceeding from source to destination. IWD algorithm is devised based on these actions of water drops in natural rivers to find an optimal path.

The IWD algorithm utilizes numerous IWDs to discover the optimal solution for a problem. Each IWD possesses two attributes namely velocity and soil. The value of both these properties may vary for each IWD as it moves in its environment. An environment signifies the problem that needs to be tackled. IWD algorithm represents the problem as a graph \((N, E)\) with node set \(N\) and edge set \(E\). Each edge of the graph is also associated with an initial soil. IWDs build solutions stepwise by moving from one node to other (visiting each node once) along the edges of the graph until an optimal solution with respect to cost or any other desired criteria is attained.

1.6 Thesis Organization

The rest of the thesis is structured as follows:

Chapter 2: Related Work
This chapter is aimed to highlight the recent developments in the discipline of scheduling for distributed computing systems. Scheduling techniques are categorized according to optimization methods, application models and objective functions. A comprehensive survey of scheduling strategies based upon optimization methods namely heuristics and metaheuristics is performed. Further, a comparative analysis of various scheduling approaches is presented focusing on methodology used, prevailing application models and optimization criteria. Finally, research problem is stated on the basis of inferences drawn from the existing work and accordingly objectives are outlined.

Chapter 3: Workflow Scheduling in Cloud

This chapter presents a workflow scheduling approach targeting to minimize execution time. The chapter begins with the description of application model and the formulation of workflow scheduling problem. The proposed approach based on IWD algorithm, a novel metaheuristic, is further discussed. Numerous IWDs begin with an entry task and seek a resource for the task. Each IWD gradually builds a solution by moving from one task to another satisfying the precedence constraints and finding suitable resource for every task. The resource for a task is chosen depending upon a probabilistic function. The probability function is modified here to enhance quality of solutions as well as to expedite the convergence of the algorithm. Finally, the recommended approach is compared with existing workflow scheduling strategies to validate its performance.

Chapter 4: Workflow Scheduling with Balanced Resource Utilization

This chapter presents an IWD based strategy, which not only reduces the execution time of workflows but also balances the load of VMs in clouds. The presented strategy prioritizes tasks using b-level ranking to speed up the convergence of best solution. Further, the probabilistic function for selecting VM for each task is modified to incorporate resource utilization of VM along with finish time achieved by the task on that VM. For performance evaluation, its results are compared with a number of renowned heuristic and meta-heuristic algorithms. Experimental results are based on three scientific workflows by varying their sizes.
Chapter 5: Cost Optimized Scheduling of Deadline Constrained Workflows

This chapter addresses concerns pertaining to execution cost while scheduling scientific workflows in clouds under deadline constraints and presents a cost-effective scheduling strategy. The proposed approach divides the user-defined deadline of the workflow into sub-deadline for each task in such a way that if each task is completed by that time, then it will guarantee that the workflow finishes ahead of the specified deadline. A swarm of IWDs collaborate in search of an optimal solution. Each IWD builds a solution by selecting a virtual machine for each task which can complete the task prior to its sub-deadline and ensures cost optimization. It considers the basic characteristics of IaaS Cloud such as elasticity, heterogeneity of infinite computing resources and pay-per-use model as well as the issues of acquisition delay and performance degradation of VMs. A comparative analysis with three existing scheduling strategies is performed to demonstrate its potential.

Chapter 6: Reliability Driven Workflow Scheduling under Deadline and Budget Constraints

This chapter investigates the problem of scheduling deadline and budget constrained scientific workflows in clouds. Here, the objectives are to minimize execution time and cost while maximizing the reliability. We propose a hybrid approach, which combines Intelligent Water Drops algorithm with Genetic Algorithm (IWD-GA) to resolve this multi-criteria optimization problem and provides a wide range of tradeoff solutions called as Pareto-optimal solutions to the users. It implies that an individual solution cannot optimize multiple objectives simultaneously, but the solutions will be minimizing one objective while compromising one or more objectives. It works in two stages. In the first stage, IWD algorithm is utilized to optimize cost within user specified deadline constraints. Referring to second stage, the schedule obtained from first stage is incorporated into the initial population of GA. The non-dominating sorting procedure incorporated into GA further allows it to achieve a broader range of trade-off solutions with respect to makespan, cost and reliability. The competence of the presented technique is examined by comparing it with two well-known metaheuristics: non-dominated sort genetic algorithm (NSGA-II) and hybrid particle swarm optimization (HPSO).
Chapter 7: Energy and Cost Aware Scheduling of Deadline Constrained Workflows

This chapter implements a workflow scheduling strategy focusing on minimization of makespan, execution cost and energy usage under deadlines specified by the user. To achieve our aim, we have proposed a hybrid of Intelligent Water Drops algorithm and Genetic Algorithm (IWD-GA) which provides non-dominated solutions to the user. The processors are supposed to be Dynamic Voltage Scaling (DVS) enabled which lets them adjust voltage and frequency and thus helps in lowering power consumption. The effectiveness of the suggested policy is shown by comparing it with two existing metaheuristics-based scheduling techniques.