USER CONTEXT LOGIC DEPENDENT INTEGRATED DYNAMIC WEB SERVICE SELECTION MODEL

A THESIS

Submitted by

N. BALAJI

in partial fulfilment for the award of the degree

of

DOCTOR OF PHILOSOPHY

DEPARTMENT OF COMPUTER SCIENCE

SCHOOL OF ENGINEERING AND TECHNOLOGY

PONDICHERRY UNIVERSITY

PUDUCHERRY 605 014

INDIA

DECEMBER 2015

PONDICHERRY UNIVERSITY
PUDUCHERRY 605 014

SCHOOL OF ENGINEERING AND TECHNOLOGY
DEPARTMENT OF COMPUTER SCIENCE

BONAFIDE CERTIFICATE

Certified that this thesis titled **"USER CONTEXT LOGIC DEPENDENT INTEGRATED DYNAMIC WEB SERVICE SELECTION MODEL"** is the bonafide work of **Mr. N. BALAJI** who carried out the research under my supervision. Certified further that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion of this or any other candidate.

Place : Puducherry
Date : 29-12-2015

Dr. M.S. SALEEM BASHA
RESEARCH SUPERVISOR
Department of Computer Science
School of Engineering and Technology
Pondicherry University
Puducherry – 605 014

ABSTRACT

Organizations are renovating their primary operations to the internet for more automation, universal view and effective commercial processes. The newest development in this modern system has a new approach, called Web Services. The web service selection operation in the traditional approaches, has been viewed as a textual representation based comparison and retrieval mechanism. The service request string has been compared with the WSDL description of the service and the services that matches the string has been returned to the requestor. The users are given the option to select the service that satisfies the QoS requirements from the set of discovered services.

In real, the users are not much aware of selecting the best service using the values of the QoS factors. This enables a technique to abstract the QoS factor selection from the novice user and the system should choose a best suitable service according to the user requirement. Moreover, the user service request query with the functional and non-functional attributes which are considered as static values and the possible combination of QoS attributes is not taken into consideration. Therefore, the existing system performs the service selection operation with the limited search space and so there exist a lot of chances to oversight the possible best combination of QoS parameters to satisfy both the requestor and the provider.

The ultimate aim of the research is to design and propose a User Context Logic Dependent Integrated Dynamic Web Service Selection Model using Self-Organization Genetic Algorithm based constraint Initialization. This was motivated by the inherent shortfalls of the existing web service selection models in the Web Service Computing environment. The First goal is defined so as to develop a web service clustering model based user context logic method, and the second goal is defined so as to design an effective Self-Organized Genetic Algorithm to optimize the proposed web service selection model based Combinatorial Optimization model for improving the efficiency of the service retrieval operation. Third objective is to assess the performance of the proposed selection framework using suitable Performance Assessment Factors. All the goals are measureable and of course proved with appropriate set of experiments.

This research proposed an effective integrated web service selection framework that segregates the user specified QoS parameters as positive and negative based on their impact on the service utilization. Further, the optimal combinations of the parameters are identified using the novel self-organizing genetic algorithm model with constraint initialization technique. To evaluate the performance of the proposed web service selection model, the experimental setup has been developed with 1000 web services obtained from QWS Dataset and 21 QoS factors associated with each service.

The performance has been compared with the most recent and the best working GA models identified in the study. The comparison of the outcomes and the promising experimental results demonstrate the impact of the proposed models in terms of precision, recall, f-measure, computation time, convergence rate, error rate, average convergence and convergence diversity. The assessment reveals that the proposed model shows the performance improvement with respect to the existing works CGA-Rand, CGA-NN and CGA-VV as 51.35%, 53.06% and 25.56% respectively. The constructive and encouraging results justify the significance and necessity of the proposed line of research and of course it may encourage further enhanced investigation in the identified area of research.

ACKNOWLEDGEMENTS

First of all, I thank the Almighty for blessing, wisdom and everything

At first, I would like to express my sincere gratitude to my supervisor, **Dr. M.S. SaleemBasha**, for being a great motivator, advisor and supporting me throughout my research with his patience and knowledge. One simply could not wish for a better or friendlier supervisor.

My special thanks to **Prof. R. Subramanian**, HOD & Dean, Department of Computer Science, School of Engineering and Technology, Pondicherry University, Puducherry, **Prof. P. Dhavachelvan**, Professor, Department of Computer Science and Engineering, Pondicherry University, **Dr. A.J. Ajay Vimal Raj.** Assistant Professor,Department of Electrical and Electronics Engineering, Pondicherry Engineering College,Puducherry and **Dr. T. Vengattaraman**, Assistant Professor, Department of Computer Science, Pondicherry University for their great support in my research and making quality publications.

I also wish to thank my fellow research scholars and all the teaching & nonteaching members of the Department of Computer Science, Pondicherry University for their cooperation and moral support throughout my research.

I would like to thank my Parents, **Mr. T. Natarajan** and **Mrs. N.Devaki**, Brother, Sisters and Friends.

[N. BALAJI]

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

XML	-	Extensible Markup Language
SOAP	-	Simple Object Access Protocol
WSDL	-	Web Service Description Language
UDDI	-	Universal Description Discovery Integration
HTTP	-	Hypertext Transport Protocol
FTP	-	File Transfer Protocol
SMTP	-	Simple Mail Transport Protocol
BPEL4WS	-	Business Process Execution Language for Web Service
API	-	Application Programming Interface
HTML	-	Hypertext Markup Language
DTD	-	Document Type Description
QoS	-	Quality of Service
OWL-S	-	Web Ontology Language - Service
EC	-	Evolutionary Computation
GA	-	Genetic Algorithm
CO	-	Combinatorial Optimization
iPSOA	-	improved Particle Swarm Optimization Algorithm
NUM.	-	Non Uniform Mutation
SLA	-	Service Level Agreement
WS-QoS	-	Web Service – Quality of Service
CA	-	Concept Analysis
WSES	-	Web Service Emergent System
MS-PSO	-	Multi Swarm Particle Swarm Optimization
PSO	-	Particle Swarm Optimization

MPSO	-	Modified Particle Swarm Optimization
EA	-	Enterprise Architecture
SOE	-	Services Oriented Enterprise
SOA	-	Services Oriented Architecture
SOC	-	Service Oriented Computing
NASCIO	-	National Association of State Chief Information Officers
COM	-	Context Oriented Model
QWS	-	Quality Web Service
P2P	-	Peer to Peer
WSim	-	WordNet Similarity
NGD	-	Normalized Google Distance
WSCM	-	Web Service Cluster Model
CLM	-	Concept Lattice Model
SWSSM	-	Semantic Web Service Selection Model
UCLM	-	User Context Logic Model
SSSM	-	Semantic Service Selection Model
SP	-	Service Provider
UBR	-	UDDI Business Registry
TSP	-	Travelling Salesman Problem
VRP	-	Vehicle Routing Problem
SA	-	Simulated Annealing
TS	-	Tabu Search
CI	-	Constraint Initialization
NN	-	Nearest Neighbour

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

A web service can be thought of as a web application which uses XML based standards for communicating with external systems for providing the necessary service to the user. The Web Service Architecture working group [19] defines a web service as a software system designed to support interoperable machine-to-machine interaction over a network and has an interface described in a machine-processable format (specifically WSDL) where other systems interact with the system in a manner prescribed by its description using SOAP messages. Web Services are *encapsulated, loosely coupled, self-describing, self-advertising, uniquely addressable, standards based and platform independent contracted functions* offered through standard protocols [40, 46].

The Web Services can be defined using different terminologies as follows:

Web Services are "independent, modular applications, accessible via the Web, that provide a set of functionalities to businesses or individuals" [31].

Web Services are “interface that describes a collection of operations that are network accessible through standardized XML messaging”.

The following are the characteristics of the web services [50].

- Web services are *reusable*; the services are modules developed to offer the potential reuse ability.
- Services share a formal *contract*; the service requires a formal contract that defines the terms of the service interaction and information exchange.
- Services are *loosely coupled*; services are developed to operate independent without the need of other service, though they can interact on demand.
- Services are *composable*; two or more services may be combined in order to provide a complete business application package. This enables business logic to be denoted at various levels of modularity and offer reusability.
- Services are *autonomous*; every service has been controlled within its defined limit described in its corresponding service description and is not dependent on other services for its invocation.

- Services are *stateless*; services need not to be maintained any state information as it inhibits the loosely coupled ability of the services.
- Services are *discoverable*; services should permit the WSDL of the corresponding service discoverable and understood by the user/service to use it.
-

The following Table 1.1 [31] describes some of the characteristics of web services and the technology that enables these functionalities to be achieved.

Table 1.1 Web Service Characteristics

Web Service Characteristics	**Underlying Technology**
Self-Describing	WSDL
Self-Advertising	UDDI
Uniquely Addressable	Internet Protocols
Standards Based	XML, SOAP, WSDL and UDDI
Platform Independent	XML and SOAP

1.2 WEB SERVICE COMPUTING MODEL

Web service computing model are different from traditional distributed computing models. Web service architectures provide a framework for creating and deploying loosely coupled applications. One of the consequences of loose coupling is that any entity that a web service may interact with other that may not exist at the point of time the web service is developed. New web services may be created dynamically just as new web pages are added to the web and web services should be able to discover and invoke such services without recompiling or changing any line of code.

The web service architecture [50] allows businesses to expose business assets as services. Standardizing interactions amongst services has the added advantage that any enterprise can out-source parts of its operation that it does not have expertise in. In addition, since the vision of web services enables web services to dynamically find new web services that it can interact with, enterprises can find new providers for the service relatively quickly. A specific application of this dynamism is in the e-procurement arena. For example, the average sourcing/procurement cycle in enterprises is of the order of 3-4 months. Of this time, about 50% of the time is spent in identifying the appropriate suppliers, about 20% of the time in handling the RFQ (request for quotes) process, and an additional 10% of the time is spent in negotiating the appropriate deal. The ability to dynamically find suppliers can translate to significant time savings, and therefore to lowering of costs. Essentially, the procurement and fulfilment business process are modelled as services, and a hub is the aggregation point for the services. In such an architecture, finding a new supplier is the same as finding the fulfilment service of the supplier at the hub.

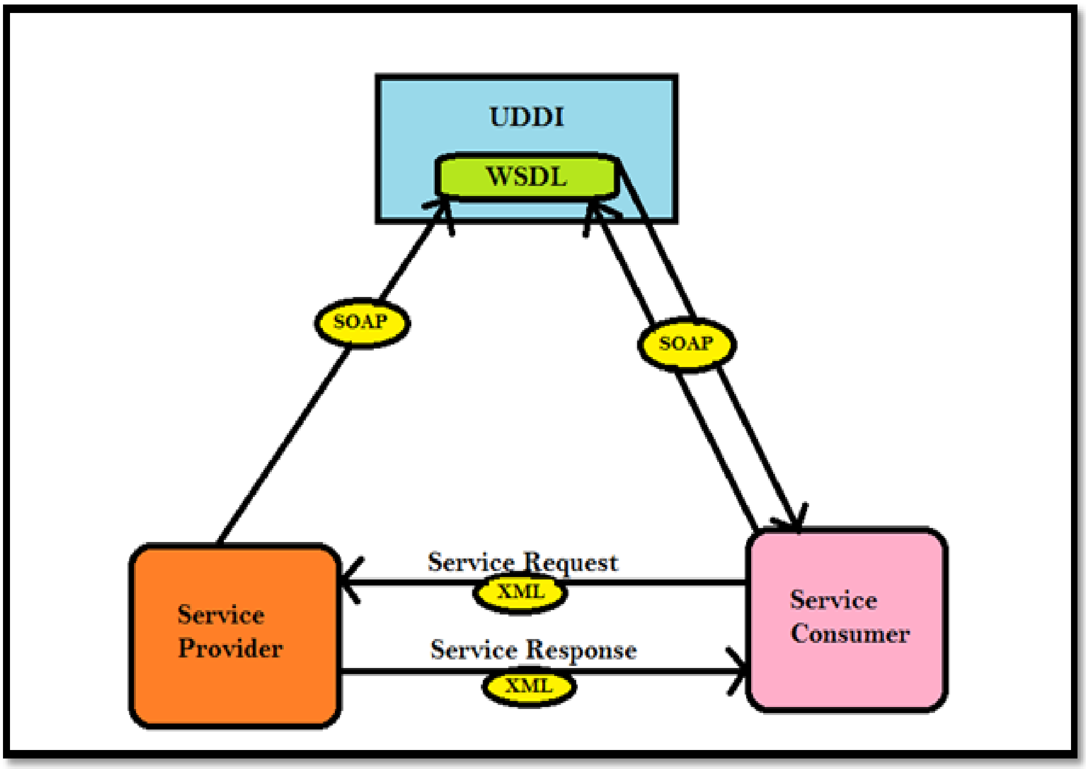

Figure 1.1 Web Service Computing Model

Web services are a standardized way of establishing communication between two web-based applications by using XML, SOAP, WSDL, UDDI, and open standards over an Internet protocol backbone. Where XML is used to encode the data in the form of SOAP message, SOAP is used to exchange information over HTTP, WSDL is used to describe the capabilities of web services, and UDDI is used to provide the list of service provider details, as shown in Figure 1.1, [50].

In the real-time scenario, if a service consumer wants to use some sort of web services then he/she must know the service provider. If a service provider validates a service consumer, it will provide the WSDL file directly and then the service consumers create a XML message to request the required service in the form of a SOAP message and the service provider returns a service response.

On the other hand, if the service consumer is not aware of the service provider, he/she will visit UDDI and search for the required service. The UDDI returns the list of service providers offering that particular service. Then, by choosing one service provider, the service consumer again generates an XML message to request the required service in the form of a SOAP message, as specified in the WSDL file of the service provider, and the service provider returns a service response.

1.2.1 Roles in Web Service Architecture

Based on the nature of the responsibilities, components in the web service architecture are divided into three roles, namely service provider, service consumer and registry [50].

Service Provider: In the view of the business operation, the providers are the owners of the service. From the view of architectural design, the providers are location from where the service implementation can be accessed by the consumers. The responsibilities of the service provider are similar to the role of server in the typical client server architecture.

The service provider is also refers to identify the individual or organization liable for offering the service [50]. In order to differentiate the role of service from the service's real provider, the following terms can be used:

- Service provider entity: the individual or organization publishing the service
- Service provider agent: the invokable web service itself, acting as an agent on behalf of its provider

Service requestor: In the view of the business operation, the service requesters / consumers are the business that entails the specific operation to be satisfied. From the view of architectural design, the requestors are individual or application that requires to access or invoke the service with specific functionality. The role of the service requestor can be played by a browser operated by a person or an application without the intervention of a user.

Similar to the service provider, the responsibilities of the service requestor are similar to the client system in the classic client server architecture. The term requestor can refer to another web service or sometime the web service provider who require another service to accomplish a task successfully [50]. Consequently, the following prolonged terms are used:

- Service requestor entity: the individual or organization requesting the web service
- Service requestor agent: the web service itself that act as an agent on behalf of its provider

Service registry: A Service Registry is a network-based directory that contains all the published services. It is a component that receives and stores service information and contracts from service providers and provides those to the interested service consumers. Service requestors find services and obtain binding information that presented in the service descriptions for services during development for static binding or during execution for dynamic binding. A service repository can also be viewed as a service broker who manages the published service providers' information such as the services, its functional and non-functional description, etc. This information includes business white page information such as service name, functionality description, and contract information. It also incorporates the green pages information like service policies, business processes, and port bindings which are essential to access. It also offers the yellow pages information for intelligent search capabilities and business classification.

These repositories allow service to:

- Find the latest versions of suitable service descriptions
- Discover new web services that meet user required criteria

The service registry can classified into two based on their authorization capabilities, namely private and public registries. The public registries receive registrations from any individual or organizations with complete details. Once registered, individual or organizations acting as a service provider can publish their services [50]. The private registries can be employed within organization to offer a centralized repository for descriptions of all services that the organization develops, leases, or purchases.

1.2.2 Web Service Operations

There are three operations involved to access the service in the web service environment. They are: publication of service descriptions, discovering or finding the required service description, and binding or invoking the discovered services. These operations can be described as follows:

- *Publishing the service:* A service description required to be published on the registry so that the service requestor can find it. The description consists of the details of the provider, functional and non-function properties of the service.
- *Discovery of the service:* In the find operation, the service requestor queries the service registry for the service required with specific functional attributes. The discovery process can be performed in two different stages for the requestor: one at the design time to access service's interface description for application implementation and other, at the real-time to identify the service binding information for invocation.
- *Binding the service:* In the bind operation, the service requestor invokes or initiates an interaction with the provider at runtime using the binding details in the service description, obtained at the discovery operation, to locate, contact and invoke the service.

1.2.3 Web Service Standards

The key standards concerned with web service environment are WSDL [10], UDDI [38], SOAP [8] and XML [44] are shown in the Figure 1.2.

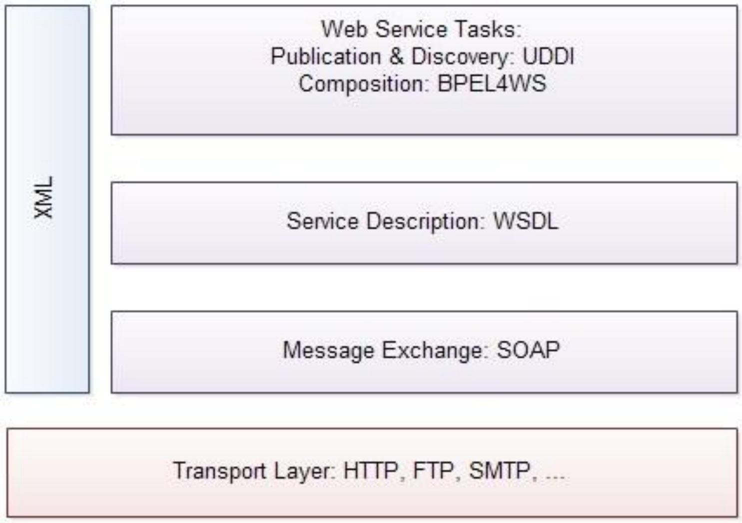

Figure 1.2 Web Service Standards

1.2.3.1 The Web Services Description Language (WSDL)

In order to invoke a web service by the requestor, the programmatic interface description of the corresponding web service must be accurately defined. The web service description is an XML grammar for specifying the attributes of a service such as what it function, where it is located, the values of its non-functional attributes and how it can be invoked. A WSDL document defines services as collections of network endpoints, or ports. In WSDL [10], the abstract definition of endpoints and messages is separated from their concrete network deployment or data format bindings.

1.2.3.2 The Simple Object Access Protocol (SOAP)

SOAP [8] is a standard for sending messages and making remote procedure calls over the Internet. It is independent of the programming language, object model, operating system and platform. It uses HTTP as the transport protocol and XML for data encoding. However, other transport protocols may also be used such as FTP, SMTP or even raw TCP/IP sockets. SOAP defines two types of messages, Request and Response, to allow service requesters to request a remote procedure and to allow service providers to respond to such requests.

1.2.3.3 Universal Description, Discovery, Integration (UDDI)

UDDI [30] defines a common means to publish information about businesses and services. It can be used to check whether a given partner offers a particular Web Service, to find companies in a given industry with a given type of service and to locate information about how a partner or intended partner has exposed a Web Service in order to learn the technical details required to interact with that service. The UDDI specifications consist of an XML schema for SOAP messages and a description of the UDDI APIs specification. The UDDI APIs contain messages for interacting with UDDI registries. Inquiry APIs are provided to locate businesses, services, bindings or tModels. Publishing APIs are included for creating and deleting UDDI data in the registry.

1.2.3.4 XML – Extensible Markup Language

Extensible Markup Language (XML) [44], which has become the common language of the computing business, is a tag-oriented language that looks superficially like HTML, but its purpose is different. HTML describes the way how a document should look like when displayed in a browser. Whereas, XML can describe the meaning of the given data, independent of the way that it is displayed. Briefly, XML is to remove the ambiguity from data.

Adding these XML style tags to the data makes it easy to parse and use. An additional feature of XML is the existence of a description of what the tag structure must look like in a document for it to be valid. The original description was called the Document Type Description (DTD), but it is fast being made obsolete by the XML schema. They both serve the same purpose, but the XML schema is more powerful and allows for more precise descriptions.

1.2.3.5 Hypertext Transport Protocol (HTTP)

Hypertext Transport Protocol (HTTP) is the workhorse of the Web. It is managed by the W3C as well [50]. Its current version is 1.1, and all activity has ceased on it because it is considered complete and stable. The purpose of HTTP is to provide a protocol to move requests and responses between clients and servers. HTTP carries any information that is placed in it from point A to point B without regard for its data type. As a result, it is a popular way to transport SOAP messages between clients and Web Services.

1.3 WEB SERVICE SELECTION

The web service requestor creates a request message with necessary information to discover the suitable service from the registry. Nowadays, the web service providers as well as consumers are more concerned with the QoS guaranteed web services. Thus, it is necessary to select a service that effectively satisfies the consumer requirements from the collection of service discovered from the registry. From the consumer's perspective, QoS based web service selection is a multi-criteria decision mechanism that requires knowledge about the service and its QoS descriptions [18]. Though, most of service consumers are not knowledgeable to obtain the best selection of web service based on its described QoS characteristics. Therefore, they just rely on the QoS promised by the service provider; yet most of the service provider do not assure and guarantee the QoS values offered. Thus, selecting the suitable web service based on the user required QoS factors become a prominent requirement and also QoS factors are used to illustrate the web services' overall significance.

In general, web service selection can be performed in two ways as follows:

- *Selection based on the functional Matchmaking:* In this approach, the services are selected purely based on the comparison of the functional attributes of the service with the user requirement.
- *Selection based on Quality of Service (QoS):* In this approach, the services are selected based on the values of different QoS factors as required by the user. The Service consumer may require the different range of values of QoS [18] factors offered by the corresponding service.

1.4 PERFORMANCE IMPROVEMENT IN WEB SERVICES SELECTION

In the traditional approaches, the web service selection operation has been viewed as a textual representation based comparison and retrieval mechanism. In these systems, the user request is obtained as the string that describes the functional requirement of the service. The string has been compared with the WSDL description of the service and the services that matches the string returned to the requestor. The users are given the option to select the service that satisfies the QoS requirements from the set of discovered services. Of course, the users are not much aware of selecting the best service using the values of the QoS factors. This enables a technique to abstract the QoS factor selection from the novice user and the system should choose better suitable service according to the user requirement.

Moreover, the user service request query with the functional and non-functional attributes are considered as static values so the possible combination of QoS attributes are not taken into consideration. Therefore, the existing system performs the service selection operation with the limited search space and there exist a lot of chances to oversight the best possible combination of QoS parameters to satisfy both the requestor and the provider.

1.4.1 Issues and Challenges in Web Services Clustering

The related research work presented in the previous section clearly indicates the factors that affect the performance of the web service selection [52]. Thearea of web service selection is developing as a conspicuous technology both in technical and business verticals and numerous research works are being developed in this domain, yet there are challenges to be addressed. The following discussion clearly identifies the challenges in this domain with respect to the related works presented in the previous section.

- The web service developer or provider must use only the technical words specifying the functionality, in other words, the semantic information is not considered.
- Lattice based techniques for discovering web services with composite backups consider the operation similarity, but the QoS constraints are not satisfied.
- Trust-QoS based semantic service selection method minimizes the selection duration but the service does not execute any meaningful operation.
- A novel architecture for web services composition extend the standard web service composition model to suit exclusively the web services composition, which introduces too many new interactions thereby increasing the time.
- In selection based user requirements, efficient web service selection is guaranteed with web service analysis but testing is not considered.
- Semantic service selection based on user preference cluster is a model which increases the service selection recall ratio with the cost of increasing the average response time.

- The semantics based automated service selection uses a semantic based service categorization and semantic enhancement of the service request, but the QoS parameters and the automatic service selection are not considered.
- In web services selection based on semantics and clustering, the services are discovered semantically, but more time is required during the pre-processing stage.
- The semantic service search engine identifies many semantic services during search, but the prototype does not support even pure WSDL and RESTful services.
- Web Services Selection based on multi-criteria approach can discover the relevant services in less time with some QoS parameters [18,28,35], but the test set taken is very small to evaluate the proposed approach and also more QoS parameters can be considered.
- In the existing web service selection system, the semantic relevance among the services is not considered, but the quality factors are considered.
- In web service selection framework using OWL-S advertisements, the system has been evaluated to perform with high precision and recall

against the typical query processing system, but the quality factors of the web services are not taken into consideration.

The challenges faced by the web services selection listed above are mostly based on the factors that affect the performance of the web service selection. The attributes of the challenges can be identified as follows:

- Difficulty inrepresenting functionality
- QoS Constraints
- Increase in response time
- Increase in the pre-processing time
- Lack of support for certain services
- Insufficient test data

To summarize, any research in the area of Web Services Selection in future needs to address the above challenges with the attributes mentioned. Thus in this research work, concept lattices fordiscovering web services are used in order to address those issues.

1.4.2 Need for Improvement in Web Services Clustering

There is a clear need for improvement as the existing approach doesn't take semantic information into account while clustering web services. The existing web service selection system does not consider the semantic relevance among the services, only the quality factors are considered. This is a huge drawback. The current situation also calls forimprovement in representing functionality, response time, pre-processing time etc., Lattice based techniques for discovering web services with composite backups consider the operation similarity, but the QoS constraints are not satisfied. Consideringthe semantic information and the service lattice with QoS constraint is the answer to this problem.

1.5 EVOLUTINARY COMPUTATION IN COMBINATORIAL OPTIMIZATION

Evolutionary Computation (EC) is one of the prominent optimization approaches for combinatorial problem with large search space. Among varieties of evolutionary computation techniques, Genetic Algorithm (GA) is the highly preferred and justified for its effectiveness over several hard combinatorial problems. The GA belongs to the family of evolutionary computation models inspired by biological evolution, natural selection and survival of the fittest in living organisms. In 1975, John Holland [1975] introduced the concept of GA in his book "Adaptation in Natural and Artificial Systems". GA is a panacea for resolving all optimization problems and GA has two important features: first, it is a stochastic algorithm where selection and reproduction are performed randomly; second, GA always works with a population of solution, which offers with the benefit of assortment and robustness to the technique. The need of GA is to find the best solution in the large search space which is the collection of all feasible solutions among with desired solution.

GA has been proved for its significance in optimization of various combinatorial optimization problems like Vehicle routing [23], Drilling of printed circuit boards [57], Overhauling gas turbine engines [24], X-Ray crystallography [41], computer wiring [57], sequencing and scheduling [48] and the order-picking problem in warehouses [11, 37, 43, 45].

1.6 MOTIVATIONS

The motivation of the work reported in this thesis has three folds; there does not exist a peculiar Framework for User Context Logic Dependent to Bridge the Gap between the Service Discovery and the Service Selection; to portray the QoS based web service discovery/selection operation as a combinatorial optimization problem; importance of effective clustering for web services and the significance of genetic algorithm in effectively solving the complex combinatorial problem.

Firstly, the user context base service operation is the recent and poplar work that can identify the service based on the user requirement on the QoS of the service searched. The typical scenario of performing the web service selection process as a textual comparison and user controlled QoS selection restricts the size of the search space for the best suitable service. And also, selecting the service with all the possible combination of QoS factors can enhance the possibility of obtaining the best solution that can satisfy the requirements of both the service requestor and the service provider. This motivates to formulate the web service selection process as a combinatorial problem with QoS factors as a permutation element.

Secondly, there is a clear need for improvement as the existing approach doesn’t take semantic information into account while clustering web services. The existing web service selection system does not consider the semantic relevance among the services, only the quality factors are considered. This is a huge drawback. The current situation also calls for improvement in representing functionality, response time, pre-processing time etc., Lattice based techniques for discovering web services with composite backups consider the operation similarity, but the QoS constraints are not satisfied. Considering the semantic information and the service lattice with QoS constraint is the answer to this problem.

Finally, though a collection of permutation coded optimization algorithm exists in the literature that can effectively solve the combinatorial problems. Thus, a recent and best working optimization problem can find the best optimal service from the collection of services discovered by considering it as a combinatorial problem. It is also necessary to modify the existing algorithm to manage with the QoS factor combination methodology. This motivates to propose an effective permutation based combinatorial optimization algorithm to solve the web service selection based combinatorial problem effectively.

1.7 OUR CONTRIBUTIONS

Based on the discussions so far, the web service selection process need to be enhanced using the combination of the QoS factors and also a suitable combinatorial optimization. It also has to be developed to identifyeffectively the optimal solution. In this perspective, the aim of this research is to propose and develop an effective optimal web service selection framework and optimize its services using Self-Organizing Genetic Algorithm based constraint

Initialization. Further the experimental evaluation schemes are designed to justify the significances of the proposed model with respect to the existing best working models of GA.

The ultimate aim of the research is to Design and Propose a User Context Logic Dependent Integrated Dynamic Web Service Selection Modelusing Self-Organization Genetic Algorithm based constraint Initialization.

For achieving the above, this research aim to contribute its outcome as three goals as follows:

- To design a Web Service Clustering Model that enables User Context Logic Dependent Integrated Dynamic Web Service Selection Framework.
- To design and develop a Self-Organized Genetic Algorithm tooptimize the proposed CO model.
- To evaluate and validate the performance of the proposed selection framework using suitable Performance Assessment Factors.

In order to effectively carry out the goals set, the research has been segregated into contributions as listed below:

- To Formulate a User Context Logic Dependent Framework to Bridge the Gap between the Service Selected with and without the QoS considerations.
- To Design the Web Service Selection as Combinatorial Optimization Problem and Methodology to Optimize using GA.
- To Develop and Propose an Optimal Web Service Selection Framework with Constraint Initialization for GA.
- To Design and Develop the Testbed using Benchmark Instances from QWS Datasets and Performance Factors.
- To Evaluate the Performance of the Proposed Web Service Selection Framework using Genetic Algorithm based Performance Factors.

1.8 ORGANIZATION OF THE THESIS

A lot of effort has gone into the organisation of the thesis so that arguments, hypotheses, illustrations and clarifications are coherent and lucid.

The underlying view is that the work should be an important and strong link in the chain of research in the area concerned.

- Chapter 1 provides a broad overview of the Web Service Computing Model, Web Service Selection Models and also the improvements required for the Integrated Dynamic Service Selection.
- Chapter 2details the basic background information and the motivation for the research in the area of web services and concept lattices.
- Chapter 3 reviews Goals and Research Methodology, how the Web Service Computing Model can be used to enrich Conceptual Stack Mapping. The chapter introduces the fundamental ideas behind the web service similarity measures.
- Chapter 4 describes Web Service Clustering Model that enables User Context Logic Dependent Integrated Dynamic Web Service Discovery and Selection Framework, and briefly outlines their adoption and the overall system design.
- Chapter 5 elucidates the experimentation and evaluation of the Framework proposed for the Dynamic Web Service Selection Framework.

- Chapter 6 presents the Web Service Selection as a Combinatorial Problem and corresponding Genetic Algorithm to solve it and briefly outlines the overall system design.
- Chapter 7derives the experimentation and evaluation of the Genetic Algorithm Model for Integrated web service operation Framework.
- Finally, Chapter 8 provides the concluding remarks of the work presented in this thesis and the future enhancements of the proposed line of research.

CHAPTER 2

LITERATURE STUDY

2.1 INTRODUCTION

In recent times, several service providers create the service with the same functionality and consequently Quality of Service (QoS) of the service become the deciding factor for the optimal service selection as per the user requirement. QoS is referred to a set of non-functional performance factors like service throughput, price, availability, reputation and reliability. The current UDDI technology supports selection of service based on its functional attributes and it is necessary to enhance its ability to select the service with respect to the QoS attributes [40]. Thus, the web service selection based on the QoS attributes of the service should be considered as a vital,complex problem and efficient problem solving techniques should be imposed to identify the best optimal solution for the same. In this perspective, this chapter is devoted to study the various literature works on the QoS aware web service selection methodologies and also on the different approaches devised using the bio-inspired methodology to effectively perform the web service selection in the complex web service environment. This review helps to identify the research gap in the recent works on considering the web service selection as a complex problem and also the effective methodologies to solve it affluently.

Morevover, several works have been proposed for web service clustering in order to facilitate service selection and discovery. The major challenges associated with the discovery of web services are to find an appropriate web service description and to reduce the time taken to discover the web service. These challenges can be addressed by using the semantic information and clustering of web service. There are numerous methods in the literature for clustering web services using semantic information. A quick overview of some of the works can be obtained from the following papers; a selection of works that are classified according to their adapted techniques is discussed.

2.2 USER CONTEXT AND QoS AWARE DYNAMIC WEB SERVICE SELECTION

In the web service environment with more number of functionally similar services, the QoS based web service selection is the critical criteria to satisfy the service consumers indeed. In such scenario, QoS aware web service selection from the list of candidate web services discovered with the similar functionality, as requested by the user query, based on their QoS descriptions plays a significant role. The QoS properties of the service are described as a QoS information in the UDDI, along with the service's functional description, for effective matchmaking and retrieval. These QoS information have substantial influence on expectancy of theconsumers and the experience of using a web service [19]. Hence, it can be utilized as a key measure to discover and select the web service from a large set of similar functionality candidate services. Though, it is mandatory that the QoS based selection should be performed only after the functional matching with the users' service request.

A QoS broker based architecture for web service selection and ranking mechanism has been proposed in [18]. The QoS broker is responsible for the selection and ranking of functionally similar web services. This model selects the web services based on the prospective levels of satisfaction of requester's QoS constraints and context preferences. The selection and ranking of the web service are performed based on the QoS values of the corresponding service and in this model, the selection problem has not been formulated and the broker model is centralized, so it is prone to single point of failure problem.

An improved Particle Swarm Optimization Algorithm (iPSOA) had been proposed to deal with the delinquencies over the QoS aware service selection and discovery in the composite web service environment [51]. Authors have redefined the factors such as speed, position operations to make the algorithm to suit for QoS aware selection. Moreover, a Non Uniform

Mutation (NUM) technique is enforced to the global best solution to improve the population diversity to overcome the problem of premature convergence.But, the authors did not define and use many of the QoS factors associated with the web services, this makes the proposed work subject to justification for the factors not considered in the work.

A QoS mediator agent based web service selection technique has been proposed in [19], where QoS consultant acts as a middleware between the service consumer and the provider. In the proposed technique, user context preferences on QoS factors is considered inservice selection and it also helps to select pertinent service according to the users requirement in an automated fashion. A three-way satisfaction [22] for web service selection used for the selection of better service from the large community of similar web services. In this model,the master web service calculates the SCORE, based on the QoS values of the service, of another decedent web service based on capacity, execution time and availability. The proposed approach helps to solve the problem of selection web service, to a certain extent, within a community of similar domain.The test bed developed has not been precise and performance evaluation factors are not exactly infer the effectiveness of the proposed work with respect to the web service selection operation.

In [42] as novel End-to-end Quality-of-Service algorithm has been proposed to model the service selection using associative classification. This approach classified the candidate web services to a different level / class of the QoS values. They classified the services within the each class, with respect to their distances from the user service request for the required QoS values. And the approach used classical data mining algorithms to identify the most suitable service based on the user requested QoS criteria [35]. The authors did not normalize the values of the different QoS factors, since different factors would be represented in different scales. This enables the work not effective in the calculation of the score for the service using which the optimal solution is being identified. And also the combination QoS factors are not considered for the web service selection method. In other words, the authors considered only fourQoS factors to identify the optimal service. In the real world, a collection of QoS factors has to be considered in combination to identify the optimal service for the user.

In the work [40] the authors have presented the web service discovery and selection methodology in which the non-functional attributes of the service are represented using the ontology. Authors built a detailed ontology, using the domain experts, to enhance the web service selection process in the interpretation of reusability and design. Using the concept of QoS and ontology, a new framework for service selection using dimensionless matrix is developed to deliver a desired ranking and selection of service based on the user context QoS preferences [17]. Though, the problem formulation is not precise, which makes the framework proposed is not suitable for the real, large and complex web service computation environment. And also the benchmark instance used for experimentations are not standard such that the analyses are subject to justification in a standard environment.

The author [46] proposed an agent based framework for web service selection that satisfy the user context critical QoS factors preferences. The authors also devised a model to extend the UDDI to accommodate the QoS information using user SLA which is specified using the XML [13]. In [46] discussed an effective WS-QoS architecture that allows QoS aware service specifications and also the formulated a broker based service selection model for effectual QoS aware service selection [18, 56]. The authors did not define and use many of the QoS factors associated with the web services, this makes the proposed work subject to justification for factors not considered in the work.

The author [2] proposed a method to cluster web services based on lattices using Concept Analysis (CA). A lattice structure for clustering web services was used and it was supported with candidate backup services to ensure continuous functionality. String edit distance was used to find the similarity between the service operations and thus the service lattice was built. This measure is used in keyword based clustering technique, but this measure cannot be used for semantic clustering of services.

The aouthor [52] proposed a method which aims to cluster web services before web service discovery based on OWL-S language. The web service similarity discussed here is based on an accurate concept semantic similarity of the domain ontology. Concept semantic similarity is considered. The influencing factors of concept semantic similarity based on the hierarchy are mainly dependent on the depth of the hierarchy. The depth of the hierarchy has been calculated based on the ontology hierarchy. It is more exhaustive in the concept classification.

The author [1] proposed an approach to discover web services with its candidate substitutes. Concept lattices are used to classify web services, depending on the similarity estimated between the operations. Lattices are used to discover web services along with candidate backups to ensure the continuous functionality in case of composite web applications. Though lattices are generated based on their operation similarity but QoS is not considered.

2.3 BIO-INSPIRED ALGORITHM FOR WEB SERVICE SELECTION

In this section, the different bio-inspired algorithms proposed for effective web service selectionhave been studied and their corresponding drawbacks are identified.

The author [55] proposed a bio-inspired system for efficient web service selection and composition in the complex web service environment. In the work, authors introduced a novel concept called bio-entity, as an autonomous agent, is designed to represent a web service and also constructed a Web Service Emergent System (WSES) to find the desirable characteristics of web services using the perceptions of from self-organizing, cooperating, and compositing of services. The main significances of the WSES model are the mechanism of systemic intelligence and the automatic service selection & management though the web service emergence.But, the test bed developed has not been precise and performance evaluation factors are not exactly infer the effectiveness of the proposed work with respect to the web service selection operation.

In the work [12] authors presents a new technique for web service selection, for composition, inspired by the behaviour of bees. The model proposed uses an enhanced planning graph model combined with a bee-inspired optimization algorithm to find the service selection solution that satisfies the requestor constraints. Authors used ontological concepts and weights associated to userepreferencesto represent the functional and non-functional requirements of the request respectively. The proposed model is justified that it optimizes the service selection process without considering the complete problem search space and avoids the local optimum stagnancy problem. But, the algorithm formulated for service selection operation has not been altered to improve the selection operation efficiently.

In [47] proposed a web service selection model using Multi Swarm Particle Swarm Optimization (MS-PSO) algorithm inspired by the animal collective behaviour, the movement and the intelligence of the swarm. The proposed modelcontributed an effective and optimal web service selection technique using the Quality of Service (QoS) of the services. Authors claims that the proposed algorithm can handle a variety of complex real world optimization problems and proved the performance of the proposal using the existing GA and PSO techniques. Authors justified the performance of the proposed work using a set of five benchmark functions which is not supportive to consent the significance of the proposed work. Moreover, the bio-inspired algorithm used for web service selection mechanism has not been modified to suit and optimize the web service selection operation.

An effective middleware platform for QoS-aware web service selection based on Ant Colony System is proposed by [53]. The proposed approach obtains several task execution paths in a model of web services composition and also proposed genetic algorithm to derive the best service. The primary significance of the work is that the analyses is based on the measurement of the quality of web services and makes user satisfactory degree as the evaluation indicator, which is suitable to a dynamic environment. Though, the proposed model doesnot suit for large sized service selection scenario and the service retrieval evaluation factors are not used to assess the effectiveness of the proposal.

In the work [14] authors proposed a Modified Particle Swarm Optimization (MPSO) algorithm for Search Based Quality of Service (QoS) selection and ranking predictions. The model uses the records of an active user to elude the conflicts that occurs in the existing approaches and also avoids the irrelevant solutions which can be generated in PSO. The mechanism claimed to achieve high efficiency and reliability, albeit the QoSvalues of services are incomplete in nature, using the collection of information from the neighbour user's results. The QoS dataset used for the service selection are irrelevant and cannot be compared with real time web service environment still holds as a problem. The combination QoS factors is not considered for web service selection method. In other words, authors considered only two or more QoS factors to identify the optimal service. In real world, a collection of QoS factors has to be considered in combination to identify the optimal service for the user. Authors did not define and use many of the QoS factors associated with the web services, this makes the proposed work subject to justification for factors not considered in the work.

A new heuristic method called “Bees Algorithm” inspired from honey bees’ behaviour to optimize the discovery and selection of appropriate web services has been proposed to meet the customer requirements, in least time and taking intoaccount the QoS properties [7]. The proposed algorithm has six major steps such as random exploration, domain similaritymeasure, sorting of suitable registry, selection of suitable registry, exploration of “elite” registry’s neighbourhood, stop on maximal similarity, selection of the service providing the required QoS. The cooperative behavior of employee bee, onhooker bee and scout bee results on high scalability and promising results in terms of exploring optimal service search space. The proposed bees algorithm changes only one service parameter per iteration, takes large amount of iteration, consequently higher response time and also the convergence rate is not promising may lead to failure of service response often.

In [54] authors proposed a bio-inspired algorithm for web service discovery and selection model in pervasive environments. The work exploits a social behaviour and ant inspired service request routing mechanism, which views request messages as artificial ants, and utilizes pheromone and social utility as the routing hints that direct request messages to nodes owning better optimal service for the request. The difficulties in the service selection problem has not been formulated precisely and the fuzzy based service operations prone to overburden the throughput of the overall performance of the system. The test bed developed has not been precise and performance evaluation factors are not exactly infer the effectiveness of the proposed work with respect to the web service selection operation. And also the benchmark instance used for experimentations are not standard such that the analyses are subject to justification in standard environment.

In [28] authors proposed a QoS based service selection methodology that adopts genetic algorithm to identify the optimal web service whichsatisfy the user's QoS requirements. Another methodusinf the semantic web service concept is also proposed to address and define non-functional quantities for selection of web services, such as WSMO [21], OWL-S [49], and SAWSDL [36]. Authors did not normalize the values of the different QoS factors, since different factors would be represented in different scales. This enable the work not effective in the calculation of the score for the service using which the optimal solution is being identified. The bio-inspired algorithm used for web service selection mechanism has not been modified to suit and optimize the web service selection operation.

2.4 SUMMARY

In this chapter, the litertature survey has been performed in two folds. The first is in the view of QoS aware web service selection and the later is bio-inspired algorithm based service selection optimization.

In the perspective of QoS aware web service selection, this model of selection is critical since the consumer has to prefer better service from the collection of sevices of same functionality. The QoS of the service that actually provide the sophitication to the consumer to enable better utilization of the service. But, none of the existing technique the QoS based web service selection has not been correctly formulated as a complex problem as it should be. Though, authors proposed different techniques for effective selection, improper problem formulation and small size experimental analyses makes the existing work not sultable for real time large sized web service environment.

The similarity measure considered so far for clustering lattices is based on the string edit distance only. This poses the problem of not considering the semantic information and clustering of services and lattice based clustering cannot be neglected as they have numerous advantages. There are many semantic web service discovery frameworks available in the literature. Butmostof the research works consider keyword match making and only a few consider the semantic information during web service discovery. Furthermore, the quality parameters are neglected during the services discovery.

In the perspective of bio-inspired algorithm based web service selection optimization, the exisiting algorithms have not been modified to suit and optimize the web service selection operation precisely. And also this selection algorithm should be considered as the combinatorial problem since every QoS factor plays vital role in the selection of the optimal service.Moreover, The test bed developed has not been accurate, the benchamark instances and the performance evaluation factors are not exactly infer the effectiveness of the web service selection operation.

CHAPTER 3

GOALS AND RESEARCH METHODOLOGY

3.1 INTRODUCTION

"Web Service based Systems promise to bring significant business value out of existing IT assets through increased operational efficencies, optimized business processes, and the ability to adapt and change quickly,". Platform and language independent and flexible access to information is indeed a complex task and resource intensive too. Web Service based Systems simplify this through standard protocols which treat all platforms equally. Through such systems, it is possible to offer data Services to a wide variety of business partners and the requests can originate from anywhere. The terms, Enterprise Architecture (EA), Services Oriented Enterprise (SOE), Service-Oriented Architecture (SOA) and Service Oriented Computing (SOC) are being exposed in connection with the Web Service based Systems to an ever wider and more influential audience.

Web Service based Systems promise to be a significant innovation that will provide the ability to pick and choose business and technology services, and will allow the trade out of Services based on organizational re-design, new strategic intent, legislative requirements, or business process modifications (USA National Association of State Chief Information Officers (NASCIO)).

From its inception Web Service based Systems have been a lightning rod for dissension among enterprise architects, solution architects and application architects [31]. Enterprise architects view such systems as a business initiative that should be guiding how information technology assets receive investment and how they relate to the business' goals and mission. Solution architects view such systems as a means to deliver solutions faster using the tenets of loose-coupling and finer-grained Services, which enable faster construction. Finally, application architects see Web Service based Systems as an infrastructure to deliver applications based on Service interfaces.

Having seen the wider adaptability and acceptance of Web Service based Systems by enterprises, it is widely accepted that they offer the following advantages over traditional approaches to distributed computing.

- Web Service based Systems offer business Services across the platforms and thereby the application architects can conveniently choose a particular service for any kind of platforms on which the system need to be developed.
- Location independence is considered to be a major advantage through which the application developer can make use of the Web Services transported from anyplace to everywhere.

- That the Services need not be at a particular system or particular network is the greatest benefit of using Web Services for application development. Such benefit really brings out the developer from the pressure of locating all necessary Services from a single location and use.
- Web Service Systems are built on a complete loosely coupled approach which brings in high level of modularity and the developer need not re-architect the entire application for a minor change in a particular section of the application.
- Web Service based Systems also offer the advantage of Authentication and Authorization support at every level of the application building, which makes the entire system highly secured.
- The search and connectivity to other Services is dynamic, which enables the system developer find the Services which are just fit for the applications being developed and the connectivity which extends the usage of a particular Service by getting it coupled with another Service.

The field of Web Service based Systems and Service Oriented Architecture are the current topic of research and development in Computer Science, which is especially oriented to the needs of gaining and processing information in large-scale distributed systems such as the Internet. It is expected that Web Services will be employed to a higherextent for the realization of new kinds of application systems.

Service Oriented Architecture(SOA) is an architectural style for building Web Service based Systems that usesWeb Services available in a network such as the web. It promotes loose coupling between software components so that they can be reused. A Web Service is an implementation of a well-defined business functionality and such a service can then be consumed by clients in different applications or business processes [5]. Web Services are:

- Software components with well-defined interfaces are implementation-independent. An important aspect of SOA is the separation of the Service interface (the what) from its implementation (the how). Such Services are consumed by clients that are not concerned with how these Services will execute their requests.
- Self-contained pieces of code which can perform predetermined tasks and loosely coupled, which offer the benefit of independence.
- Web Services can be dynamically discovered from the Service Provider and if need be, the Web Services can be combined and the Composite Services can be built from the aggregates of other Services.

The Universal Description, Discovery, and Integration (UDDI) specification defines a way to publish and discover information about Web Services [38]. UDDI is a public registry designed to house information about businesses and their services in a structured way. Through UDDI, one can publish and discover information about a business and its Web Services. This data can be classified using standard taxonomies so that information can be found based on categorization. Most importantly, UDDI contains information about the technical interfaces of a business's Service.

Web Service based Systems have stretched the Universal Description, Discovery and Integration (UDDI) Web Services standard to the limit, and it's time for a new standard [38]. IBM says that the UDDI standard for registries isn't cutting it, and the time is now for a new registry standard more focused on today's service based systems realities. "Our clients are telling us that they have an integration pain point," (Andrew Hately, a manager at IBM's Software Group) "We need to create a new standard and the time is now". says that so far all attempts to set up publicly accessible UDDI-based Service brokers have not been very successful.

For these reasons, Semantic Web Service Selection has been the chosenarea of elaborate study and intensive research by many a research worklike this one. Selection of Semantic Web Services are interesting to users onlyif those architectures address their issues of interest. After the reviews on

Web Service based Systems and Service Oriented Architecture, the following concerns have been identified. These are expected to be the concerns of different technologies and are the basis and motivation for this thesis.

- The available connection information is just not enough in Web Service registries of federated organizations; it should be able to share more about the Service requester's connection information to make better predictions.
- Web Service ServerArchitecture should be enhanced from its current state and it should support automated Web Services Registration, Availability and Quality Assessment.
- Maintaining data integrity should be the top prioritized concern and the Web Service registries should not have any weakness in maintaining the data integrity of Services once they have been registered with the system.
- The Web Services' information on an UDDI registry is to be updated as and when the changes are done by the Web Services Providers and hence the UDDI registries are expected to keep the information for registered Services up-to-date.
- For a Web Service registry to be successful as a resource recovery Service, it must be able to respond to requests from the Service Consumers in the least possible time with fewer message passes across the network.

- A Web Service registry must ensure high availability of the registered information to its users and the available information must be consistent across all the registries which are maintaining information about the services.

The concerns of an effective Semantic Web Service Selection are well laid out and that is the reason why the aim of this thesis is to deliver an effective model and an error free approach to provide Services information which is consistent. Due care has been taken to see that consistent information reaches the users in minimum response time.

A service lattice reveals the invisible relation between the services, easing the selected and identifying the possible alternatives. The major drawback of the previous work is that the semantic information is not considered during clustering of web services.The present research considers lattices as they are easy to represent and also provide the hidden relation between the web services that are present. Anattempthas also been made to intake the semantic information and produce high performance in the web services.

This research proposesa similarity measure that calculates the semantic similarity of the operations provided by the web services. The proposed similarity measure considers the harmonic mean of *WordNet* and *Normalized Google Distance* for calculating the similarity between the operations.

3.2 AIM, GOALS AND OBJECTIVES

3.2.1 Introduction

The primary aim of web service is to facilitate interactions between the software systems using XML standards. The architecture of web services is also designed in such a way that it employs XML standards to define and describe web services. This architecture also enables some of the web services functionalities such as selection and composition to be done automatically to a certain extent.

3.2.2 Aim

The main aim of the proposed research is toDesign a User Context Logic Dependent Integrated Dynamic Web Service Selection Modelusing Self-Organization Genetic Algorithm based constraint Initialization.The motivation has been the inherent shortfalls of the existing models of webserviceselection in the Web Service Computing environment.

3.2.3 Goals

The goals of this research are set so as to formulate a User Context Logic Integrated Dynamic Selection of web servicesto improve the performance in terms of various QoS factors corresponding to the web service requested in the

UDDI. So, in principle, the goals of the proposed research have three folds as follows:

i. To Design a Web Service Clustering Model that enables User Context Logic Dependent Integrated Dynamic Web Service Selection Framework.

ii. To Design and Develop a Self-Organized Genetic Algorithm to Optimize the proposed CO model.

iii. To Evaluate and Validate the performance of the Proposed Selection Framework using suitable Performance Assessment Factors.

The First goal is defined so as to develop a web service clustering model based user context logic method, and the second goal is defined so as to design an effective Self-Organized Genetic Algorithm to Optimize the proposed web service selection model based Combinatorial Ooptimization model for improving the efficiency of the service retrieval operation. Third objective is to assess the performance of the proposed selection framework using suitable Performance Assessment Factors.All the goals are measureable and of course proved with appropriate set of experiments.

3.2.4 Objectives

In view of achieving the goals defined, the research is organized into several phases and the major objectives are listed as follows:

1. To Design the Web Service Selection as Combinatorial Optimization Problem and Methodology to Optimize using GA.
2. To Formulate a Context Oriented Model (COM) to Bridge the Gap between the Service Request and the Service Discovered.
3. To Develop and Propose an Optimal Web Service Selection Framework with Constraint Initialization for GA.
4. To Design and Develop the Testbed using Benchmark Instances from QWS Datasets and Performance Factors.
5. To Evaluate the Performance of the Proposed Web Service Selection Framework using Genetic Algorithm based Performance Factors.

3.3 RESEARCH METHODOLOGY

3.3.1Research Methodology Framework

The layered view of the experimentation framework is illustrated in the Figure 3.1. It consists of five different layers; Base Layer, Functionality Layer, Performance Attributes Layer,Operations Layer and the Assessment Criteria Layer. The responsibilities of each layer are defined and described in the following sections.

3.3.3.1 Base Layer

The Base layer of the experimentation framework is the most important part of the present work,The current stage of Web Services Publish, Disocveryand Selection modelsare largely regulated based on Centralized UDDI registries. Although centralized registries can provide effective methods for Web Services selection, they suffer from problems associated with centralized systems.

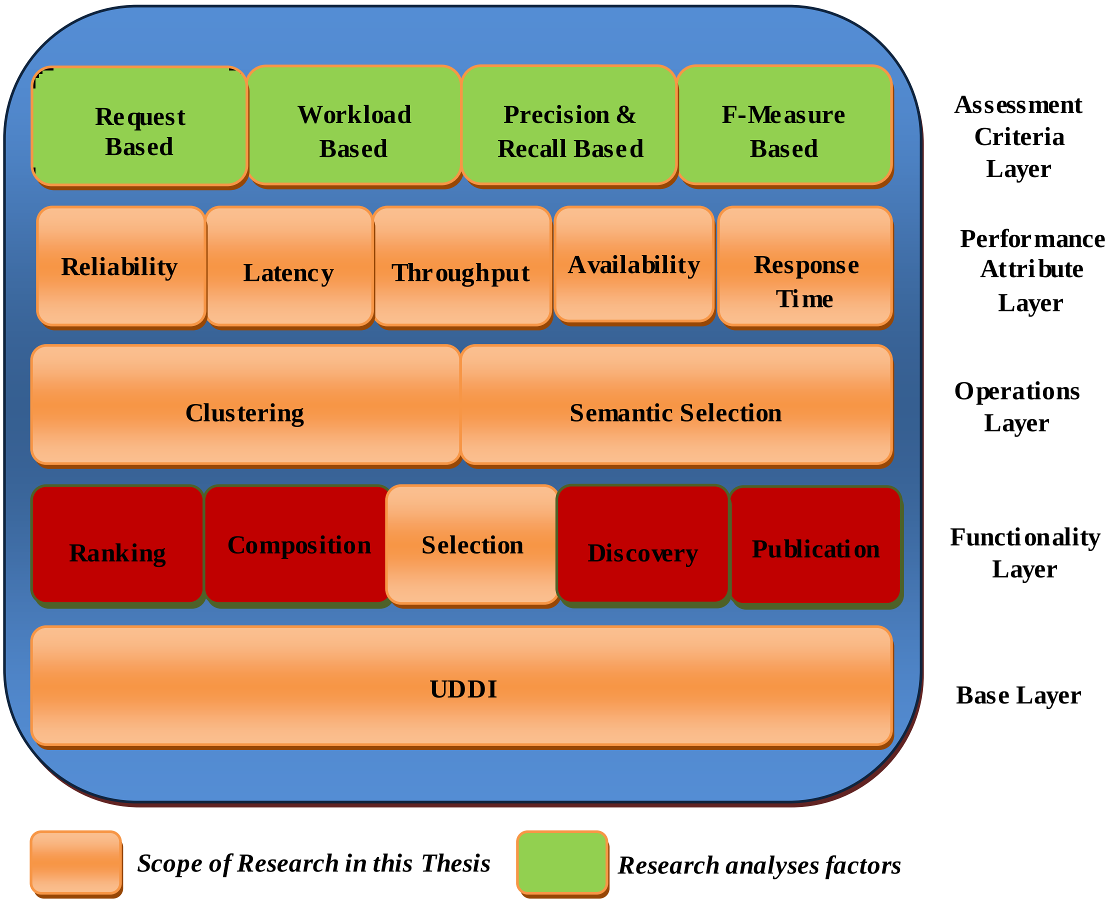

Figure 3.1 Experimentation Framework

The main advantages of Peer to Peer (P2P) systems are their very high robustness and scalability due to inherent decentralization and the ability to utilize large amounts of resources available on peers connected to the UDDI network. The P2P-based Distributed UDDI framework makes the decentralized system more scalable than traditional Web Service centralized UDDI systems by way of distributing the system function among few peer UDDI nodes and not focusing on only one UDDI server.

3.3.3.2 Functionality Layer

This layer is responsible for addressing various functionalities that the UDDI is concerned with. The main functionalities of UDDI are Composition, Selection, Ranking, Discovery and Publication. Web Service publishing means registering a Web Service in the UDDI registry and making it available to the Service Consumers. This will also guarantee the transfer of Web Services description to the consumer, which will be useful for the consumer to learn the way to interact with that Web Service.

According to predefined business requirements, web service composition refers to a process of adaptively composing a set of available Web Services into a business process flow. Web Service Discovery deals with finding a set of services that corresponds to a predetermined user request while Web Service selection deals with choosing a service from a set of discovered services. Web Service ranking is the process of assigning rank to the discovered Web Services based on user requirements to enable easy service selection.

3.3.3.3 Performance Attributes Layer

This layer possesses the core characteristics of this research, whose components are the various performance QoS attributes of the Semantic Web Service Selection which are managed in the UDDI system. The QoS attributes concerned with this layer are Reliability, Latency, Throughput, Availability, and Response Time.TheQoS attributes of Semantic Web Service Selection are,

- The *Response Time*, it is mainly based on the time duration between a service user sending a request and receiving the corresponding response. It is measured in millisecond.
- The *Availability*, it is the ratio of the number of successful invocations to total invocations. It is measured in percentage.
- The *Throughput*, it is the total number of invocations for a given period of time. It is measured in invocations per second.
- The *Latency*, it is the time taken for the server to process a given request. It is measured in milliseconds.
- The *Reliability*, it is the ratio of the number of error messages to total messages. It is measured in percentage.

Although all these attributes have been identified, due to time constraint, this research focuses only on select performance attributes which have the maximum effect on Semantic Web Service Selection model, such as Response Time, Availability, Throughput, Latency and Reliability.

3.3.3.4 Operations Layer

This layer is responsible for different types of operations performed in the system. Clustering and Semantic Selection are the two major operations identified for research on this layer. The Semantic Selection of the web services are done keeping in mind the issues related to Latency,Availability, Reliability, Throughput and Response time of the Services' information. Operations in this layer are performed in such a way that the system experiences an appreciable performance in terms of the performance attributes discussed in the performance attributes layer.

3.3.3.5 Assessment Criteria Layer

The Assessment Criteria Layer of the framework is proposed to take care of the responsibilities of ascertaining whether the proposed User Context Model for Semantic Web Service Selection framework functions and delivers the services as envisaged. It is proposed to assess the framework on the following basis: Request based, Workload based, Precision & Recall based and F-Measure based. Basically the assessments are done in the performance attributes layer to prove that the base layer of the proposed framework is able to achieve effectively the functions proposed in the functionality layer for processes identified in the operation layer. Various methods of assessments of the proposed system for the expected and proposed qualities are discussed below.

- Request based assessments are performed to verify the total number of requests with response time in the semantic clustering method. The proposed User Context Model for Semantic Web Service Selectionallows the highest performance when compared to the previous works.

- Workload based assessments are performed to verify the total number of User load with response time for semantic clustering method. It is expected to be as minimal as possible in the proposed User Context Model for Semantic Web Service Selection compared to the traditional clustering model.
- Precision and Recall based assessments are performed to the precision values of the existing and the proposed methods and they clearly indicate that the proposed methods yield high precision and the value tends to be the ideal case.The recall values of the existing and the proposed methods are assessed and they clearly indicate that the proposed methods yield high recall and the value tends to the ideal case.
- F-Measures based assessments indicate the improved values of F-measure, which is a harmonic mean of precision and recall of the discovered services in the proposed method. The F-measure values of the existing and the proposed methods clearly indicate that the proposed methods yield high F-measure and the value tends to the ideal case.

Based on the above discussions on the different methods to be employed for the assessment of the proposed system, it is proposed to demonstrate the better performance of the proposed User Context Model for Semantic Web Service Selection.

3.3.2 Similarity Measures

A novel similarity measure is proposed, which calculates the semantic similarity of the operations for discovering the web services. The proposed similarity measure considers the harmonic mean of WordNet [25] and

Normalized Google Distance [16] for calculating the similarity between the operations. If op_{ij} and $op_{kj}$are the two j^{th} operation provided by the two distinct web services ws_iandws_k, then the semantic similarity can be calculated as ***SemSim(op_{ij}, op_{kj})*** as shown in equation (1),

$$SemSim\,(op_{ij}\,, op_{kj}\,) = \frac{2*WSim\,(op\,_{ij}\,,op\,_{kj})*NGD(op\,_{ij}\,,op\,_{kj})}{WSim\,(op\,_{ij}\,,op\,_{kj})+NGD(op\,_{ij}\,,op\,_{kj})} \quad ..(1)$$

where,

- WSim(x,y) is the WordNetSimilarity score of the two words x and y.
- NGD(x,y) is the Normalized Google Distance between the words x and y.

This research considers both the WordNet and Normalized Google Distance as the combined score would be more appropriate for measuring the relatedness of the web service operations.The present research has chosen these two similarity measures since they measure the semantic similarity between the words as assessed by Google Search Engine and WordNet dictionary.

3.4 SUMMARY

Web service based systems have considerable advantages and they hold a lot of promise. Apart from being inexpensive and reliable they have a self-healing infrastructure that reduces management costs, truly real-time decision-making applications and possibility of the compilation of a unified taxonomy of information. The field of Web Service based Systems and Service Oriented Architecture are the current topic of research and development in Computer Science, which is especially oriented to the needs of gaining and processing information in large-scale distributed systems such as the Internet.

From the perspective of the business value, the Web Service based Systems provide the benefits of the ability to more quickly meet customer demands, and to lower costs associated with the acquisition and maintenance of technology. Earlier research never took into account the semantic information during clustering of web services.The main aim of the proposed research is to propose a framework to semantically cluster and discover the Web Services using QoS based User Context Model. A similarity measure that calculates the semantic similarity of the operations provided by the web services is proposed. The proposed similarity measure considers the harmonic mean of *WordNet* and *Normalized Google Distance* for calculating the similarity between the operations.

It is a goal of this research to design a Web Service Clustering Model that enables time effective and User Context dynamic Selection of Web Services. It is also intended to develop a framework to discover the Web Services semantically from the Cluster developed based on the QoS Properties.

This chapter outlines the proposed Web Service Cluster Model wherein the various similarity measures and the formal concept analysis are discussed. After the detailed discussion on the proposed Web Service Cluster

Model (WSCM), the Concept Lattice Model (CLM), Semantic Web Service Selection Model (SWSSM) and User Context Logic Model (UCLM) are discussed before presenting the Overall Experimentation Framework. Consequently, the mapping model between the web service computing model and the conceptual stack is brought forth. The layered view of the experimentation framework has been elaborately illustrated, defined and described for clarity and logical progress of the research.

CHAPTER 4

INTEGRATED DYNAMIC WEB SERVICE SELECTION MODEL

4.1 INTRODUCTION

A Semantic Service Selection Model (SSSM) is used to select the web services that are clustered semantically. The selection also supports the backup services which are used in the case of composite web services. The selection of backup services are essential in case any web service forming the compositeweb application fails leading to the failure of the entire application. The potential backup services can be an additional feature in the model that lists some of the web services that performs some or all of the operations performed by the discovered web service.

4.2 PROPOSED FRAMEWORK FOR INTEGRATED DYNAMIC SERVICE SELECTION MODEL

Figure 4.1 presents the framework for integrated dynamic service selectionimplementation of web services. Source manager manages the service logic, enhances productivity and boosts application quality as it automatically versions files, labels and organizes them as they change during the development process. It locates the service in the service repository using service locator and checks authorization to use the service with service contract. Then it develops interface and service module for the extracted logic. Business logic evaluator evaluates the developed new logic and checks whether it is free from errors. Also it checks whether the extracted logic has dependency with any part of the service.

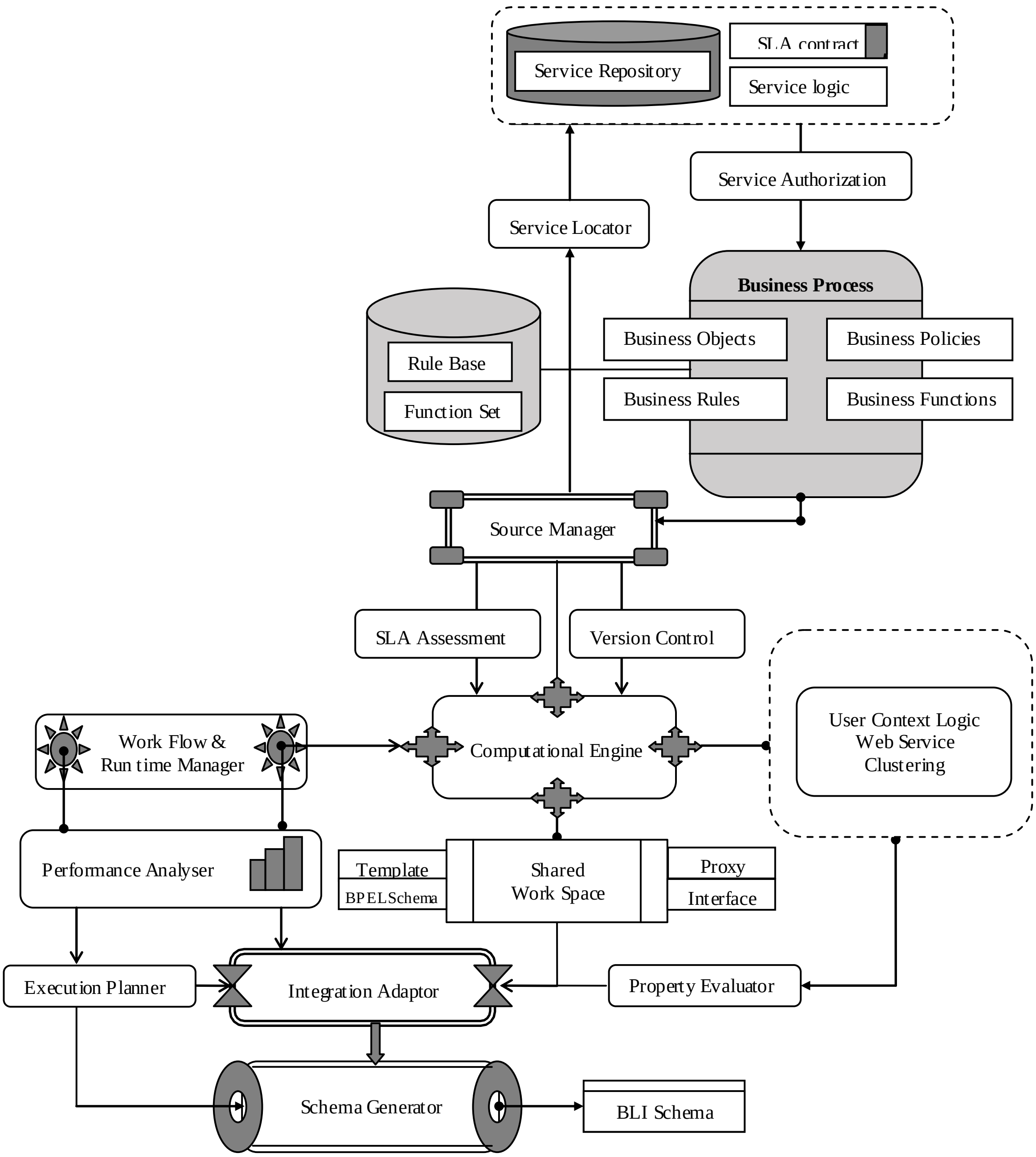

Figure 4.1 Integrated Dynamic Web Service Selection

Finally it builds the service. Workflow/ Runtime Manager analyze the build service and performance is evaluated through property such as computability and traceability. If all the rules and functions in the new logic are computable then whole logic is computable. Similarly, checks whether all rules and functions in the logic are traceable. Then it deploys and creates stub. This new logic is maintained as new version, business rule for the new logic also embedded into it.

Computational engine gets newly created stubs and analyzes the dependencies between them through FSM simulator and PBF evaluator. It places the stubs in the shared workspace and identifies the structure needed to integrate. In the shared workspace, new business logic is developed with the stubs using corresponding template. Property evaluator identifies the required property from the available properties such as union, composition, serialization and reducibility. It constructs the business logic composing the stubs in right way as fulfilling the client's requirement. Performance analyzer analyzes the performance of the newly developed logic. It evaluates five metrics such as timing, hardware counter, synchronization delay, memory allocation and tracing. According to the metrics formulated by performance analyzer, execution planner identifies efficient way to execute the logic and it troubleshoots if any part of the logic is performing poorly.

User Context Logic Web Service Clustering is module in which web services are clustered semantically based on the user context logic and from which web services can be dynamically selected. The working principle of Semantic Service Selection Model has been discussed in the next section. Finally Integration Adaptor builds and deploys the logic developed in the shared workspace. For the deployed logic, WSDL schema will be generated. This WSDL schema will be awarded to the client, he can develop front end for his application with this schema. Thus this framework integrates the service automatically without any developer's intervention securely and reliably at low cost.

4.3 SEMANTIC SERVICE SELECTION MODEL

The system design is based on the problem description and the objectives mentioned above. A test bed has been designed to address the objectives of this research and Figure 4.2 shows the semantic service selection model of the research. The following are the components used in the experimental test bed.

Service Provider: The service provider is the entity that has the web service and the service repository is maintained by the service provider. The Service Provider (SP_1, SP_2,SP_n in Figure 4.2) is termed as the actual owner of the web service and he registers the service to the UDDI which can be used by the service requestors.

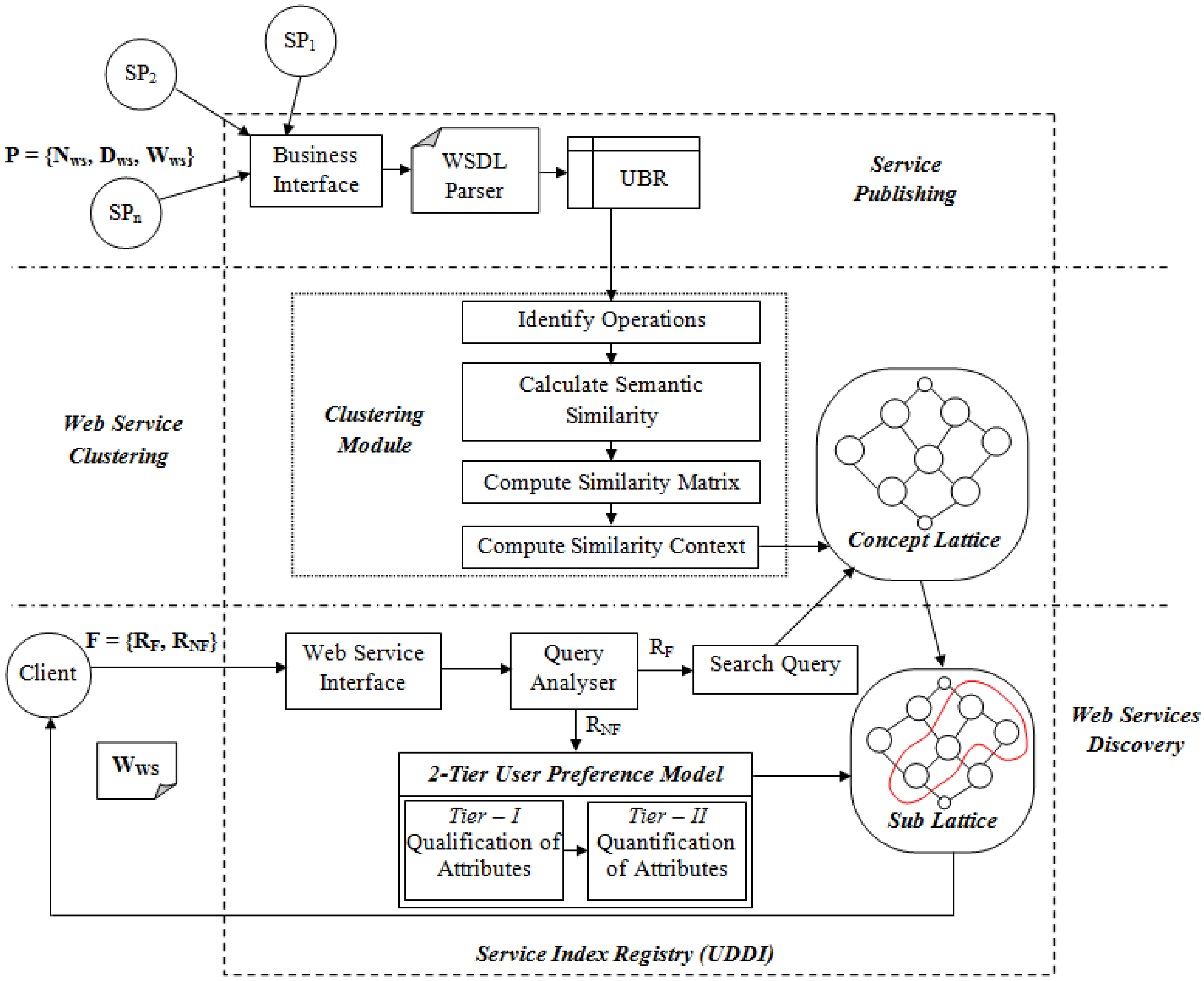

Figure 4.2 Semantic Service Selection Model

WSDL Parser: The WSDL parser is used to parse the WSDL file for finding the service name, descriptions, operations and end point of the web service. The data obtained from the WSDL parser are tabulated in the UDDI Business Registry (UBR) which is then looked up during the selectionprocess.

The WSDL parser implements the Algorithm 1 presented below which can be viewed as XML parsing.

Algorithm 1 WSDL Parser

In: Service Descriptions W_{ws}

Out: Service Name N_{ws}, Operations O_{ws} and Location L_{ws}

$O_{ws} = null$

for each service ws in W_{ws}

    get Service Name N_{ws}

    for each port p inws

        get Service Location L_{ws}

        get PortName, PortType

        for each Operation op in PortType

            $O_{ws} = O_{ws} U OperationName$

        end

    end

end

Business Interface: The business interface is used by the service providers for registering them as a service provider and to register the available service to the service registry. The business interface receives the service descriptions in the form of a WSDL document and the functionalities are obtained from the service descriptions.

UDDI Business Registry: The UDDI Business Registry (UBR) is responsible for creating the registry of web services. The service provider publishes the web service using the WSDL file along with the functional descriptions which are parsed by the *WSDL Parser* component and the service is registered with the UBR with its own description. Each entry in the UBR has its own service description and specifications which the service requestors search for finding the appropriate web services.

Clustering Module: The clustering module clusters the web services using concept lattices approach. A concept lattice behaves as a cluster of web services and the search process can be easily done in the lattice.

Web Service Interface: The web service interface is used by the service requestor or the client to search the services available in the service registry. The web service interface is the front end and the client communicates to the service registry using this component.

Query Analyser: The query analyser is an important component responsible for analysing the search query which is provided by the client and it classifies the query into functional requirement and non-functional requirements which in turn are given as inputs for the respective components.

User Preference: The user preference component is used to set the user preferences to the concept lattice. The user preference component obtains the non-functional requirements as the input from the *query analyser* component.

Search Query: The search query component obtains the functional requirement from the *query analyser* and searches the sub lattice formed as a result of applying the *user preferences* on the concept lattice and finally returns the web service descriptions to the client.

4.4 USER CONTEXT LOGIC MODEL

A *2-tier User Context Logic Model* (UCLM) is also proposed in order to improve the user desirable search results. The model is designed to reduce the semantic gap between the service request and the discovered service. Tier I deals with the Qualification of Parameters while Tier II deals with Quantification of the Qualified Parameters.

Definition 1:A User Context Logic Model (UCLM) can be described as follows,

$$UCLM: P_{qualify} \rightarrow P_{quantify} \quad (1)$$

where

- $P_{qualify}$represents the qualification of parameters
- $P_{quantify}$represents the quantification of the qualified parameters

The ***if-then*** relation between the two tiers indicates that the parameters must be qualified for quantification.

Definition 2: The quantification of parameters can be described as a tuple set containing

$$P_{quantify} = \{SR_{id}, PR_{id}\}$$

where

- SR_{id} is the unique identifier for the service requestor
- PR_{id} is the set of preferences for the particular service requestor and it can be defined as

$$PR_{id} = \{P_1, P_2, \ldots\ldots, P_n\}$$

where P_i refers to any of the quality of the service parameter.

This work considers only five QoS parameter (n=5) values which are shown below:

- *Response Time,* which is the time duration between service users sending a request and receiving the corresponding response.
- *Availability,* which is the ratio of the number of successful invocations to total invocations.

- *Throughput*, which is the total number of invocations for a given period of time.
- *Reliability*, which is the ratio of the number of error messages to total messages.
- *Latency Time*, which is the time taken for the server to process a given request.

The user preferences are obtained from the user and a sub-lattice is formed based on the preferences from the clustered lattice. This will reduce the semantic gap and the time for selection. The sub-lattice formed from the clustered lattice contains the services and operations that can be relevant to the user as the preferences will be set by the user. The semantic gap between the user request and the discovered web services can be reduced to a large extent using this approach.

4.5 SEMANTIC SELECTION MODEL

The clustering is based on lattices, as they are easy to implement and provide the hierarchical relation among the services. The semantic service selection model can be viewed as a sub-lattice search on a graph. The SSM can be realised if the clustering technique is based on lattices. The lattices can be represented using graphs and it will be easy to search on graphs rather than a vast search space. The output of lattice clustering approach gives us a lattice that is a cluster of web services based on its operation and ontology. The lattice can also be used to find the potential backup services for the discovered web service. Thus a lattice based clustering will enable semantic service selection. The Algorithm 2 depicts the depth first search algorithm used in this research.

Algorithm 2 Sub-LatticeSearch(L_{SUB}, n)

In: Sub-Lattice L_{SUB}, Query q, Parameters p

Out: Service Name N_{ws}

Label n as explored

For all links l in AdjacentLinks of L_{SUB} do

    If link l is not explored then

        Get the N_{ws} of node n with q and p

        If node of N_{ws} is not explored then

            If SemSim evaluates to TRUE then

                Call Sub-LatticeSearch(L_{SUB}, node of N_{ws})

            EndIf

        Else

            Label node of N_{ws} as a back link

        EndIf

    EndIf

End

The classical depth first search algorithm can be used to search the sub-lattice as lattices can be represented using graph data structure. The links are explored from the sub-lattice recursively and the proposed semantic similarity measure is evaluated for calculating the relevance, which discovers the desirable web services as per the user request. Thus the proposed framework can enable the selection of user desired results.

4.6 SUMMARY

To summarize the chapter, a semantic service selection model with a novel semantic similarity measure has been proposed. A two-tier user preferential model has also been used to discover user desirable services from the quality point of view. The user preferences are obtained from the user and a sub-lattice is formed based on the preferences from the clustered lattice. This will reduce the semantic gap and the time for selection. The sub-lattice formed from the clustered lattice contains the relevant services and operations since the preferences are set by the user. The semantic gap between the user request and the discovered web services can be reduced as a result of this approach.

CHAPTER 5

COMBINATORIAL OPTIMIZATION MODEL FOR DYNAMIC WEB SERVICE SELECTION

5.1 INTRODUCTION

In the existing approaches, the web service selection operation has been perceived as a textual representation based comparison and retrieval mechanism [29]. In these systems, the user service request query with the functional and non-functional attributes are considered as static values and so the possible combination of QoS attributes are not taken into consideration. Therefore, the existing system performs the service selection operation with the limited search spaceand so there exist a lot of chances to oversight the possible best combination of QoS parameters to satisfy both the requestor and the provider. In this perspective, an effective optimal web service selection framework has been proposed in this research. The proposed framework considers both the functional and non-functional attributes of the service to discover the user required service from the UDDI. Then, it considers the retrieved services as a part of a combinatorial optimization problem and it is optimized using the novel GA with constraint initialization. The resulting service is the best optimal solution with the combination of QoS parameters as required by the user and also it conform the win-win approach of web service selection.

5.2 WEB SERVICE SELECTION AS A COMBINATORIAL PROBLEM

In the proposed approach, a set of services S_i where i ϵ {1, 2,….,n} and each service has its own set of Functional and Non-Functional parameters. The functional parameter refers to the set of factors that are directly concerned with the functionality of the service, on the other hand, the non-functional parameter refers to the set of factors that define the quality of the service irrespective of the functionality of the service.

The Functional parameter includes Input, Output, Precondition and the Effect and the Non-functional parameter includes Availability (Av), Reliability (Rel), Throughput (Thput), Incentive (Inc), Service Reputation (SeRup), Service Provider Reputation (SePRup), Accessibility (Acc), Successability (Succ), Standard Adoptability (Std_Ada), Standard Conformability (Std_Conf), Transaction Integrity (Tran_Int), Collaborability (Colla), Informability (Infm), Controllability (Ctrl), Authorization (Auth), Authentication (Authen), Non-Repudiation (NonRep), Privacy (Priv), Response time (ResT), Price (Price), and Penalty (Penalty).

In the general web service model, the service selection operation search for service, using user given keyword, in constraint of both functional and non-functional parameters. This approach of search would increase in the search space of the relevant web service. Thus, initially filtering the service with respect to the functional requirements and then shifting using the non-functional requirements would reduce the size of the search space at greater extend. For example, let the UDDI consists of 10,000 services of various domains. Consider that user wants to retrieve a Bus ticket booking service, then it is necessary to search both functional and non-functional properties of all the 10,000 services in order to retrieve an optimal service and consequently it would be time consuming and cumbersome task.

One simple way to deal with the scenario is that instead of searching and analyzing all the 10,000 services in the UDDI, users can search for the service within a collection of services which has similar functionality of that user requests. In order to achieve this, each service in the UDDI has to be segregated based on its domain. Thus, a new selection request would first filter the requested service based on the domain using functional requirements and then based on non-functional requirements. This would reduce the overhead of the search and also the overall search space would be reduced. Thus, as given in the example above, the user requested bus ticket booking service which isdiscovered based on the domain that is its functional properties and then the bus ticket booking service non-functional parameters are analyzed to get an optimal bus ticket booking service.

In other words, in order to reduce the search space of the web service discover, the services that match the user query (Uqur) based on the Functional parameters or text based matchmaking (keyword or dictionary based service selection) are retrieved. Now the services are again filtered based on the user context specified QoS parameters (like Price, Response Time, Reliability, Availability) maximum and minimum range, thus the services that matches the user specified QoS parameter range criteria would be retrieved as candidate servicesS_i.Now, these candidate services are given as input to the further optimization process of the serviceselection.

Services retrieved from the QoS matchmaking would be considered as a matrix (Sij) of services that can be represented as.

$$S_{ij} = \begin{bmatrix} q_{11} & q_{12} & q_{13} & \cdots & q_{1j} \\ q_{21} & q_{22} & q_{23} & \cdots & q_{2j} \\ \vdots & \vdots & & & \\ \vdots & \vdots & & & \\ q_{i1} & q_{i2} & q_{i3} & \cdots & q_{ij} \end{bmatrix}$$

where,

- ‘i’represents the total number of services in the UDDI
- ‘j’ represents the particular QoS parameter and
- ‘q’ represents thecorrespondingQoS parameter value

The notable part of the web service selection is that the user defines the range of QoS parameters required and on the selection process the value of the corresponding [parameter can be chosen within the range so that it can be beneficial to the user and also the provider. This can be considered as the WIN-WIN policy approach of solving any problem. Thus, during service selection the service parameter values can be optimized in combinatorial fashion to obtain the best and the optimal solution for the request of the user. Hence, the optimization problem used in our proposed system can be considered as the combinatorial optimization problem (a finite number of feasible solutions). Similar to the famous combinatorial problems like Travelling Salesman Problem (TSP) and Vehicle Routing Problem (VRP), the web service selection can be solved as a combinational based optimal solution finding problem.

5.3 SOLVING WEB SERVICE COMBINATORIAL PROBLEM USING GA

The combinatorial optimization is the mathematical study of identifying anbestarrangement, grouping, ordering of objects that are usually determinate in numbers. The effective methodology to solve any combinatorial optimization is a meta-heuristic optimization. As contrasted to the exact methods of problem solving thatassure to offerhe best optimum individual for the problem, the meta-heuristic optimization only tries to provide a good, but not exactly the best optimum solution for the problem. The meta-heuristic problem solving techniques are not tangled to any specific type of problem and are general approach that can be modified to solve any optimization problem.

The meta-heuristic optimization techniques are flexible, often global optimizers and robust to the size and the instance of the problem considered. The common and famous meta-heuristic techniques are Simulated Annealing (SA), Genetic Algorithm (GA), Tabu Search (TS), Ant Colony and Swarms Computation. Among the others, Genetic Algorithm is the preferred meta-heuristic approach to solve complex combinatorialproblems with very large search space. The traditional GA consists of following steps initial population (population seeding), selection, crossover, mutation and termination constraint in which first step occurs once and the rest of the steps are repeated until the termination condition is satisfied [32, 33].

The methodology to solve web service selection using genetic algorithm is discussed as follows:

The first step of the GA is to generate a set of possible solutions randomly as initial population or population seeding. Since the seeded individuals are to be propagated throughout the lifespan of the GA at least in a partial manner, the quality of individual solutions in the initial population plays a critical role in determining the quality of the final solution that can be obtained using GA. In this phase, the initial populationwould be randomly generated based on the user request query.The non-functional requirements are the key components in generating the initial solutions. Each selected service would be considered as a solution or chromosome.

Selection is an important phase in GA, which selects two or more appropriate individuals from the current population for recombination. The selection process performs operations such as selecting good-fit individuals for crossover or mutation operation and also for elitism transfers. During selection step, better solutions of the current generation are selected for two purposes; the first is to transfer the better fit solutions to the next generation to employ elistis and the second one is to perform crossover operation on the selected solutions in order to improve the fitness further.

The crossover phase comprehends the construction of the offspring using the individuals selected from the current population. In general, the crossover operator has to get as much good information from the parent individuals.Several crossover operators are available like one-point crossover, two-point crossover, cut and slice uniform crossover, half uniform crossover and three parent crossover. During crossover, each non-functional components of each individual are modified to satisfy both the max-min value and also the user required value.

Mutation is an important GA operator to enhance the diversity of the population and hence facilitates GAs to explore potential solution areas of the search space. Thus, the ultimate purpose of the mutation stage is to provide insurance against the irrevocable loss of genetic information and hence to maintain diversity within the population.During mutation, it should be supervised that the values are transformed within the range of the max-min value of the corresponding QoS component.

Fitness evaluation involves defining an objective or fitness function against which each chromosome is tested for suitability for the environment under suitable constraints. As the algorithm proceeds,it would be expected that the individual fitness of the "best" chromosome to increase as well as the total fitness of the population as a whole. Though, maintaining diversity between the best and worst individual in the population is also a mandatory factor.

5.4 OPTIMAL WEB SERVICE SELECTION FRAMEWORK

The overall design of the proposed system for the web service selection, considering as a combinatorial optimization problem, has been shown in the Figure5.1, which illustrates the selection and selection of the service from a list of services. The proposed system consists of three main stages, namelyfunctional matchmaking,numeric based QoS range matchmaking and combinatorialoptimization.

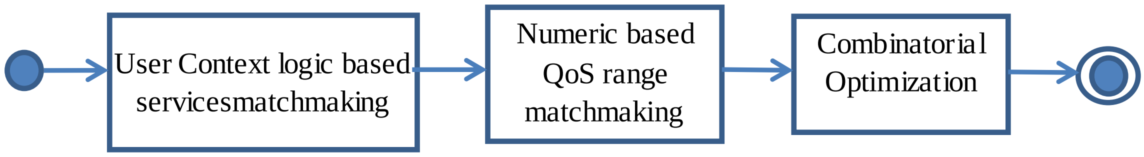

Figure5.1 OverallStructure of the Proposed System

The primary purpose of the each stage has been given below:

*Stage 1 - Functional Matchmaking:*It filters the large collection of servicesusing the user request query based on their functional attributes.

Stage 2 -Numeric based Qos range Matchmaking: The non-functional factor values of the filtered services are retrieved and examined with respect to the user specified QoSfactorvalue max-min range.

Stage 3- Combinatorial Optimization: The resultant services are recombined to identify the best optimal solution among the conditional feasible solutions in combinatorial problem solving fashion.

The comprehensive Optimal Web Service Selection Framework has been presented in the Figure 5.2. It portrays the sub-modules involved at the each stage of the propose service selection framework. The flow of information from one stage to another stage have been represented by a dotted arrow and the flow of information from one sub-module to another have been represented as an arrow within each phase.The detailed functionalities of each stage and its sub-modules are discussed as follows:

The system takes a user query which may be the textual information describing the functionality of the service and also the non-functional QoSattributes values as required by the user. The functional matchmaking part of the system filters the services in the service repository (ie. UDDI), using the user request query*Uqur*, based on the functional properties like input, output, precondition and effect. The services that match these four functional properties which is represented as *f(i,o,p,e)* are discovered from the available services from the UDDI repository that is represented as *S(fm).* Thus, theresultant servicesare the collection of service that belongs to aparticular domain, same functional characteristics,based on the user function requirements.However, each service retrieved are of different QoSparameter values.

The next stage of the system is Numeric based QoS range matchmaking in which the services retrieved *S(fm)* are given as the input and their non-functional parameter values are assessed and analyzed with respect to the user specified QoSattributevaluemax-min range. At this point, the QoS attributes are divided into positive attributes and negative attributes; the positive attribute,represented as *q(p)*, refers to the QoSattribute whose value is directly proportional to the goodness of the service like availability, reliability and throughput. The negative attribute, represented as *q(n)*, refers to the QoS attribute whose value is inversely proportional to the goodness of the service like response time and price.

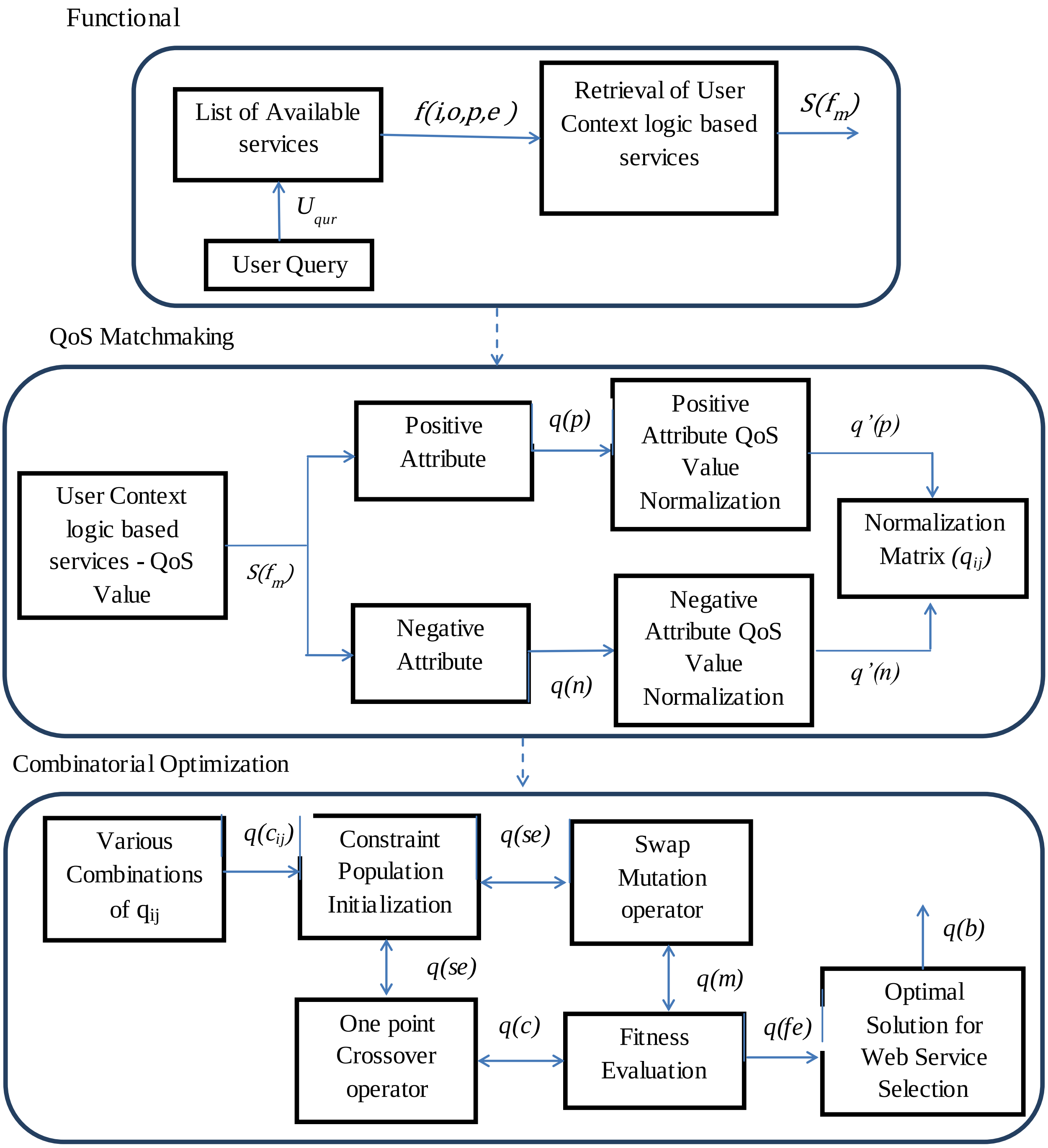

Figure 5.2Optimal Web Service Selection Framework

The segregated QoS attributes should be of different ranges and it is necessary convert them into a specific form in order to perform the standard comparative evaluation. Thus, both the type of attributes is given to the normalization process and normalized disjointedly. The normalized positive and negative QoS attributes are represented as *q'(p)* and *q'(n)* respectively. The process of normalization of different types of attributes is described in the Chapter 5. Subsequently, the normalizedvalue of the positive and negative QoS attributes are combined into a single normalization matrix represented as q_{ij}. The outcome of the stage would be the services that satisfy both the functional as well as the QoS parameter values of the requestor.

The final stage of the system is recombining and retrieving the optimal solution, using combinatorial optimization technique, for the user request from the outcome of the previous stage of the system. In the optimization phase, the genetic algorithm has been used as the optimization algorithm considering the filtered solution as the initial population. The population initialization phase of the GA has been proposed with a novel methodology such that the QoS attribute values of the each individual must be within the range of the corresponding service. Thus, the newly proposed population initialization technique can be called as Constraint Initialization and consequently the proposed GA can be referred as the Genetic Algorithm with Constraint Initialization (GA with CI).

The initial population generated would undergo a fixed number of generations of genetic operations such as one point crossover *q(c)* or/and swap mutation *q(m)* and the fitness is evaluated for each individual at each generation of the GA which is represented as *q(fe)* and the final optimal solution as *q(b)* would be obtained at the end of the fixed number of generations of the GA.

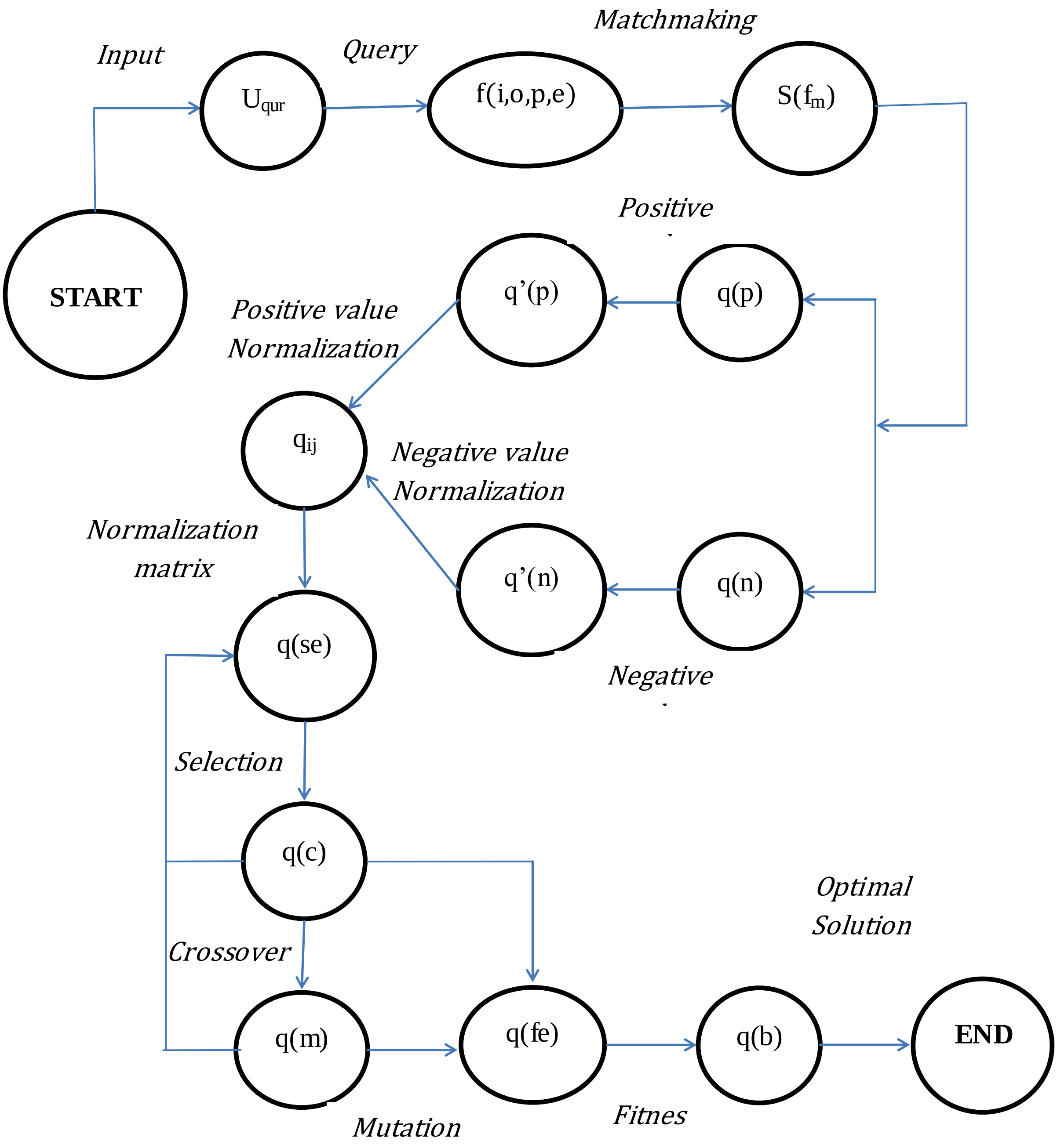

Figure 5.3 State Representation of the Framework

Thus, the output of this stage, *q(b),* is the best optimal solution which is the service that satisfy both the functional and non-functional requirements of the user and placate the concept of the win-win approach. That is, the service obtained would be beneficial to the service requestor and the service provider as well.

The working of the proposed web service selection framework can be better explained using the state transition representation. The state transition representation of the proposed optimal web service selection framework has been shown in the Figure. 5.3. The figure consists of a start state and end state; the transition of control from one state to another has been indicated by an arrow.

5.5 GENETICALGORITHM WITH CONSTRAINT INITIALIZATION

This section presents aninclusive description on the cotrsaint initialization and the objective function of the novel genetic algorithm proposed to effectively optimize the combinatorial problem of web service selection.

5.5.1 CONSTRAINT INITIALIZATION

The population initialization is a critical step in the GA that decides the final optimal solution of the GA after several generation of genetic operations. Since the seeded individuals are to be propagated throughout the lifespan of the

GA at least in a partial manner, the quality of individual solutions in the initial population plays a critical role in determining the quality of the final solution that can be obtained using GA. Permutation encoding is the common representation technique for combinatorial problems in which every solution is a series of absolute numerical or alphabet values stand for the position in a sequence. In the web service selection problem, an individual/service is made of a series of value corresponds to the QoS attribute of the service. Let the service set S and each service consists of z attributes,

$$S = \{s_{1,}s_{2,} \ldots\ldots s_u\}$$

$$S_k = \{a_{1,}a_{2,}a_{3,}a_4, a_5, a_z\}$$

where,

- u is the total number of services in the UDDI
- z is the total number of QoS parameters for the k^{th} service in S

The positive and negative attributes of the service can be represented as follows,

Positive attributes -> $S_{pk} = \{a_{1,}a_{2,}a_{3,}a_4, a_5, a_{zp}\}$

Negative attributes -> $S_{nk} = \{a_{1,}a_{2,}a_{3,}a_4, a_5, a_{zn}\}$

where,

- zp is the total number of positive attributes of the service S_k
- zn is the total number of negative attributes of the service S_k

The number of services POP_{query} retrieved from the UDDI for the user query can be represented as,

$$POP_{query} = \begin{bmatrix} S_1 \\ S_2 \\ S_3 \\ \cdots \\ S_m \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \ldots\ldots a_{1z} \\ a_{21} & a_{22} & a_{23} & a_{24} \ldots\ldots a_{2z} \\ a_{21} & a_{22} & a_{23} & a_{24} \ldots\ldots a_{2z} \\ & & \cdots\cdots & \\ a_{m1} & a_{m2} & a_{m3} & a_{m4} \ldots\ldots a_{mz} \end{bmatrix}$$

where,

- ‘m’ is the total number of services retrieved for the user query
- ‘z’ is the length of the attributes of the service

The population initialization should have the constraint such that the value of the each QoS component of the service should be within the maximum value of the corresponding component (in case of a positive attribute) or within in minimum value of the corresponding component (in case of a negative attribute).

$$Ga_{ij} > User(a_{ij}) \text{ and } Ga_{ij} < Max(a_{ij}) \text{ if } a_{ij} \text{ is a positive attribute}$$
$$Ga_{ij} < User(a_{ij}) \text{ and } Ga_{ij} > Min(a_{ij}) \text{ if } a_{ij} \text{ is a negative attribute}$$

where,

- Ga_{ij} is the system generated value for the attribute a_{ij}
- $User(a_{ij})$ is the user required value for the attribute
- $Max(a_{ij})$ is the maximum value of the attribute offered by the service provider

- $Min(a_{ij})$ is the minimum value of the attribute offered by the service provider

Since the value of the system generated QoS attribute values are constrained between the user specified and the maximum/minimum value of the corresponding attribute, the population initialization mechanism is referred as the "Constraint Initilization". The proposed algorithm with this initialization is referred as Generic Algorithm with Constraint Initialization (GA with CI).

5.5.2 OBJECTIVE FUNCTION

A genetic algorithm must have one or more objective function which defines the methodology to evaluate the fitness of an individuals generated by the algorithm. The objective function depends on the nature of the problem required to be optimized. In the case of the combinatorial problem of web service selection, the objective function can be formulated as follows:In order to identify the best suitable web service for thequery of the user, a collection of candidate services S_i< S_1 , S_2 ,……, S_n> are considered and each service $S_i$would have a set of QoS parameters $q_{ij}$where j ϵ 1, 2,….,m. The symbol q_j of S_i and q_j of S_k where $i \neq k$ denotes the j^{th}QoS parameters of service S_i .Thus the objective function $f(q)$which can be used to calculate the fitness value of a service can be given as,

$$f(q) = max\left(\sum_{j=1}^{m} q_j\right) \;, \forall S_i \text{ where } i \in 1,2,\ldots,n$$

where,

- q_j is the j^{th}QoS parameter of the service.
- S_i is the i^{th} service in the service set S.

Fitness value can be calculated using equation that is summing up the set of q_j values where $j \in 1, 2,\ldots,m$ of service $S_i$wherei $\in 1, 2,\ldots,n$. This objective function would be calculated for each candidate solution to find the fitness of the corresponding service. The lesser fitness services are removed from the population, whereas the best fit individuals are selected for possible recombination operation.

5.6 SUMMARY

In this chapter, the web service selection operation has been portrayed as a combinatorial optimization problem and also an effective optimal web service selection framework has been proposed. The proposed framework consists of three stages namely, Functional Matchmaking, Numeric based Qos range Matchmaking, Combinatorial Optimization. The user query is the input of the first stage and further, the outcome of each stage is retained as the input for the next stage of the system. The combinatorial optimization stage uses a novel gentic algorithm with constraint initialization to identify the best optimal solution that satisfy both the function as well as the QoS parameter values of the requestor.

CHAPTER 6

EXPERIMENTATION AND RESULT ANALYSIS – PHASE - I

6.1 INTRODUCTION

The experimental setup has 1000 web services obtained from QWS Dataset [4]. The properties of the web services such as its name, operations, endpoints, bindings and QoS parameters including availability, response time, throughput, latency time and reliability are obtained by parsing the description of the web services. All properties of the services are tabulated in a database and they are simulated to act as UDDI Business Registry (UBR). The available set of services has been classified into some 21 domains. The number of services in each domain is listed in Table 6.1

Table 6.1 Classification of Domains

Domain	Airline	Automobile	Bio-Informatics	Banking	Conversion	Dictionary
No. of Services	37	48	83	30	87	34
Domain	Library	Employment	Entertainment	Financial	Education	Messaging
No. of Services	36	39	30	70	37	72
Domain	News	Postal	Miscellaneous	Search	Social Networking	Tourist
No. of Services	34	30	89	46	48	38
Domain	Tracking	Verification	Weather	*Total No. of Services*		
No. of Services	32	50	30	*1000*		

6.2 EXPERIMENTAL SETUP

This section describes the experimental setup developed for investigating the proposed framework to Semantically Cluster and Discover the Web Services using QoS based User Context Logic Model proposed in the Chapter 4. Experiments were performed over the benchmark instances obtained from the QWS Dataset [6, 9].

In the current scenario, there is no standard web service selection testing methodology and so, a list of real time web service provider are gathered and registered with the UDDI. In the UDDI, 1000 web services from 21 diverse domains have been stored which provide operations obtained from QWS Dataset as given in the Table 6.1. The users are given with an input text box to enter the keywords for web service search. The entered keyword will be searched for the semantics on the web service description available in the UDDI registry. This process retrieves a group of related web services for the input keyword. A set of metrics has been used to measure the relatedness between the different web services retrieved.

A Universal Discovery, Description and Integration (UDDI) has been realizedusing open source technologies. The collection of 1000 real web services arecollected from various repositories from the dataset for testing. Several researchers propose models for addressing the issues in the area of web services but only a few consider the real web service as a test dataset. This research uses such large real web services for testing the proposed model. A

UDDI is created with the available web services and the semantic service selection model is developed by extending the developed UDDI.

The objective is to calculate the aggregated score considering all the QoS parameters. The score of the services were calculated based on the values of the QoS parameters which enable the User Preference Model to discover the user relevant services. The QoS parameters are obtained with different scales and they are converted into a common scale from 0 to 100 which help in ranking the web services.The formula (2) used to change the scale is given below and the scaling is done according to the minimum and maximum values of the existing scale as well as the new scale.

$$ScaledValue = \frac{(Value_{ToScale} - Value_{FromMin})*(Value_{ToMax} - Value_{ToMin})}{(Value_{FromMin} - Value_{FromMax}) + Value_{ToMin}} \quad ..(2)$$

The formula for calculating the aggregated score (3) after considering all the QoS parameters is shown below. The parameters such as availability, throughput and reliability are given a positive indicator as they are directly proportional to the performance. The increasing value of the above parameters will increase the performance. The parameters such as response time and latency time are given a negative indicator as they are inversely proportional to the performance. The decreasing value of the above parameters will increase the performance. The services are ranked according to their score in the respective domains.

$Score_{Agg} =$
$(Availability + Throughput + Reliability) - (ResponseTime + LatencyTime)$ ..(3)

The experimentation of the proposed system was performed with Java Servlets. A UDDI structure is implemented and all the 1000 services are registered into that UDDI and made into a cluster using lattice based clustering technique. The selection algorithms proposed in *AzmehZenia et al.* and the two proposed algorithms areimplemented in the UDDI.

6.2.1 Assessment Criteria

The evaluation of web service selection can be viewed as similar to the evaluation of information retrieval system which can be used as metrics used for evaluating the former. The following are some of the metrics (adapted from [15]) used in accordance with web services. The metrics for evaluation include precision, recall and f-measure.

6.2.1.1 Precision (P)

Precision is defined as the fraction of retrieved services that are relevant to the user request. The formula (4) for calculating the value of precision is given below.

$$Precision = \frac{|S_{Relevant}| \cap |S_{Retrieved}|}{|S_{Retrieved}|} \quad ..(4)$$

where,

- $|S_{Relevant}|$ is the number of services that are relevant to the request.
- $|S_{Retrieved}|$ is the number of services that are retrieved.

The value of the precision lies between 0 and 1 and is normally expressed in percentage.

6.2.1.2 Recall (R)

Recall is defined as the fraction of relevant services that are retrieved to the user request. The formula (5) for calculating the value of precision is given below.

$$Recall = \frac{|S_{Relevant}| \cap |S_{Retrieved}|}{|S_{Relevant}|} \qquad ..(5)$$

where,

- $|S_{Relevant}|$ is the number of services that are relevant to the request.
- $|S_{Retrieved}|$ is the number of services that are retrieved.

The value of the recall lies between 0 and 1 and is normally expressed in percentage.

6.2.1.3 F-Measure

F-measure is an evaluation metric that considers both precision and recall. F-measurecanbe defined as the harmonic mean of precision and recall. The formula (6) for calculating the value of precision is given below.

$$F - Measure = \frac{2 * Precision * Recall}{Precision + Recall} \qquad ..(6)$$

The value of the recall lies between 0 and 1 and is normally expressed in percentage.

6.2.2 Experimental Phases

Experiments and Result Analyses are performed in two phases namely evaluation of the proposed Web Services Clustering Model andevaluation of the performance of the proposed Semantic Service Selection Model (SSSM) based on Clustering and UPM Models. Phase – I is evaluation of the performance of the Proposed Web Service Clustering Model.

6.2.2.1 Web Service Clustering Model

In this section of experimentation, the performance of the proposed web service clustering model has been evaluated based the factor of the total number of relevant service retrieved from the UDDI based on the user requested query. The proposed clustering technique has been compared against the recent and best working technique proposed by [2]. Each technique has been repeated for different number of queries ranging from 20 to 100 and the results shown are the average of 25 runs of the corresponding experiments. The Table 6.2 depicts the number of relevant services retrieved using different clustering techniques for the sample of services from the Airlines domain.

Table 6.2 The number of relevant services retrieved using different techniques

Domain	Query	Ideal Case (N_I)	Total Service retrieved			Relevant Service Retrieved		
			AzmehZenia (NE)	SSSM (NP)	UPM (NU)	Azmeh (NE)	SSSM (NP)	UPM (NU)
Airlines	20	21	11	**17**	**19**	7	**13**	**16**
	40	25	14	**26**	**26**	9	**23**	**24**
	60	28	17	**30**	**29**	13	**25**	**26**
	80	32	19	**33**	**33**	15	**27**	**29**
	100	37	22	**35**	**39**	17	**30**	**35**

Table 6.3 Rate of improvement w.r.t different service retrieval techniques

Domain	Query	Ideal Case (N_I)	NE/NI	NP/ NI	NU/ NI	% of improvement NE to NP	% of improvement to NU	% of improvement NP to NU
Airlines	20	21	0.33	0.62	0.76	85.57	128.57	23.08
	40	25	0.36	0.92	0.96	155.55	166.66	4.35
	60	28	0.46	0.89	0.92	92.30	100	4.00
	80	32	0.47	0.84	0.90	80	93.33	7.41
	100	37	0.46	0.81	0.94	76.47	105.88	16.67

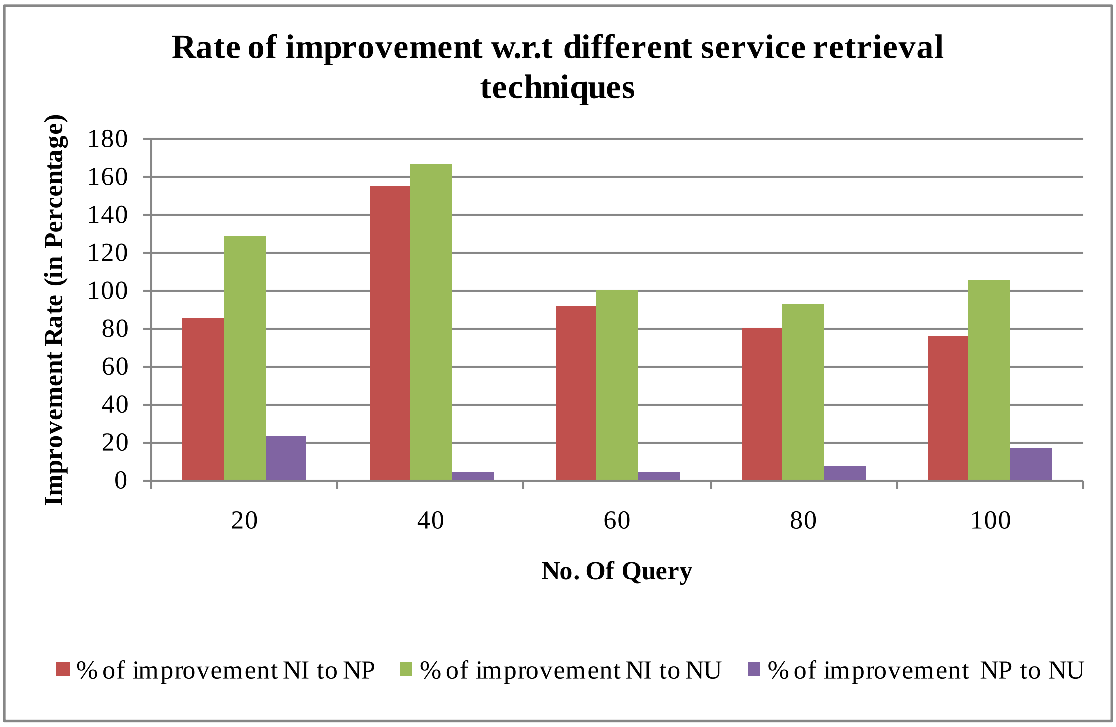

Figure 6.1 Rate of improvement w.r.t different service retrieval techniques

Based on the experimental results shown in the Table 6.2, the rate of improvement of different clustering and service retrieval techniques has been given in the Table 6.2. From the Table 6.2, it can be observed that the proposed semantic clustering model has more than 75% of improvement over the existing clustering method for almost all the different varieties of queries requested to the UDDI. Moreover, the proposed UPM model has obtained improved results over the existing method by obtaining more than 100% of improvement for almost all the different varieties of queries as illustrated in the Figure 6.1.

It can be observed that the rate of improvement of the proposed user preferential model over the proposed clustering technique is not significant for each different varieties of queries. This shows that the semantic service selection models utilize the enhanced clustering model designed to improve the relevance of the service retrieved with respect to the different serviced requested from the UDDI.

6.2.2.2 Semantic Service Selection Model

Phase – II evaluation of the performance of the proposed Semantic Service Selection Model (SSSM) based on Clustering and UPM Model.In this section of analyses, the performance of the proposed user preferential based semantic web service retrieval model has been evaluated based the factor of the Precision, Recall and F-measure derived from the total number of relevant service retrieved from the UDDI based on the user requested query. The proposed technique has been compared against the recent and best working technique proposed by [3].

The different quality measure results obtained from the existing (NE) and the proposed (NP and NU) service retrieval techniques for the sample domain of airlines have been shown in the Table 6.4. From the Table 6.4, it can be derived that the proposed techniques outperform the best existing model of service retrieval w.r.t. the different performance measures. It can also be observed that with increase in the number queries raised, the value of the quality measures are also get increased.

Table 6.4 Quality measures analysed for Precision service retrieval techniques

Domain	No. of Queries	Azmeh (NE)	SSSM (NP)	UPM (NU)
Airline	20	0.636364	**0.764706**	**0.842105**
	40	0.642857	**0.884615**	**0.923077**
	60	0.764706	**0.833333**	**0.896552**
	80	0.789474	**0.818182**	**0.878788**
	100	0.772727	**0.857143**	**0.897436**

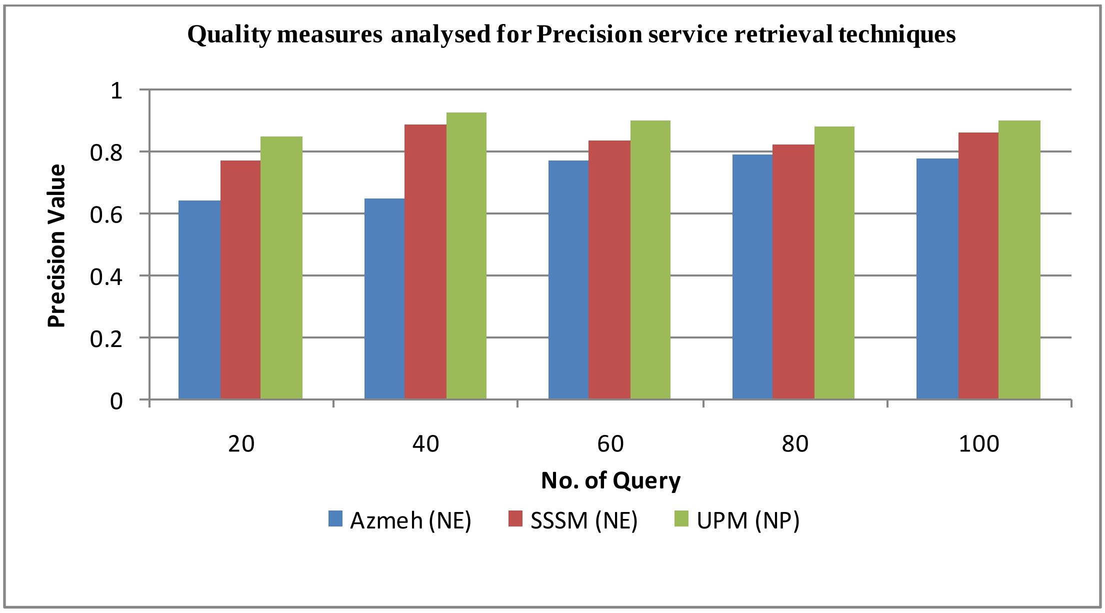

Figure 6.2 Precision measure for different service retrieval techniques

Table 6.5 Rate of Precision improvement for different service retrieval techniques

Domain	Query	% Improvement NI to NP	% Improvement NI to NU	% Improvement NP to NU
Airline	20	20.16807	32.33083	10.12146
	40	37.60684	43.58974	4.347826
	60	8.974359	17.24138	7.586207
	80	3.636364	11.31313	7.407407
	100	10.92437	16.13876	4.700855

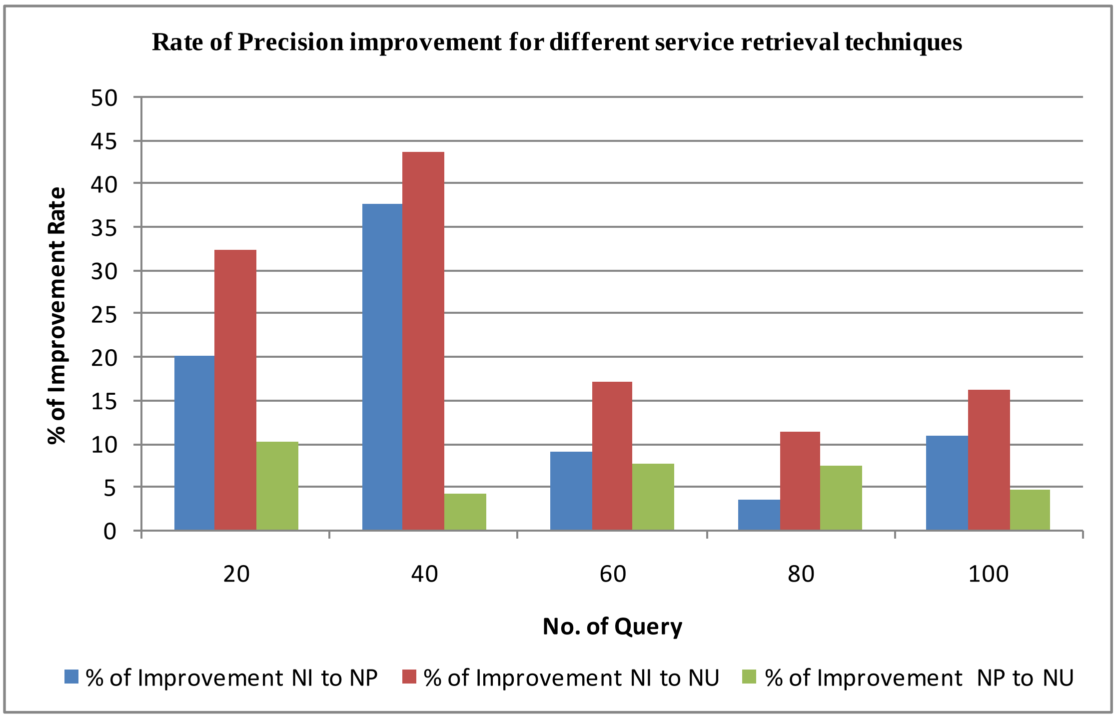

Figure 6.3 Precision improvement rate for different service retrieval techniques

Table 6.6 Quality measures analysed for Recall service retrieval techniques

Domain	Query	Azmeh (NE)	SSSM (NP)	UPM (NU)
Airline	20	0.333333	**0.619048**	**0.761905**
	40	0.36	**0.92**	**0.96**
	60	0.464286	**0.892857**	**0.928571**
	80	0.46875	**0.84375**	**0.90625**
	100	0.459459	**0.810811**	**0.945946**

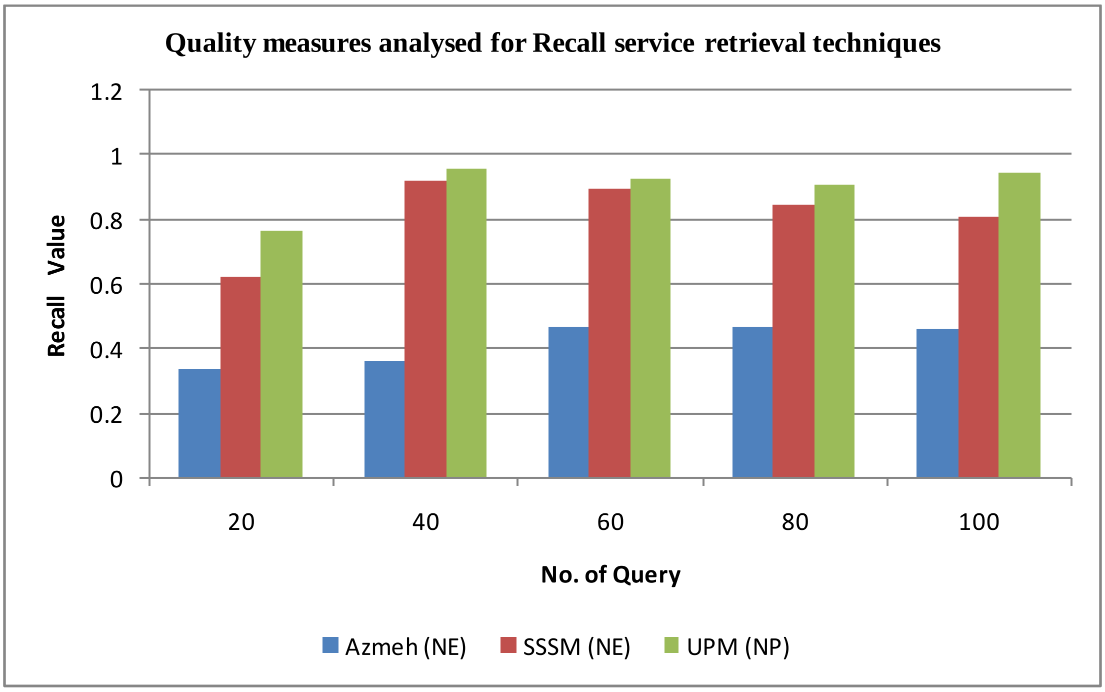

Figure 6.4 Recall measure for different service retrieval techniques

Table 6.7 Rate of Recall improvement for different service retrieval techniques

Domain	Query	% Improvement NI to NP	% Improvement NI to NU	% Improvement NP to NU
Airline	20	85.71429	128.5714	23.07692
	40	155.5556	166.6667	4.347826
	60	92.30769	100	4
	80	80	93.33333	7.407407
	100	76.47059	105.8824	16.66667

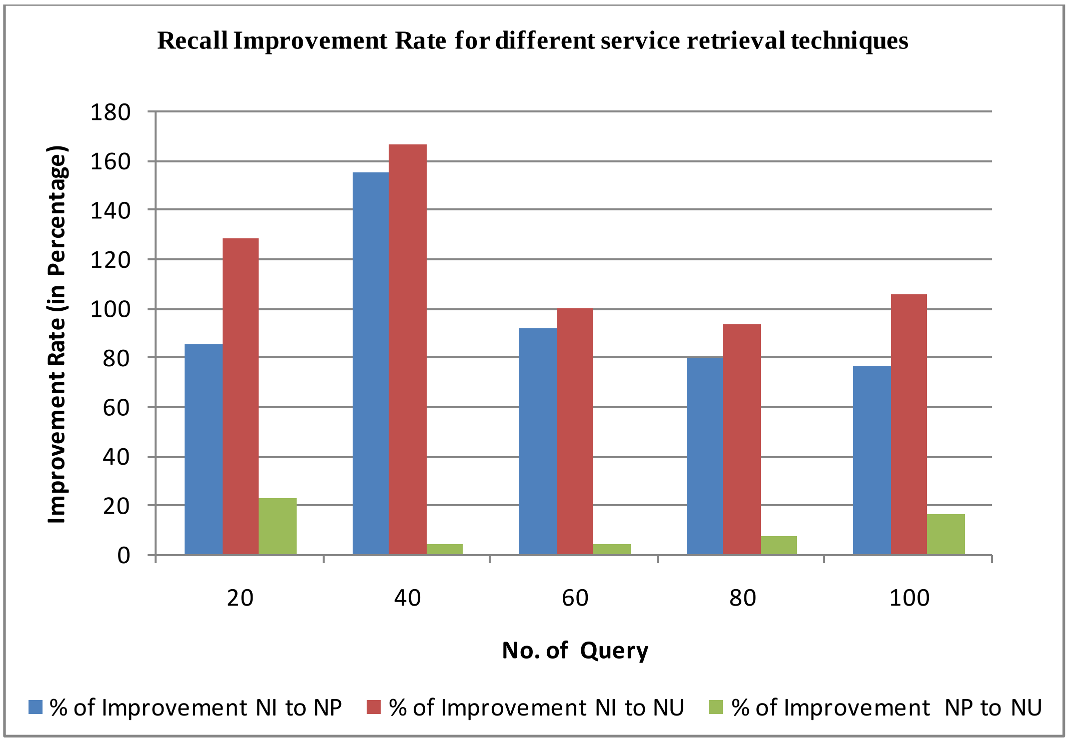

Figure 6.5 Recall improvement rate for different service retrieval techniques

Table 6.8 Quality measures analysed for F-Measure service retrieval techniques

Domain	Query	Azmeh (NE)	SSSM (NP)	UPM (NU)
Airline	20	0.4375	**0.684211**	**0.8**
	40	0.461538	**0.901961**	**0.941176**
	60	0.577778	**0.862069**	**0.912281**
	80	0.588235	**0.830769**	**0.892308**
	100	0.576271	**0.833333**	**0.921053**

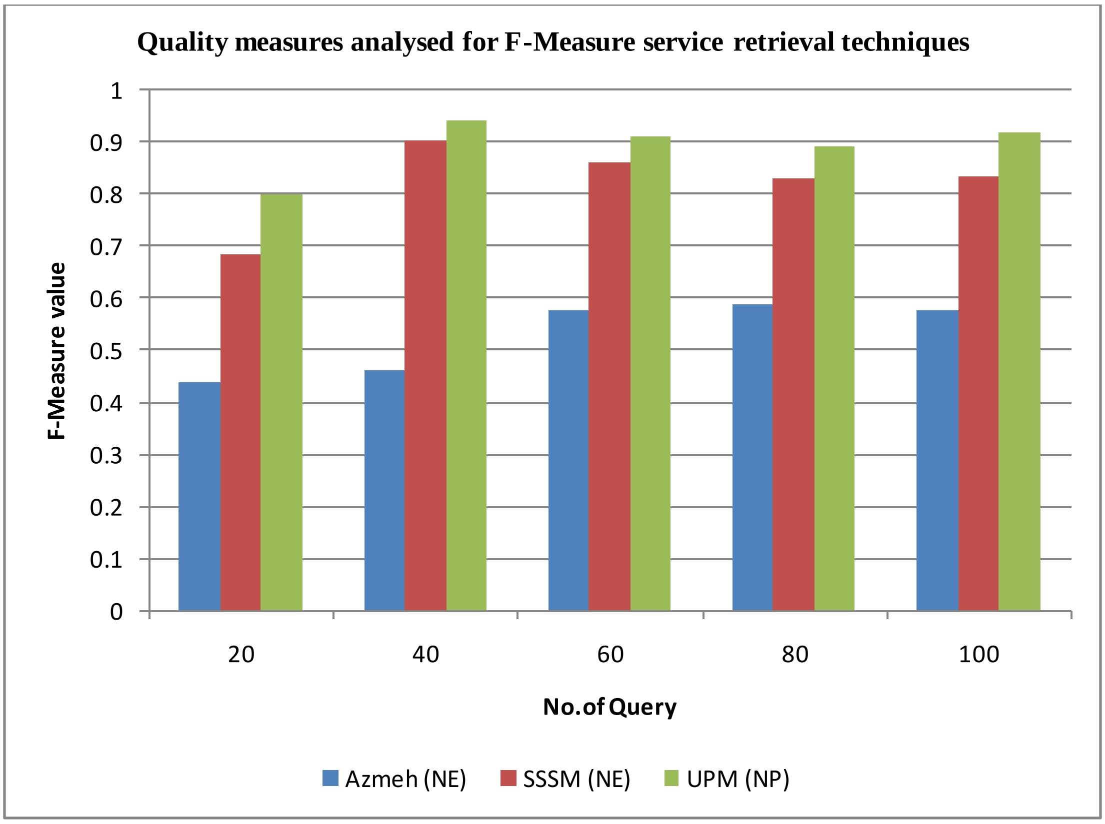

Figure 6.6 F-Measure for different service retrieval techniques

Table 6.9 Rate of F-Measure improvement for different service retrieval techniques

Domain	Query	% Improvement NI to NP	% Improvement NI to NU	% Improvement NP to NU
Airline	20	56.39098	82.85714	16.92308
	40	95.42484	103.9216	4.347826
	60	49.20424	57.89474	5.824561
	80	41.23077	51.69231	7.407407
	100	44.60784	59.82972	10.52632

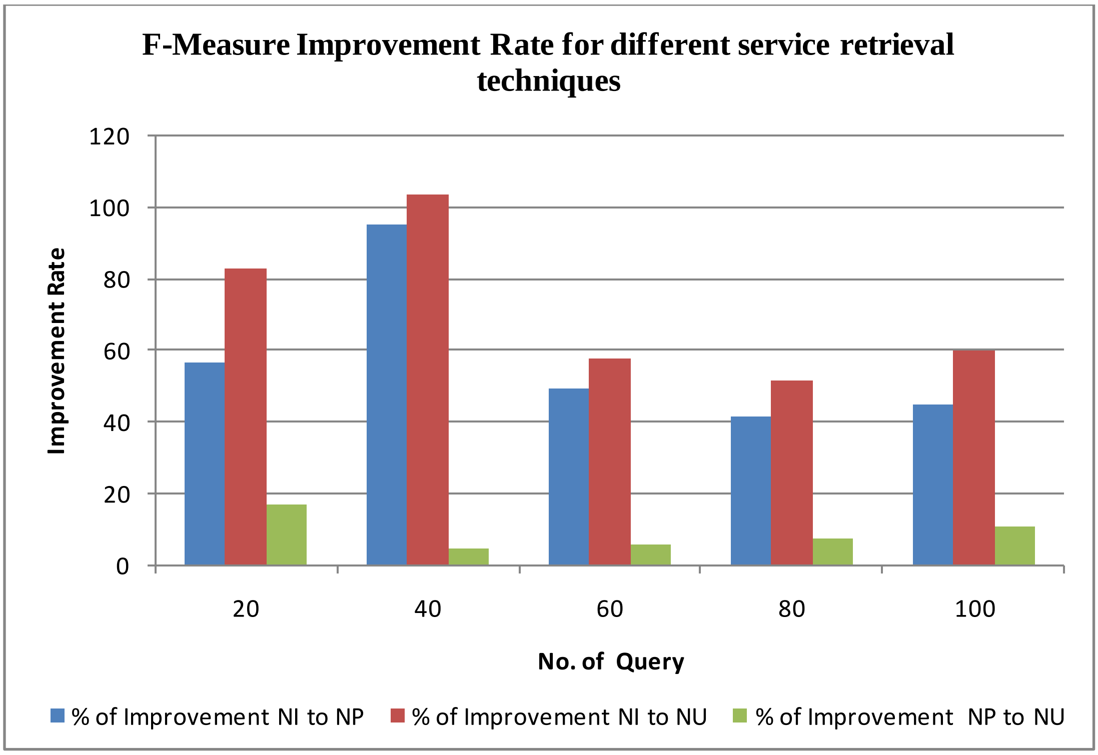

Figure 6.7 F-Measure improvement rate for different service retrieval techniques

This concludes that the performance of the proposed technique would not degrade with an increase in the varieties of queries raised on the UDDI. The comparison of different service retrieval techniques for the quality factors Precision, Recall and F-measure have been shown in the Table 6.4, 6.6 and 6.8respectively.The comparison of different service retrieval techniques rate of improvement for the quality factors Precision, Recall and F-measure have been shown in the table 6.5, 6.5 and 6.9 respectively.The comparison of different service retrieval techniques for the quality factors Precision, Recall and F-measure have been shown in the Figures 6.2, 6.4 and 6.6 respectively.The comparison of different service retrieval techniques rate of improvement for the quality factors Precision, Recall and F-measure have been shown in the Figures 6.3, 6.5 and 6.7 respectively.

To consolidate,the proposed framework is modelled for discovering user relevant services that have reduced the semantic gap between the services requested and the services discovered from the registry. The functional requirements of the service requested are mapped with the discovered services and the non-functional requirements are mapped to the QoS parameters of the services retrieved from the UDDI. Thus, the proposed clustering and the user preferential model has improved the efficiency of the service selection operation which has been shown using the three critical factors of the service retrieval discernment.

6.3 SUMMARY

The challenges faced when discovering user relevant services can be resolved by using the proposed similarity measure. The quality attributes can be resolved by using the proposed 2-tier user preference model. The experimental results depict the improved precision, recall and F-measure values of the proposed methods. Precision can be used as a measure for evaluating web service selection since it quantifies the number of services returned that are relevant. High precision indicates that the system has returned most number of the relevant documents from the retrieved set.Recall can be used as a measure for evaluating web service selection since it quantifies the number of relevant services returned over the number of services that are returned. It can be viewed as a measure of completeness. High recall indicates that the system has retrieved most of the relevant documents from the entire dataset.F-measure can be used as a measure for evaluating web service selection. The higher the value of F-measure,the more accurate the relevant web services discovered by the system. Thus this service selection model can be used to discover user desired services.

CHAPTER 7

EXPERIMENTATION AND RESULT ANALYSIS – PHASE - II

7.1 INTRODUCTION

In this chapter, the performance evaluation of the proposed algorithm has been carried out with respect to the standard assessment factors of the Genetic Algorithm. This assessment is devoted to validate the significances of the proposed algorithm alongside the recent and the best working models of the GA using various problem specific performance factors. The experimental setup has 1000 web services obtained from QWS Dataset [4]. The properties of the web services such as its name, operations, endpoints, bindings and QoS parameters including availability, response time, throughput, latency time and reliability are obtained by parsing the description of the web services. The available set of services has been classified into some 21 domains as given in the Chapter 6.

7.2 EXPERIMENTAL SETUP

The GA parameters and the corresponding values are depicted in the Table 7.1. Elitist strategy is followed to ensure that the fittest individuals in each generation are carried out to the next generation in order to avoid the replacement of best fit individuals with poor individuals in the successive generations. For each technique, the executions are carried out for 50 times and the average of each case has been considered for experimental analyses. For experiments, the crossover and mutation operators used are Greedy Crossover (GX) and Swap Mutation operator respectively.

Table 7.1 GA Configuration Parameters for Experiments

S. No	Parameter	Value / Technique
1	Population Size	100
2	Generation Limit	250
3	Initialization Technique	Random, Nearest Neighbor (NN), ODV-VV and Constraint Initialization
4	Crossover Method	Greedy crossover (GX), Self-Organized crossover
5	Crossover Probability	0.6
6	Mutation Method	Swap Mutation
7	Mutation Probability	0.02
8	Elitism	True (3 individuals)
9	Termination Condition	Generation Limit

7.3 ASSESSMENT CRITERIA

The evaluation of the proposed algorithm for web service selection has been performed with respect to the standard factors used to analyze the optimization capability of the different GA models considered in the experimentation process. The metrics for evaluation include computation time, individual quality, error rate, average convergence and convergence diversity are summarized below.

Computation Time: It is defined as the total time taken to generate the initial population with *'n'* individuals. This factor is used to measure the computational complexity of the population seeding technique and is directly proportional to the number of cities in the TSP instance.

Error Rate (%): It could be defined as the percentage of difference in the fitness value of the solution with the known optimal solution for the problem. It can be given as,

$$Error\ Rate\ (\%) = \frac{Fitness\ - Optimal\ Fitness}{Optimal\ \ Fitness} \times 10 \qquad (7.1)$$

This factor can be categorized into two types; individual with high error rate refers to the individuals with worst fitness value in the population. Similarly, individuals with least error rate are to be referred as the individuals with best fitness value in the population. The convergence rate is just the complement of the error rate and it can be represented as,

$$Convergence\ \ Rate\ (\%) = 1 - \frac{Error\ Rate(\%)}{100} \times 100 \quad (7.2)$$

Therefore, the result analysis is performed in the perspective of the error rate.

Average Convergence (%): It is defined as the average of the convergence rate of solutions in the initial population generated [Kaur et al 2008, Heikki et al 2007]. It can be given as,

$$Average\ \ Convergence\ (\%) = 1 - \frac{Average\ \ Fitness - Optimal\ Fitness}{Optimal\ \ Fitness} \times 100$$

(7.3)

where,

- $AverageFitness$ is the average fitness value of solutions in the population

- $Optimal\ Fitness$ is the known optimal value of the corresponding instance

This factor is used to measure the quality of the population generated as an average of the population as given in the Eqn. (7). An initial population with good average convergence can increase the performance in terms of convergence time and helps to explore the search space better. A significant gap should be maintained between average convergence and best & worst convergences of the population for better search space exploration.

Convergence *Diversity (%):* The convergence diversity of the population refers to the difference between the convergence rate of the best and worst individuals generated in the population. This can be represented as,

$$Convergence\ Diversity\ (\%) = CR_{Best\ Indiv} - CR_{Worst\ Indiv} \quad (7.4)$$

where,

- $CR_{Best\ Indiv}$ is the convergence rate of the best fitness individual in the population.
- $CR_{Worst\ Indiv}$ is the convergence rate of the worst fitness individual in the population.

In GA, the convergence diversity of the initial population plays a critical role to increase the chance of evolving optimal solutions and to avoid premature convergence. On the other hand, if the initial population lacks in diversity, it can explore only a small part of the search space and may never find the global optimal solution. This factor shows the variety of the individuals in the population, which is an important factor used to overcome the pre-mature convergence problem. Thus, the population of individuals with high convergence diversity is capable of exploring still more part in the search space.

7.4 RESULT ANALYSES

In this section, the effectiveness of the proposed GA model along with CGA-Random, CGA-NN and CGA-VV models under similar experimental setups is discussed with respect to the appropriate set of performance criteria as discussed in the Section 7.3. Experimental results of the CGA-Random, CGA-NN, CGA-VV and SOGA-CI models are shown in the Table 7.2, 7.3, 7.4 and 7.5 respectively. The respective analyses w.r.t. the performance factors are as follows:

Table 7.2 Experimental results of classical GA model with Radom initialization technique

S. No	Class of QoS factors	Services Considered	Optimal Solution	Computation Time	Quality of the Solution			Error Rate (%)		Convergence Diversity	Average Convergence
					Best	Worst	Average	Best	Worst		
1	Class - I	100	2080	1.23	2224.36	2665.19	2446.37	6.94	28.13	21.19	82.39
		500	2050	7.32	2222.44	2708.32	2630.33	8.41	32.11	23.70	71.69
		1000	2075	12.03	2297.14	2780.83	2533.44	10.71	34.02	23.31	77.91
2	Class - II	100	2010	0.98	2180.86	2405.84	2279.93	8.50	19.69	11.19	86.57
		500	2010	6.34	2264.52	2736.64	2701.89	12.66	36.15	23.49	65.58
		1000	2010	10.95	2287.35	2654.95	2533.76	13.80	32.09	18.29	73.94
3	Class - III	100	1935	0.92	2245.04	2617.56	2321.43	16.02	35.27	19.25	80.03
		500	1980	5.23	2327.42	2870.19	2799.16	17.55	44.96	27.41	58.63
		1000	2035	9.09	2382.51	2761.25	2633.43	17.08	35.69	18.61	70.59
4	Class - IV	100	1753	0.88	2115.55	2216.65	2132.28	20.68	26.45	5.77	78.36
		500	1862	4.68	2305.40	2610.76	2540.72	23.81	40.21	16.40	63.55
		1000	1965	8.76	2486.79	2690.52	2612.47	26.55	36.92	10.37	67.05
5	Class - V	100	1598	0.82	2070.41	2146.75	2140.10	29.56	34.34	4.78	66.08
		500	1706	3.55	2255.54	2347.12	2324.98	32.21	37.58	5.37	63.72
		1000	1792	7.04	2423.38	2654.88	2526.71	35.23	48.15	12.92	59.00
6	Class - VI	100	1421	0.69	1953.07	2157.73	2034.43	37.44	51.85	14.40	56.83
		500	1581	2.97	2231.25	2567.04	2434.35	41.13	62.37	21.24	46.02
		1000	1675	6.21	2405.72	2476.62	2429.92	43.63	47.86	4.23	54.93
7	Class - VII	100	1345	0.56	1969.91	2321.14	1996.60	46.46	72.58	26.11	51.55
		500	1456	2.67	2173.78	2484.87	2279.60	49.30	70.66	21.37	43.43
		1000	1563	5.01	2377.86	2663.97	2630.06	52.13	70.44	18.31	31.73
8	Class - VIII	100	1279	0.41	1982.07	2476.01	2045.29	54.97	93.59	38.62	40.09
		500	1456	1.92	2297.67	2756.82	2378.63	57.81	89.34	31.54	36.63
		1000	1568	3.59	2518.89	2637.78	2550.98	60.64	68.23	7.58	37.31
9	Class - IX	100	1185	0.34	1937.23	2295.93	2284.11	63.48	93.75	30.27	7.25
		500	1321	1.76	2197.03	2380.50	2292.73	66.32	80.20	13.89	26.44
		1000	1657	2.99	2802.85	3279.51	2915.01	69.15	97.92	28.77	24.08
10	Class - X	100	1158	0.3	1991.63	2290.21	2130.89	71.99	97.77	25.78	15.99
		500	1406	1.63	2458.04	2750.23	2586.73	74.83	95.61	20.78	16.02
		1000	1668	2.82	2963.39	3246.83	3090.92	77.66	94.65	16.99	14.69

Table 7.3 Experimental results of classical GA model with Nearest Neighbour initialization technique

S. No	Class of QoS factors	Services Considered	Optimal Solution	Computation Time	Quality of the Solution			Error Rate (%)		Convergence Diversity	Average Convergence
					Best	Worst	Average	Best	Worst		
1	Class - I	100	2080	2.51849225	2082.15	2097.97	2086.19	0.10	0.86	0.76	99.70
		500	2050	8.586195097	2069.59	2102.02	2082.68	0.96	2.54	1.58	98.41
		1000	2075	12.4415386	2171.13	2239.13	2190.73	4.63	7.91	3.28	94.42
2	Class - II	100	2010	2.369277382	2029.68	2090.18	2027.78	0.98	3.99	3.01	99.12
		500	2010	6.877470649	2093.91	2106.02	2085.22	4.17	4.78	0.60	96.26
		1000	2010	11.19170237	2109.47	2130.82	2112.65	4.95	6.01	1.06	94.89
3	Class - III	100	1935	1.764833851	2060.19	2143.03	2126.60	6.47	10.75	4.28	90.10
		500	1980	6.073329677	2038.76	2106.97	2054.42	2.97	6.41	3.44	96.24
		1000	2035	9.538084657	2180.58	2229.30	2205.45	7.15	9.55	2.39	91.62
4	Class - IV	100	1753	1.871209754	1919.53	1931.82	1914.96	9.50	10.20	0.70	90.76
		500	1862	5.932042121	2035.80	2047.13	2018.39	9.33	9.94	0.61	91.60
		1000	1965	9.761046993	2197.38	2253.39	2195.28	11.83	14.68	2.85	88.28
5	Class - V	100	1598	1.511549899	1801.10	1808.74	1791.46	12.71	13.19	0.48	87.89
		500	1706	3.73936144	1944.72	2002.98	1950.87	13.99	17.41	3.41	85.65
		1000	1792	7.51702413	2096.83	2163.73	2104.98	17.01	20.74	3.73	82.53
6	Class - VI	100	1421	1.629664066	1675.70	1718.07	1669.18	17.92	20.91	2.98	82.53
		500	1581	4.681559558	1896.95	1958.96	1920.41	19.98	23.91	3.92	78.53
		1000	1675	7.813840494	1995.10	2058.05	2039.83	19.11	22.87	3.76	78.22
7	Class - VII	100	1345	0.997268297	1624.51	1645.94	1632.61	20.78	22.37	1.59	78.62
		500	1456	4.14218872	1775.26	1799.52	1775.00	21.93	23.59	1.67	78.09
		1000	1563	5.365673326	1921.49	1974.86	1952.27	22.94	26.35	3.41	75.09
8	Class - VIII	100	1279	2.388031976	1587.11	1615.12	1609.04	24.09	26.28	2.19	74.20
		500	1456	2.564800635	1823.15	1860.95	1840.52	25.22	27.81	2.60	73.59
		1000	1568	3.981781256	1972.90	2030.10	2019.10	25.82	29.47	3.65	71.23
9	Class - IX	100	1185	1.643194775	1516.34	1546.29	1524.83	27.96	30.49	2.53	71.32
		500	1321	2.726291877	1704.43	1754.89	1729.52	29.03	32.85	3.82	69.08
		1000	1657	3.563369477	2152.58	2221.06	2208.06	29.91	34.04	4.13	66.74
10	Class - X	100	1158	2.274825397	1517.54	1558.16	1544.91	31.05	34.56	3.51	66.59
		500	1406	1.888597069	1856.10	1916.54	1889.05	32.01	36.31	4.30	65.64
		1000	1668	3.539040599	2225.24	2281.74	2257.30	33.41	36.79	3.39	64.67

Table 7.4 Experimental results of classical GA model with ODV-VV initialization technique

S. No	Class of QoS factors	Services Considered	Optimal Solution	Computation Time	Quality of the Solution			Error Rate (%)		Convergence Diversity	Average Convergence
					Best	Worst	Average	Best	Worst		
1	Class - I	100	2080	3.547123971	2102.05	2180.49	2121.75	1.06	4.83	3.77	97.99
		500	2050	11.74329345	2112.76	2195.01	2149.55	3.06	7.07	4.01	95.14
		1000	2075	14.20808684	2168.82	2188.64	2158.68	4.52	5.48	0.95	95.97
2	Class - II	100	2010	5.202383677	2134.28	2141.45	2129.60	6.18	6.54	0.36	94.05
		500	2010	8.595237297	2118.10	2134.66	2113.78	5.38	6.20	0.82	94.84
		1000	2010	13.7352888	2124.77	2173.29	2156.21	5.71	8.12	2.41	92.73
3	Class - III	100	1935	3.560587971	2050.23	2100.23	2051.62	5.96	8.54	2.58	93.97
		500	1980	7.542778669	2130.71	2149.90	2132.86	7.61	8.58	0.97	92.28
		1000	2035	10.8742846	2134.08	2175.00	2140.99	4.87	6.88	2.01	94.79
4	Class - IV	100	1753	3.641235107	1902.15	1910.21	1883.03	8.51	8.97	0.46	92.58
		500	1862	8.271750587	1945.19	1948.91	1934.55	4.47	4.67	0.20	96.10
		1000	1965	11.3827888	2127.12	2130.67	2114.43	8.25	8.43	0.18	92.40
5	Class - V	100	1598	4.155199843	1754.19	1815.78	1770.12	9.77	13.63	3.85	89.23
		500	1706	5.158988925	1929.95	2004.87	1957.03	13.13	17.52	4.39	85.29
		1000	1792	10.58774498	1991.01	2066.83	2004.78	11.11	15.34	4.23	88.13
6	Class - VI	100	1421	3.658918172	1631.26	1695.48	1630.59	14.80	19.32	4.52	85.25
		500	1581	6.87131277	1816.75	1851.66	1841.22	14.91	17.12	2.21	83.54
		1000	1675	10.33513029	1914.66	1953.53	1924.19	14.31	16.63	2.32	85.12
7	Class - VII	100	1345	3.080798776	1542.10	1592.78	1539.63	14.65	18.42	3.77	85.53
		500	1456	4.982731989	1672.48	1731.90	1692.77	14.87	18.95	4.08	83.74
		1000	1563	7.478463943	1817.61	1873.82	1848.43	16.29	19.89	3.60	81.74
8	Class - VIII	100	1279	4.12253483	1497.65	1548.48	1520.76	17.10	21.07	3.97	81.10
		500	1456	4.510682819	1708.35	1780.75	1736.45	17.33	22.30	4.97	80.74
		1000	1568	4.661088126	1846.82	1933.91	1865.22	17.78	23.34	5.55	81.04
9	Class - IX	100	1185	4.038516949	1407.01	1465.13	1420.71	18.73	23.64	4.90	80.11
		500	1321	4.596934458	1578.89	1642.37	1593.04	19.52	24.33	4.81	79.41
		1000	1657	5.670234755	1993.99	2069.38	2048.04	20.34	24.89	4.55	76.40
10	Class - X	100	1158	2.948821646	1400.43	1472.39	1421.81	20.94	27.15	6.21	77.22
		500	1406	4.430638218	1701.16	1802.52	1796.02	20.99	28.20	7.21	72.26
		1000	1668	5.115309136	2033.86	2140.79	2088.87	21.93	28.34	6.41	74.77

Table 7.5 Experimental results of classical GA model with Constraint initialization technique

S. No	Class of QoS factors	Services Considered	Optimal Solution	Computation Time	Quality of the Solution			Error Rate (%)		Convergence Diversity	Average Convergence
					Best	Worst	Average	Best	Worst		
1	Class - I	100	2080	3.339926848	2080.00	2080.00	2080.00	0.00	0.00	0.00	100.00
		500	2050	9.956718328	2050.00	2050.00	2050.00	0.00	0.00	0.00	100.00
		1000	2075	13.08624638	2091.16	2103.90	2092.54	0.78	1.39	0.61	99.15
2	Class - II	100	2010	3.942610639	2028.87	2074.99	2031.22	0.94	3.23	2.29	98.94
		500	2010	8.27491111	2032.81	2074.47	2057.26	1.13	3.21	2.07	97.65
		1000	2010	11.99643156	2033.23	2061.62	2042.54	1.16	2.57	1.41	98.38
3	Class - III	100	1935	2.856470211	1960.24	1975.12	1969.43	1.30	2.07	0.77	98.22
		500	1980	6.575009355	2037.10	2083.33	2034.25	2.88	5.22	2.34	97.26
		1000	2035	9.649625247	2061.69	2069.77	2055.58	1.31	1.71	0.40	98.99
4	Class - IV	100	1753	3.354413963	1835.41	1858.69	1833.45	4.70	6.03	1.33	95.41
		500	1862	6.31568966	1941.99	1958.24	1941.98	4.30	5.17	0.87	95.70
		1000	1965	11.34963493	2081.17	2121.70	2080.53	5.91	7.97	2.06	94.12
5	Class - V	100	1598	3.097548921	1687.80	1706.36	1691.53	5.62	6.78	1.16	94.15
		500	1706	4.358790601	1824.27	1874.53	1848.25	6.93	9.88	2.95	91.66
		1000	1792	9.361583018	1925.81	1965.36	1944.40	7.47	9.67	2.21	91.50
6	Class - VI	100	1421	3.423936223	1520.81	1537.43	1518.44	7.02	8.19	1.17	93.14
		500	1581	6.242409635	1737.69	1751.88	1740.15	9.91	10.81	0.90	89.93
		1000	1675	8.985105895	1823.61	1855.31	1830.73	8.87	10.77	1.89	90.70
7	Class - VII	100	1345	1.732240449	1472.20	1498.50	1481.32	9.46	11.41	1.96	89.86
		500	1456	4.152681677	1602.21	1631.60	1613.33	10.04	12.06	2.02	89.19
		1000	1563	6.852940274	1729.09	1761.63	1736.69	10.63	12.71	2.08	88.89
8	Class - VIII	100	1279	2.648847507	1422.39	1449.82	1425.76	11.21	13.36	2.14	88.53
		500	1456	3.025135732	1627.75	1659.89	1638.49	11.80	14.00	2.21	87.47
		1000	1568	4.273437475	1762.13	1797.73	1778.11	12.38	14.65	2.27	86.60
9	Class - IX	100	1185	2.306051636	1338.64	1366.29	1342.60	12.97	15.30	2.33	86.70
		500	1321	3.885654701	1500.00	1531.65	1503.92	13.55	15.95	2.40	86.15
		1000	1657	5.351526816	1891.21	1931.97	1898.14	14.13	16.59	2.46	85.45
10	Class - X	100	1158	2.40202826	1328.45	1357.66	1334.02	14.72	17.24	2.52	84.80
		500	1406	3.844311023	1621.18	1657.53	1627.88	15.30	17.89	2.59	84.22
		1000	1668	3.727332101	1933.03	1977.20	1949.00	15.89	18.54	2.65	83.15

7.4.1 Computation Time

The computation time of the model is the total time taken to select the services based on the QoS chosen by the users. The analysis based on computation time elucidates that the variant of GA which has the capability to derive the relevant services within the required span of execution. The Figure 7.1 shows the comparison on the performance of the different variants of the GA with respect to the computation time factor for the various classes and the numbers of test services. From the graph, it can be exemplified that the classical GA (CGA) with random initialization has the least computation time regardless of the class of QoS factors as well as the number of services considered. For most of the test settings, the CGA with NN technique perform better than the CGA with VV and the proposed SOGA model. The proposed model yields better computation time, for all the test instances, than the recent and the best working GA variant CGA with VV initialization technique.

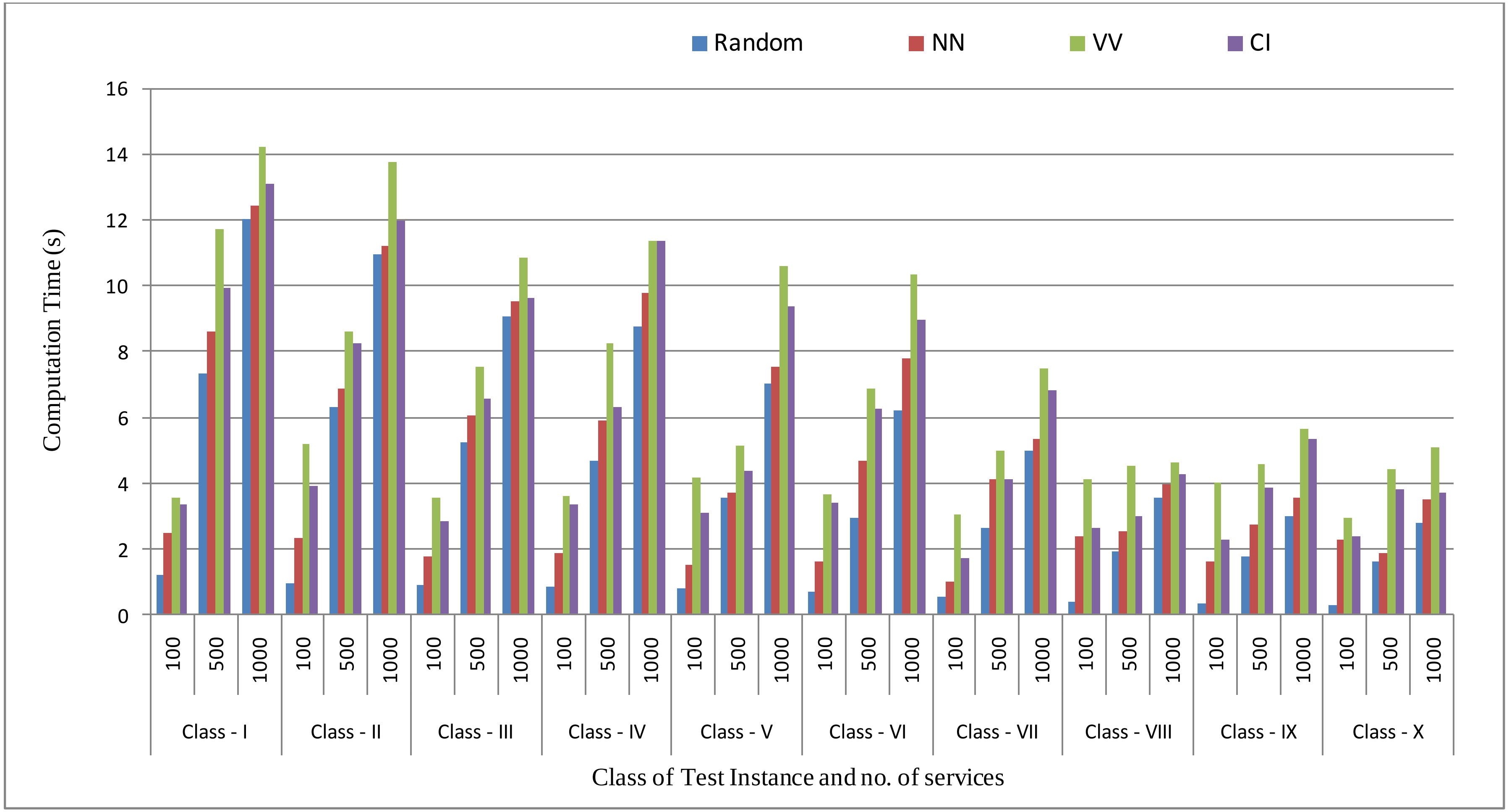

Figure 7.1. Computation Time based performance analyses on different GA Models

7.4.2 Quality

The quality of the individuals refers to the fitness value obtained by the individuals at the completion of the 200 generations. The quality of individual in the population can be assessed in two ways, the best, average and the worst quality; which refer to the highest, average and the lowest fitness value attained by an individual in the population respectively. The Figure 7.2 shows the comparison on the performance of the different variants of the GA with respect to the factor best quality individual in the population for the various classes and the numbers of test services. The graph can be interpreted that the technique with least fitness value produces the best quality individual. From the analyses, it can visualized that the proposed SOGA with constraint initialization outperform all the other competitive GA models for all the class of the QoS factors. The CGA with VV technique perform second best for all the test service instances, followed by the CGA with NN. The CGA with random initialization perform produces the individual with worst quality reflects that the technique is not suitable for web service selection operation for better relevant service retrieval. This, the proposed model yields the best quality individuals which can retrieve the relevant services based on the QoS requirement of the requestors irrespective of the number of QoS factors and the services in the UDDI.

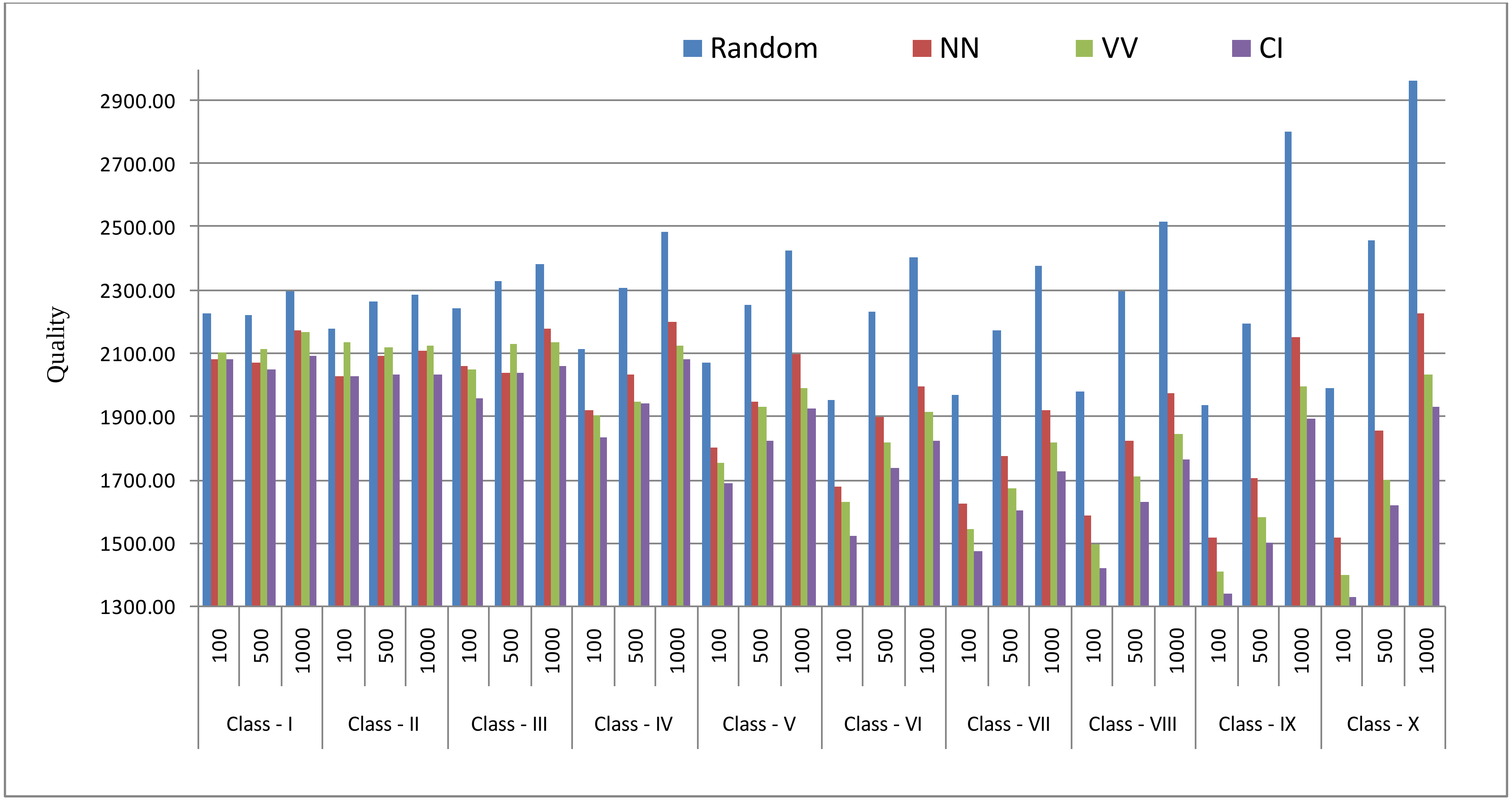

Figure.7.2. Quality of the individuals based performance analyses on different GA Models

7.4.3 Error Rate

The error rate of the individual shows the ability of the considered GA variants to produce the individual with the fitness value as much closer to the optimal value found for the particular test instance. This factor can be categorized into two types; individual with high error rate refers to the individuals with worst fitness value in the population. Similarly, individuals with least error rate are to be referred as the individuals with best fitness value in the population. The assessment on the performance of the considered GA variants based on the error rate factor for various test services has been shown in the Figure 7.3. From the graph, the following can be observed: For all the problem instances, the proposed SOGA with CI technique offers the least error rate consistently followed by the CGA-VV technique. The minimum error rate achieved by SOGA-CI and CGA-VV techniques is 0.02% and 1.06% respectively for the instance of 100 services with Class – I of QoS factors. The proposed method has obtained the optimal solution for 2/3 Class – I test services where as none of the other competitive techniques able to obtain the optimal solution for any of the instance. The maximum error rate obtained by the SOGA-CI is 15.89% for the Class – X of QoS factors and 1000 services UDDI, for which the other techniques CGA-random, CGA-NN and CGA-VV attained 77.66%, 33.41% and 21.93% respectively.

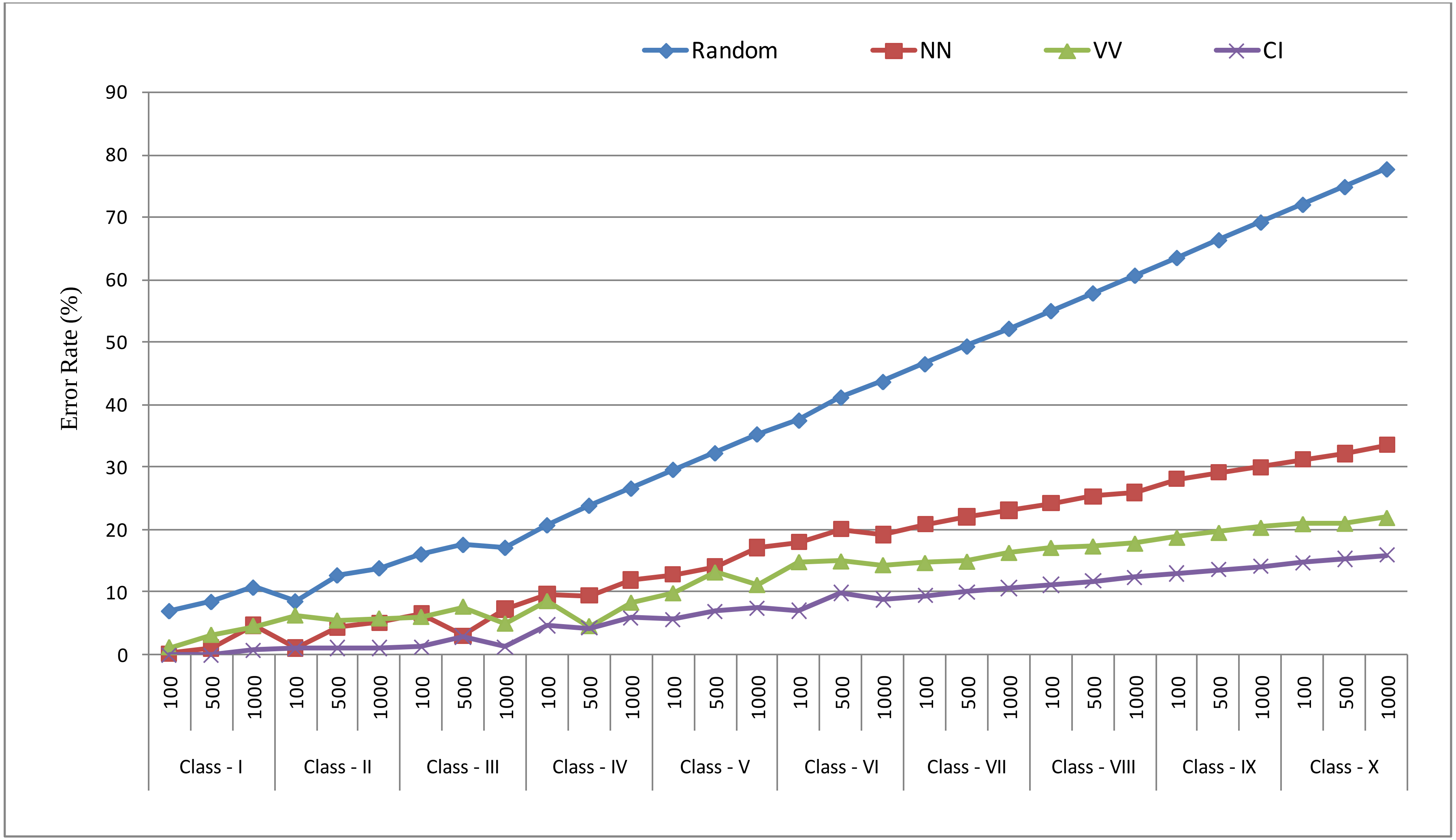

Figure7.3 Error Rate based performance analyses on different GA Models

7.4.4 Average Convergence

In GA, improvement in the average convergence depicts the progress towards the optimal point as a whole population instead of a single individual. This factor plays a vital role in identifying the ability of the GA variant to improve the fitness of the collection of individuals so that the better individuals can be formulated at the completion of the GA. The average convergence (%) obtained using different GA variants for various classes of QoS factors and service size is elucidated in Figure 7.4.From the figure 7.4, it can be perceived that the proposed SOGA-CI technique has the ability of better average convergence than that of the other GA variants considered for the experimentation. The CGA-VV outperforms the other two variants of the GA for all the varieties of the test service considered. The test service instance regardless of the QoS factor Class and the number of services in the UDDI, the capability to derive the relevant service, as a collection of population, based on the user request is found to be inherent characteristic of the proposed SOGA-CI model. This feature can enable the service selection system to retrieve the better collection of service from which user can choose the better suitable service in the way of the satisfaction of the provider as well the requestors.

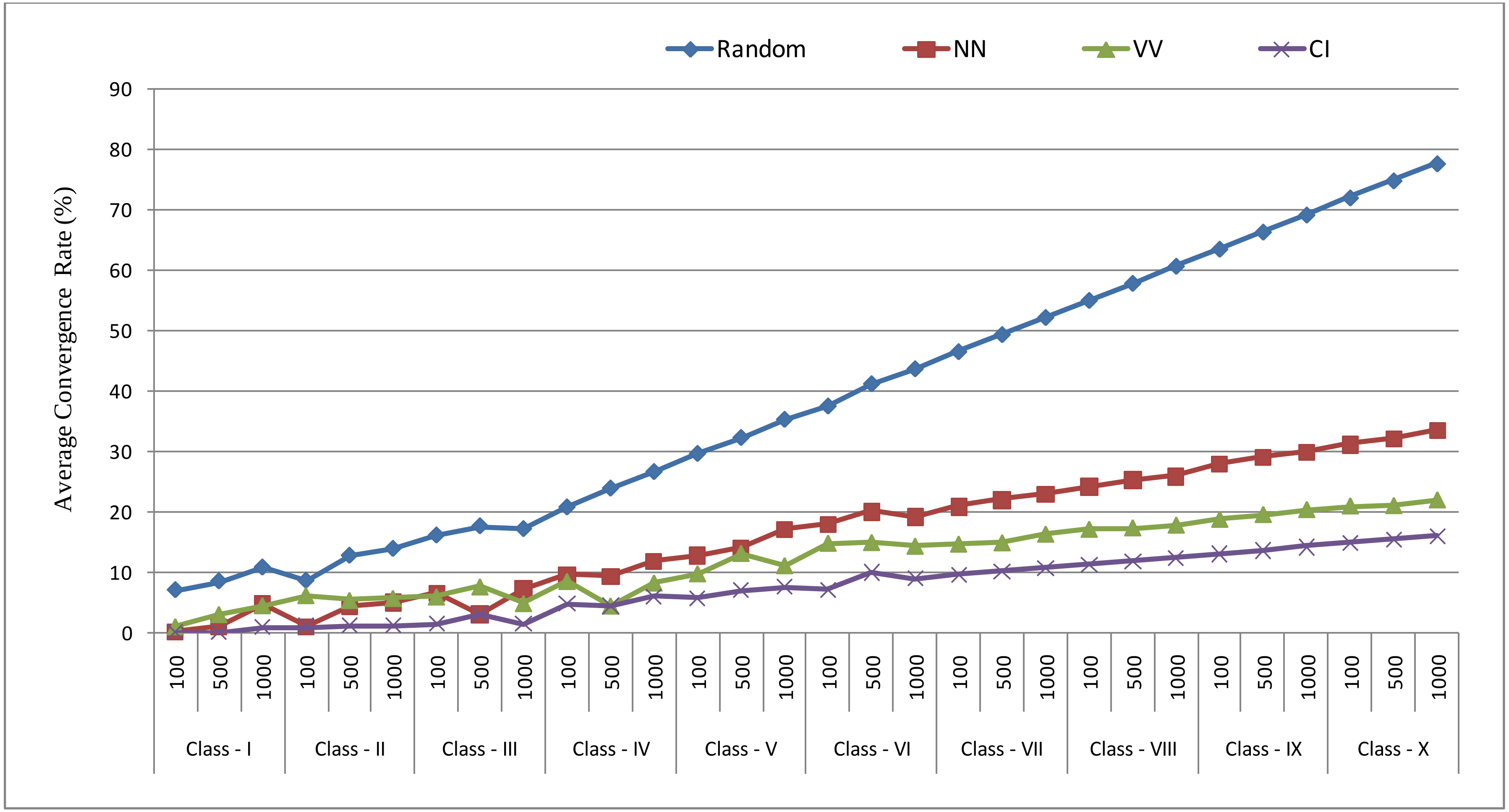

Figure 7.4 Average Convergence based performance analyses on different GA Models

7.4.5 Convergence Diversity

The convergence diversity shows the fitness based dispersion of the individuals in the population, which used to overcome the pre-mature convergence problem. It is the difference in the fitness of the best and the worst individuals produced by the GA variants for the given number of generations. .The assessment on the performance of the considered GA variants based on the convergence diversity factor for various test services has been shown in the Figure.7.5. From the graph, it can be witnessed that the CGA-random offer the maximum convergence diversity for all the test service instances which is distantly followed by the CGA-NN, CGA-VV and the proposed model. The performance of the later three models near similar though there is considerable difference in the performance with respect to the different class of QoS factors and service size. The value of the convergence diversity is expected to be the large at the start of the GA and the least towards the completion of the GA. On this perspective, the proposed model has the best valued convergence diversity when compared to that of the other models. The CGA-VV perform better than that of the CGA-NN, whereas the CGA-random offer the worst performance w.r.t the factor convergence diversity. Thus, the proposed model converge towards the optimal solution as a population rather than the individual which improves the chances of obtaining the better solution for any test instances.

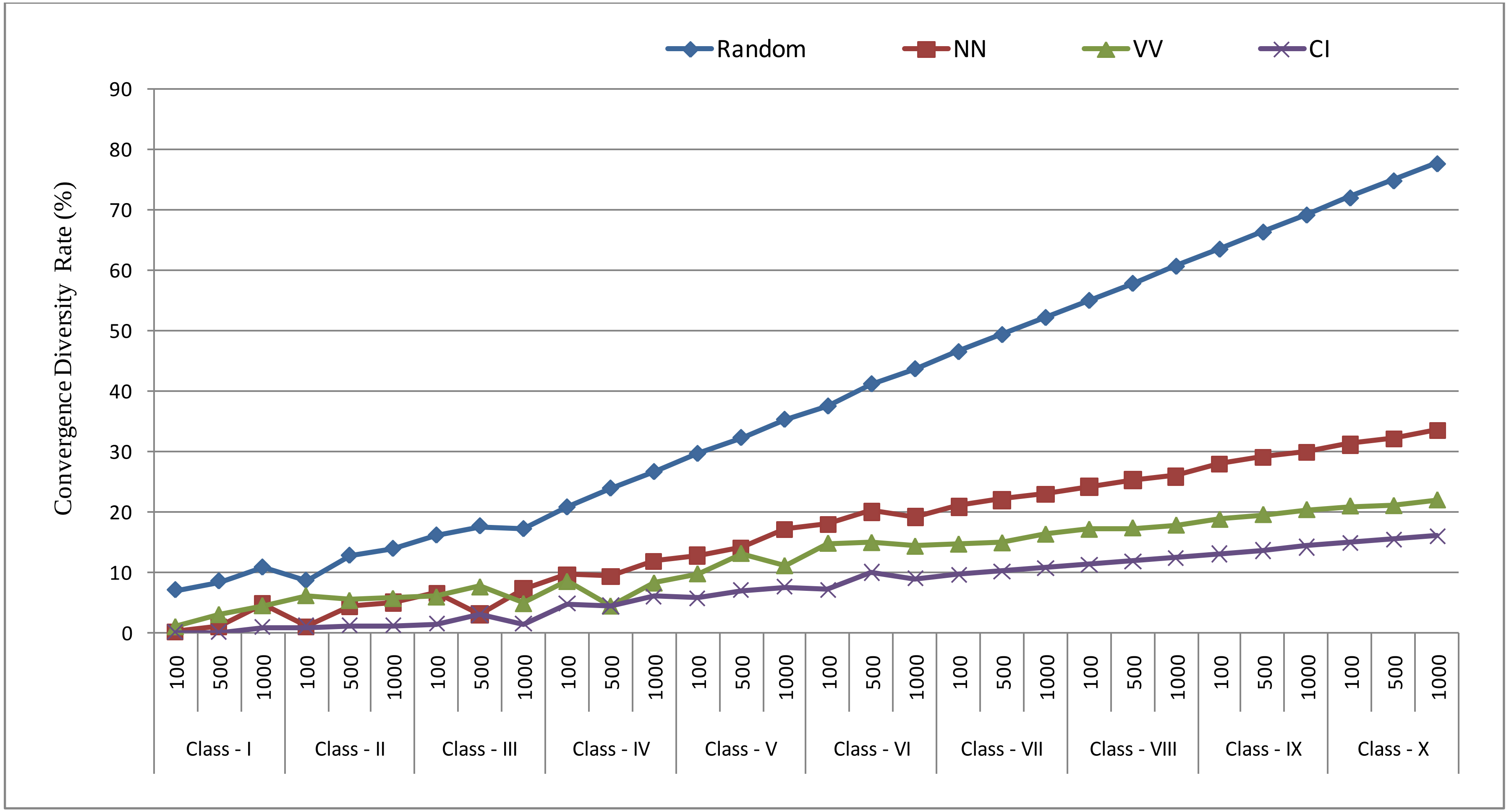

Figure7.5 Convergence Diversity based performance analyses on different GA Models

7.5 SUMMARY

In summary, the performance evaluation of the proposed algorithm has been performed in terms of the standard assessment factors of the Genetic Algorithm. This chapter of assessment aims to validate the importance of the proposed algorithm alongside the recent and the best working models of the GA using various problem specific performance factors. From the analyses, it can be justified that the proposed SOGA-CI model outperforms the existing and the best performing GA models on retrieving the services from the UDDI based on the various QoS factors opted by the service requestors. The other GA models considered CGI-VV, CGA-NN and CGA-random in the overall performance ranking order.

CHAPTER 8

CONCLUSION AND FUTURE ENHANCEMENT

8.1 CONCLUSION

The core of this research work is defined as to propose and develop an effective user context logic dependent integrated dynamic web service selection framework and optimize its services using Self-Organizing Genetic Algorithm model with Constraint Initialization. The Goals of this research are derived such that to design an enhanced model for Permutation-Coded Genetic Algorithm with effective population seeding technique to improve the performance in terms of precision, recall, f-measure, computation time, convergence rate, error rate, average convergence and convergence diversity. So, in principle, the goals of the proposed research have three folds as follows:

- To Design a Web Service Clustering Model that enables User Context Logic Dependent Integrated Dynamic Web Service Selection Framework.
- To Design and Develop a Self-Organized Genetic Algorithm to Optimize the proposed CO model.
- To Evaluate and Validate the performance of the Proposed Selection Framework using suitable Performance Assessment Factors.

The First goal is defined so as to develop a web service clustering model based user context logic method, and the second goal is defined so as to design an effective a Self-Organized Genetic Algorithm to Optimize the proposed web service selection model based Combinatorial Ooptimization model for improving the efficiency of the service retrieval operation. Third objective is to assess the performance of the proposed selection framework using suitable Performance Assessment Factors. All the goals are measureable and of course proved with appropriate set of experiments.

In the present research, a framework for semantic selection of services based on the properties clustered using concept lattice has been developed. A novel similarity measure for assessing the semantic relevance during service discovery is proposed to bridge the semantic gap between the service request and the service provided. A two tier User Context LogicModel (UCLM) is proposed to support the service discovery with respect to the non-functional requests. The Tier I of the UCLM deals with the qualification of the QoS parameters, where the user is presented with the available quality parameters for defining them in the model. The Tier II of the UCLM quantifies the qualified QoSparameters, where the user will set the preference values.

This research proposed an effective integrated web service selection framework that segregates the user specified QoS parameters as positive and negative based on their impact on the service utilization.Further, the optimal combinations of the parameters are identified using the novel self-organizing genetic algorithm model with constraint initialization technique. In order to evaluate the performance of the proposed model in both retrieval and algorithm parameters, performance evaluation of the proposed research has been done in two phases.

Through the Phase–I evaluations, it is demonstrated that the significances of the proposed work had been analyzed with respect to the different service retrieval assessment factors. This phase of assessment reveals that the proposed model, based on precision factor, obtained the performance improvement of the proposed work with respect to CGA-Rand, CGA-NN and CGA-VV are 51.35%, 53.06% and 25.56% respectively. Through the Phase–II assessments, it is demonstrated using the standard assessment factors of the Genetic Algorithm. This analysis justified that theproposed SOGA-CI model outperforms the existing and the best performing GA models on retrieving the services from the UDDI based on the various QoS factors opted by the service requestors.The constructive and encouraging results justified the effectiveness, significance and necessity of the proposed line of research and of course it may encourage further enhanced investigation in the identified area of research.

8.2 FUTURE ENHANCEMENT

This work outlines an enhanced service selection model that obtains the best optimal service in terms of the combination of QoS factors based on the service request's constraints. The proposed model had been proven for its significance with respect to the recent and the best working GA models studied in the literature. It can be noted that the research formulated the service selection operation as a combinatorial problem and also an effective GA model to solve it. This model concentrated on a novel GA with modified population initialization stage, whereas the crossover and mutation stages are kept unmodified. In this direction, the proposed work can be enhanced by proposing a GA model with modified crossover and mutation operator can certainly improve the performance, applicability and the associated qualities of the GA models towards the complex and multifarious service selection problem domain.

REFERENCES

1. AzmehZeina, HamouiFady, Huchard Marianne, UrtadoChristelle, "Backing Composite Web Services Using Formal Concept Analysis" International Symposium LIM – 2010.

2. AzmehZenia, Huchard Marianne, MessaiNizar, TibermacineChouki, "Using Concept Lattices to Support Web Services Composition with Backup Services", Publications du Laboratoired'Informatique de Robotiqueet de Microélectronique de Montpellier, lirmm-00533074, version 1 -5 Nov 2010.

3. AzmehZenia, Huchard Marianne, MessaiNizar, TibermacineChouki, "Many-Valued Concept Lattices for backing Composite Web Services", Publications du Laboratoired'Informatique de Robotiqueet de Microélectronique de Montpellier, lirmm-00557258, version 1 -18 Jan 2011.

4. Al-Masri, E., and Mahmoud, Q. H., "Discovering the best web service", (poster) 16th International Conference on World Wide Web (WWW), 2007, pp. 1257-1258.

5. Al-Masri, E., and Mahmoud, Q. H., "QoS-based Discovery and Ranking of Web Services", IEEE 16th International Conference on Computer Communications and Networks (ICCCN), 2007, pp. 529-534.

6. Aabhas V. Paliwal, ShafiqBasit, VaidyaJaideep, XiongHui, Adam Nabil, "Semantics-Based Automated Service Discovery", IEEE Transactions on Services Computing, Vol. 5, No. 2, April-June 2012.

7. AchrafKarray, RymTeyeb and Maher Ben Jemaa, "A Heuristic Approach for Web-Service Discovery and Selection", International Journal of Computer Science & Information Technology (IJCSIT), April 2013.

8. Box Don, Ehnebuske David, KakivayaGopal, Layman Andrew, Mendelsohn Noah, Frystyk Nielsen Henrik, ThatteSatish, Winer Dave, "Simple Object Access Protocol (SOAP) 1.1", http://www.w3.org/TR/2000/NOTE-SOAP-20000508/, W3C Note 08 May 2000.

9. BatraShalini, BawaSeema, "Web Service Categorization Using Normalized Similarity Score", International Journal of Computer Theory and Engineering, Vol. 2, No. 1, February 2010, 1793-8201.

10. Christensen Erik, Curbera Francisco, Meredith Greg, WeerawaranaSanjiva, "Web Services Description Language (WSDL)", http://www.w3.org/TR/wsdl, W3C Note 15 March 2001.

11. Chang, P. C., Huang, W. S., & Ting, C. J. (2010), "Dynamic diversity control in genetic algorithm for mining unsearched solution space in TSP problems", Expert Systems with Applications, 37(3), 1863–1878.

12. Cristina Bianca POP, VioricaRozina CHIFU, Ioan SALOMIE, Mihaela DINSOREANU Mihaly FODOR, Irina CONDOR," A Bee-inspired Approach

for Selecting the Optimal Service Composition Solution", International Conference on DEVELOPMENT AND APPLICATION SYSTEMS, suceava, Romania, Volume 6 Issue 4, May 2010,Pages 321-344.

13. Chen Zhou, Liang Tien Chia, Bu-Sung Lee "Service Discovery and Measurement based on DAML QoS Ontology" May 10–14, 2005, Chiba, Japan.

14. Chengying Mao, Jifu Chen, Dave Towey, Jinfu Chen, XiaoyuanXie, "Search-based QoS ranking prediction for web services in cloud environments", Future Generation Computer Systems (2015) Elsevier.

15. Christopher D. Manning, PrabhakarRaghavan and HinrichSchütze, "Introduction to Information Retrieval", Cambridge University Press, 2008.

16. Cilibrasi L. Rudi, Vitanyi Paul M.B,"The Google Similarity Distance", IEEE Transactions on Knowledge and Data Engineering, Vol.19, No.3, March 2007, 370-383.

17. DAML-S Coalition. DAML-S: Web Service Description for the Semantic Web. In Proc. International Semantic Web Conference (ISWC 02), 2002.

18. Demian.A.D'Mello and V.S.Ananthanarayana, "A QoS Model and Selection mechanism for QoS-Aware Web Services", Proceedings of the International Conference on Data Management (ICDM2008), 2008.

19. DivyaSachan, Saurabh Kumar Dixit, and Sandeep Kumar," QoS aware Formalized Model For Semantic Web Service Selection", International Journal of Web & Semantic Technology (IJWesT) Vol.5, No.4, October 2014, pp. 83-100.

20. Demian.A.D'Mello and V.S.Ananthanarayana, "A QoS Model and Selection mechanism for QoS-Aware Web Services", Proceedings of the International Conference on Data Management (ICDM2008), 2008.

21. D. Fensel, M. Kerrigan, and M. Zaremba, "Implementing Semantic Web Services", 2008, Berlin: Springer.

22. Erbin Lim, Philippe Thiran, ZakariaMaamar, Jamal Bentahar, "3-Way Satisfaction For Web Service Selection Preliminary Investigation", IEEE International Conference on Services Computing (SCC), 2011: 731-732.

23. Fiechter, L "A Parallel Tabu Search Algorithm for Large Travelling Salesman Problems". Discrete Applied Mathematics and Combinatorial Operations Research and Computer Science, 1994, 51 (3), 243–267.

24. Freisleben, B., Merz, P., 1996. "Genetic local search algorithm for solving symmetric and asymmetric traveling salesman problems", In: Proceedings of the IEEE Conference on Evolutionary Computation. IEEE Press, Nagoya, pp. 616–621.

25. George A. Miller (1995). WordNet: A Lexical Database for English. Communications of the ACM Vol. 38, No. 11: 39-41.

26. Gottschalk k, Graham S, Kreger H, Snell J, “Introduction to Web Services Architecture”, IBM Systems Journal, Vol. 41, No. 2, 2002.

27. GurugeAnura, “Web Services Theory and Practices”, ELSEVIER Digital Press, ISBN 1-5555-282-6, 2004.

28. Guofeng Chang “QoS-Based Web Service Selection Approach. Software Engineering and Knowledge Engineering”: Theory and Practice Advances in Intelligent and Soft Computing Volume 115, 2012, pp 887-892.

29. Gustavo Alonso, “Web Services: Concepts, Architectures and Applications”, Springer, 2013.

30. “Introduction to UDDI: Important Features and Functional Concepts”, http://www.oasis-open.org, October 2004.

31. JP Morgenthal, “Cloud Computing Realigns Role of Service Oriented Architecture”, https://www.infoq.com/articles/Cloud-Realigns-SOA, 2011.

32. Michael P.Poland, Christopher D. Nugent, Hui Wang, Liming Chen (2012), “Genetic algorithm and pure Rndom search for exosensor distribution optimisation”, Int. J. of Bio Inspired Computation, Inderscience, - Vol. 4, No.6, pp. 359-372.

33. Miguel Lurgi, David Robertson (2011), “Evolution in ecological agent systems”, Int. J. of Bio Inspired Computation, Inderscience, - Vol. 3, No.6, pp. 331-345.

34. Murat Albayrak and NovruzAllahverdi, "Development a new mutation operator to solve the Traveling Salesman Problem by aid of Genetic Algorithms", Expert Systems with Applications, Issue 38, July 2011, pp.1313-1320, ISSN: 0957-4174.

35. Molood Makhlughian, Seyyed Mohsen Hashemi, YousefRastegari and EmadPejman, “Web Service Selection Based on Ranking of QoS Using Associative Classification”, International Journal on Web Service Computing (IJWSC), Vol.3, No.1, March 2012, pp. 1-14.

36. M. Klusch and P. Kapahne., “Semantic Web Service Selection with SAWSDL-MX”. In Second International Workshop on Service Matchmaking and Resource Retrieval in the Semantic Web. 2008. Germany.

37. Murat Albayrak and NovruzAllahverdi, "Development a new mutation operator to solve the Traveling Salesman Problem by aid of Genetic Algorithms", Expert Systems with Applications, Issue 38, July 2011, pp.1313-1320, ISSN: 0957-4174.

38. OASIS, “Introduction to UDDI: Important Features and Functional Concepts”, http://www.oasis-open.org, October 2004.

39. Ouzzani, M.; Bouguettaya, A., “Efficient access to Web Service”, IEEE Journal, DOI: 10.1109/MIC.2004.1273484, Volume 8, Issue 2, 2004, pp.34-44.

40. Pradnya A. Khutade, RashmiPhalnikar, “QoS Aware Web Service Selection and Ranking Framework Based on Ontology”, International Journal of Soft Computing and Engineering (IJSCE), Volume-4 Issue-3, July 2014, pp. 77-81. ISSN: 2231-2307.

41. Potvin, J.Y., 1996. “Genetic algorithm for the traveling salesman problem”. Annals of Operations Research 63, 339–370.

42. R.D. van der Mei and H.B. Meeuwissen “Modelling End-to-end Quality-of-Service for Transaction-Based Services in Multi-Domain Environments”, In proceedings of IEEEInternational Conference on Web Services, pp.3 – 462, Washington, DC, USA, 2006.

43. Ray SS, Bandyopadhyay S, Pal SK (2004) ”New operators of genetic algorithms for traveling salesman problem”. Cambridge, UK, ICPR- 04 2:497–500.

44. SihemAmer-Yahia, YannisKotidis, “Web-Services Architecture for Efficient XML Data Exchange”, Proceedings of the 20th International Conference on Data Engineering, IEEE CONFERENCE, 2004.

45. SemyaElaoud, Jacques Teghem,TaıcirLoukil ,"Multiple crossover genetic algorithm for the multiobjective traveling salesman problem", Electronic Notes in Discrete Mathematics, Elsevier, Issue 36, pp.939-946, May 2010, ISSN:1571-0653.

46. T.Rajendran, P.Balasubramanie, ResmiCherian, “An Efficient WS-QoS Broker Based Architecture for Web Services Selection”, International Journal of Computer Applications, Volume 1 – No. 9, 2010, pp: 79-84. ISSN: 0975 – 8887.

47. T. Geetha, “An Optimistic Web Service Selection using Multi Colony –Particle Swarm Optimization (MC – PSO) algorithm”, International Journal of Emerging Technology and Advanced Engineering, Volume 3, Issue 8, August 2013, ISSN 2250-2459.

48. Tsai CF, Tsai CW, Yang T, “A modified multiple-searching method to genetic algorithms for solving traveling salesman problem”. In: IEEE international conference on systems, Man and cybernetics, 2002, vol. 3, pp 6–9.

49. U.S. Manikrao and T.V. Prabhakar, “Dynamic selection of web services with recommendation system”, 2005. Seoul, Korea, Republic of: Inst. of Elec. and Elec. Eng. Computer Society.

50. W3C Web Services Architecture Working Group, “Web Service Architecture”, Available at http://www.w3.org/TR/ws-arch/

51. Wenbin Wang, Qibo Sun, Xinchao Zhao, Fangchun Yang, “An improved Particle Swarm Optimization Algorithm for QoS-aware Web Service Selection in Service Oriented Communication”, International Journal of Computational Intelligence Systems, Suppl. 1 December, 2010.

52. XIE Ling-li, CHEN Fu-zan, KOU Ji-song, “Ontology-Based Semantic Web Service Clustering”, 978-1-61284-449-7/11 IEEE-2011.

53. Xiao Zheng, Jun-Zhou Luo, Ai-Bo Song, “Ant Colony System Based Algorithm for QoS-Aware Web Service Selection” Conference on Grid Service Engineering and Management, 2007.

54. Xiao Zheng, QingyongWu,” Exploring Bio-inspired Algorithm for Service Discovery in Pervasive Environments”, Computer Science and Information Technology (ICCSIT), 2010.

55. YongshengDing ,Hongbin Sun, KuangrongHao,” A bio-inspired emergent system for intelligent Web service composition and management”, Knowledge-Based Systems,Eldevier, Volume 20, Issue 5, June 2007, Pages 457–46.

56. Y. Yue, X. L. Wang, and A. Y. Zhou, “Underlying techniques for web services: a survey,” in proceedings, Journal of Software, vol. 15, No. 3, 2004, pp. 428-442.

57. Zachariasen M, Dam M, “Tabu search on the geometric traveling salesman problem”, In: Proc. of intconf on metaheuristics, 1995, pp 571–587.

LIST OF PUBLICATIONS

INTERNATIONAL JOURNALS

1. Thirumaran. M, Janani. M, Dhavachelvan. P, **Balaji. N**, SaleemBasha. M.S, "Business Logic Evaluation Model Using Dependency Analysis Approach for Service Integration", American Journal of Applied Science, (Science Publications), pp.1241-1249, 2014. ISSN: 1554-3641. **(Science Publications) Impact Factor: 1.4.**

2. **N. Balaji**, G. Sambasivam, S.R. Murugaiyan, M.S. SaleemBasha, T. Vengattaraman and P. Dhavachelvan, "Appraisal and Analysis on Diversified Web Service Selection Techniques based on QoS Factors", International Journal of Engineering and Technology, Vol. 5 No. 3, pp.3012 – 3019, Jun-Jul 2013. ISSN: 0975-4024. **Indexed in Scopus.**

3. **N. Balaji**, RenukaSruthiThorati, M.S. SaleemBasha and P. Dhavachelvan, "An Efficient Web Service Selection Technique using QoS based Service Level Agreement", International Journal of Applied Engineering Research, Volume 8, Number 9 (2013) pp. 983-1001, 2013. ISSN 0973-4562. **Indexed in Scopus.**

4. **N. Balaji**, S.R. Murugaiyan, M.S. SaleemBasha, R. Baskaran and P. Dhavachelvan, "Enhancements for UDDI using User Preferential Web Service Selection Model based on SLA", International Journal of Engineering and Technology, Vol. 5, No. 5, pp.4057 – 4067, Oct-Nov 2013. ISSN: 0975-4024. **Indexed in Scopus.**

5. M. Rajeswari, G. Sambasivam, **N. Balaji**, M.S. SaleemBasha, T. Vengattaraman and P.Dhavachelvan, "Appraisal and Analysis on Various Web Service Composition Approaches based on QoS Factors". Journal of King Saud University – Computer and Information Sciences (2013), ISSN: 1319-1578. **Published in Elsevier (ScienceDirect).**

6. M. Priyadharshini, R. Baskaran, **N. Balaji**, and M.S. SaleemBasha, "Analysis on Countering XML-based Attacks in Web Services", International Review on Computers and Software", Vol. 8 No. 9, pp.2197-2204, September 2013. ISSN: 1828-6003. **Indexed in Scopus.**

INTERNATIONAL CONFERENCES

1. **N. Balaji**, G. Sambasivam, M.S. SaleemBasha, T. Vengattraman and P. Dhavachelvan, “SLA based Architecture for Web Service Selection and Ranking with QoS” 2nd International Conference on Human Resource Interactions, in Saveetha University - Chennai, India, ICHCI 2013, 23-24 August 2013. **IEEE Xplore.**

2. **N. Balaji** and M.S. SaleemBasha, “A Survey on Diversified Web Service Selection Algorithms based on QoS Factors” 2nd International Conference on Advanced Computing Methodologies, in GRIET - Hyderabad, India, 2 – 3 August 2013, ICACM 2013, pp 56–62, 2013. ISBN: 9789351071495. **Elsevier Science Technology.**