

**OPTIMIZATION OF RANKING ALGORITHM IN THE
INFORMATION RETRIEVAL PROCESS USING
EVOLUTIONARY COMPUTATION**

A Thesis Submitted

IN FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF

**DOCTOR OF PHILOSOPHY
IN
COMPUTER SCIENCE AND ENGINEERING**

By
**SHADAB IRFAN
(15SCSE301001)**

Supervisor
Dr. D. RAJESH KUMAR
Associate Professor

**SCHOOL OF COMPUTING SCIENCE & ENGINEERING
Galgotias University, Greater Noida, Uttar Pradesh, INDIA**



**GALGOTIAS UNIVERSITY
UTTAR PRADESH
(June, 2021)**

CANDIDATE’S DECLARATION

I hereby certify that the work which is being presented in the thesis, entitled **“Optimization of Ranking Algorithm in the Information Retrieval Process Using Evolutionary Computation”** in fulfillment of the requirements for the award of the degree of Doctor of Philosophy in Faculty and submitted in Galgotias University, Greater Noida is an authentic record of my own work carried out during a period from **2015-2021** under the supervision of **Dr. D. Rajesh Kumar**.

The matter embodied in this thesis has not been submitted by me for the award of any other degree of this or any other University/Institute.



(SHADAB IRFAN)

This is to certify that the above statement made by the candidate is correct to the best of our knowledge.

(Dr.D.Rajesh Kumar)

Supervisor
SCSE

(Dr. _____)

Co-Supervisor
SCSE

The Ph.D. Viva-Voice examination of Shadab Irfan Research Scholar, has been held on _____.

Sign. of Supervisor(s)

Sign. of Co-Supervisor(s)

Sign. of External Examiner

ABSTRACT

In today's era, with the increase of digital information at a rapid pace, there has been a tremendous change in the World Wide Web which urges the researchers to find different ways to manage this enormous flow of information and help in satisfying the need of the user. It has been observed that the core component of any search engine is the ranking framework that helps to improve the result quality by ranking the web pages based on user queries. A good ranking model should incorporate diverse measures that ameliorate the result quality and should not stick to a single measure. In the Information Retrieval process, the process of ranking can be enhanced by incorporating process of content similarity and link analysis both. In coming years, it has been observed that various nature-inspired ranking algorithm can be used which integrate different measures and help to optimize the quality of the result obtained. Search techniques that are inspired by nature are referred as evolutionary algorithms and are regarded as a population-based stochastic algorithm. The development of various approaches helps in retrieving the information efficiently and in this regard nature-inspired algorithms like Swarm Intelligence, Genetic Algorithm play a major role. In ranking, we associate web pages stored in the repository in order of preference, so that the pages which have high similarity index according to the user query are ranked in higher-order in comparison to other web pages. For any search engine, ranking algorithm is considered as an indispensable part which overall effect the processing result. In minimum possible time, the most beneficial results should be displayed by the ranking algorithm. The intention of my research work is to optimize the ranking algorithm using different measures that will facilitate the Information Retrieval process thereby help the user in retrieving the relevant information in minimum time duration.

It has been observed that to perform ranking the huge number of pages in the corpus has to be processed which occupy maximum time for computation and retrieval and thus lower the speed and time of retrieval. Clustering or grouping of web pages based on certain conditions helps to make small clusters. This grouping of web pages will help in performing computation easily and enhance the retrieval rate. With the flow of online streaming of data at an exponential rate, the traditional Informational Retrieval techniques are becoming inadequate for handling the large volume of data so better strategies can be adopted which can also include some nature- inspired

algorithms that assist in optimizing the process thereby enhances the retrieval effectiveness. A simple query issued by user generates loads of web pages having both relevant as well as irrelevant content. It has been observed that in some cases irrelevant pages are given more importance and thus ranked higher in the hierarchy providing inaccurate results. Search engines normally use a content-based approach that cannot be considered a single criteria that help in ranking the web pages as the content can be manipulated to enhance the ranking of the web page. Similarly, for ranking web pages focusing only on the link-based approach is also not appropriate as most of the ranking algorithms use only a link-based approach as sometimes links are induced only for navigational purposes or to increase the weightage of a page. In this regard hybrid approach can be used for enhancing efficiency.

The research aims to ameliorate the competence of the ranking algorithms that help the user in ranking their web pages according to the query generated efficiently in less time-bound. An approach has been proposed for ranking the retrieved documents where the web pages are initially grouped on basis of certain condition so that irrelevant pages can be removed during the ranking process which assists in minimizing the computational complexity of the process and help in retrieving page in less time. Using nature-inspired algorithms like the Genetic Algorithm and Artificial Bee Colony a novel technique is proposed to optimize the ranking algorithm by ranking the web pages. To incorporate the hybrid formulation of the content-approach and structural approach, a new ranking algorithm is devised that not only consider the density of terms of the user query, but also handle the link structure among the web pages. The hybrid approach helps in ranking the web pages more efficiently.

It has been observed that using the proposed approach the iterations required to achieve the target is minimized tremendously. As the number of iterations is decreased, it not only helps in lessening the process computational complexity but also helps in achieving the result in less time-bound. The proposed ranking algorithm while using certain clauses, remove irrelevant pages during the initial phase of the ranking process, which overall reduces the load. It has been found that while comparing the algorithm with other standard algorithms like PageRank an improvement in mean average precision for the proposed approach is obtained. By using various evaluations measure for accessing the ranking quality it has been observed that by using the nature-inspired optimization technique for ranking the web pages, it helps in increasing the Normalized Discounted Cumulative Gain value.

ACKNOWLEDGEMENT

First, I would like to thank Almighty who is most beneficent, the most merciful in helping me at each step for guiding me and encouraging me in times of great disappointment.

At this time of accomplishment I am greatly indebted to my research guide, and want to express my sincere thanks and gratitude to Dr. D. Rajesh Kumar, Associate Professor, School of Computing Science & Engineering, Galgotias University, Greater Noida, for his guidance, patience, motivation, and immense knowledge. His constant guidance, inspiration, and critics are sources of innovative ideas. He is the main source behind the successful completion of this dissertation. The trust shown by him was the biggest source of inspiration for me. I am very thankful for his sincere and kind advice and his constant dedication and availability to me.

I must owe a special debt of gratitude to Hon'ble Chancellor Mr. Suneel Galgotia, Mr. Dhruv Galgotia, CEO, and Hon'ble Vice-Chancellor, Galgotias University for their valuable cooperation.

I express my sincere thanks to the Dean, School of Computing Science & Engineering, Dean Research, Research co-ordinator for his guidance and moral support during my research work and all faculties of School of Computing Science & Engineering who helped me a lot in my course of research work and all those who stand behind me.

Nothing is possible without the constant support of the family. My parents are my pillars of strength and especially my Father who is the guiding source and my greatest strength who stands beside me throughout my work and without his constant support and blessings, this work would not be able to complete. The consistent support and encouragement of my aunt, my sister and a deep sense of appreciation for one of my family member Gaurav who helped me a lot in compiling my thesis. My final thanks to all my friends, and to those who stand behind me like a support and helped me in completing this dissertation.

TABLE OF CONTENTS

Title Page	i
Candidate Declaration	ii
Abstract	iii
Acknowledgement	v
Table of Contents	vi
List of Figures	xi
List of Tables	xv
List of Publications from the Thesis	xviii
List of Abbreviations and Symbols	xx
Chapter 1: Introduction	1
1.1 Motivation	2
1.2 Problem Statement	3
1.3 Relevance of The Work	4
1.4 Outline of the Thesis	6
1.5 Summary	7
Chapter 2: Literature Review	8
2.1 Information Retrieval	8
2.2 Information Retrieval and Web search	10
2.3 Challenges faced in Web Information Retrieval	11
2.4 Information Retrieval Process	12
2.5 Information Retrieval Models	13
2.5.1 Boolean model	14
2.5.2 Vector space model	14
2.5.3 Probabilistic model	16
2.5.4 2-Poisson Model	16

2.5.5 Bayesian Network Model	17
2.5.6 Language Model	17
2.5.7 Google's Page Rank Model	18
2.6 Web Mining	19
2.7 Web Mining Process	21
2.7.1 Web Content Mining	22
2.7.2 Web Structure Mining	23
2.7.3 Web Usage Mining	24
2.8 Document Clustering and Similarity Measure	26
2.9 Ranking Algorithm	27
2.10 Connectivity-Based Ranking	28
2.10.1 Query-Independent Ranking	30
2.10.2 Query-Dependent Ranking	30
2.11 Various Ranking Algorithms	30
2.11.1 Page Rank Algorithm	30
2.11.2 Weighted Page Rank Algorithm	32
2.11.3 HITS Algorithm	33
2.11.4 Weighted Link Rank Algorithm	34
2.11.5 EigenRumor Algorithm	34
2.11.6 Distance Rank Algorithm	35
2.11.7 Time Rank Algorithm	36
2.11.8 TagRank Algorithm	36
2.11.9 Relation Based Algorithm	37
2.11.10 Query Dependent Ranking Algorithm	37
2.11.11 Ranking and Suggestive Algorithm	38
2.11.12 Comparison Score Based Algorithm	38
2.11.13 Algorithm for Query Processing in Uncertain Databases	38
2.12 Performance Evaluation Measures	40
2.12.1 Precision	40
2.12.2 Recall	41
2.12.3 Harmonic Mean (F-Measure)	42

2.12.4 Average Precision	42
2.12.5 Mean Average Precision	43
2.12.6 Cumulative Gain	43
2.12.7 Discounted Cumulative Gain	43
2.12.8 Normalized Discounted Cumulative Gain	44
2.13 Optimization Techniques	45
2.13.1 Nature Inspired Algorithm	45
2.13.1.1 Evolutionary Algorithm	48
2.13.1.1.1 Genetic Algorithm	49
2.13.1.2 Swarm Algorithm	52
2.13.1.2.1 Artificial Bee Colony	52
2.14 Summary	54

Chapter 3: Design and Implementation of scRank Algorithm for Ranking Web pages 55

3.1 Introduction	55
3.2 Related Work	55
3.3 Proposed Methodology	57
3.3.1 Architecture	58
3.3.2 Algorithm	58
3.3.3 Dataset Used	60
3.3.4 Experimental Analysis	61
3.4 Result Analysis	71
3.5 Summary	78

Chapter 4: Design and Implementation of gRank Algorithm using Genetic Algorithm 79

4.1 Introduction	79
4.2 Related Work	79
4.3 Genetic Algorithm	81
4.4 Proposed Methodology	82

4.4.1 Architecture	82
4.4.2 Algorithm	83
4.4.3 Dataset Used	84
4.4.3 Experimental Analysis	86
4.5 Result Analysis	94
4.6 Summary	101

Chapter 5: Design and Implementation of BeeRank Algorithm using Artificial Bee Colony Algorithm **102**

5.1 Introduction	102
5.2 Related Work	103
5.3 Artificial Bee Colony	105
5.4 Proposed Methodology	107
5.4.1 Architecture	107
5.4.2 Algorithm	109
5.4.3 Dataset Used	111
5.4.4 Experimental Analysis	113
5.5 Result Analysis	124
5.6 Summary	131

Chapter 6: Performance Evaluation and Result Analysis of scRank, gRank and BeeRank **132**

6.1 Introduction	132
6.2 Performance Evaluation Measures	133
6.3 Performance Analysis of Proposed Algorithms	134
6.4 Summary	142

Chapter 7: Conclusion and Future Scope **144**

7.1 Conclusion	145
7.2 Future Scope	146
References	147
Appendices	158
Author's Bio-Data	168

LIST OF FIGURES

Figure Name	Page No.
Figure 2.1 IRS Structure	9
Figure 2.2 Information retrieval processes	12
Figure 2.3 The Bowtie Structure of the Web	29
Figure 2.4 EigenRumor Community Model	35
Figure 2.5 Nature Inspired Algorithms	46
Figure 3.1 Proposed Architectural Model for scRank	58
Figure.3.2. Dataset of web pages	60
Figure.3.3. Queries tested on web pages	61
Figure 3.4 Hyperlink Structure of web pages	61
Figure 3.5(a) Execution time of scRank and PageRank	71
Figure 3.5(b) Bar Graph of Execution time for scRank and PageRank	71
Figure 3.6 Number of iterations for scRank and PageRank	72
Figure 3.7(a) Result of iterations in scRank	72
Figure 3.7(b) Result of iterations in PageRank	73
Figure 3.8 Precision values for scRank and PageRank	73
Figure 3.9 Recall values for scRank and PageRank	74
Figure 3.10 Average Precision for scRank and PageRank	74
Figure 3.11 Mean Average Precision for scRank and PageRank	75

Figure 3.12 Cumulative Gain for scRank and PageRank	76
Figure 3.13 Discounted Cumulative Gain for scRank and PageRank	76
Figure 3.14 Idealized Discounted Cumulative Gain for scRank and PageRank	77
Figure 3.15 Normalized Discounted Cumulative Gain for scRank and PageRank	77
Figure 4.1 Proposed Architectural Model for gRank	82
Figure 4.2 Dataset of web pages	85
Figure 4.3 Queries tested on web pages	85
Figure 4.4 Hyperlink Structure of web pages	86
Figure 4.5 Representations of Fitness Values	90
Figure 4.6 Ranking of Nodes in gRank Model	90
Figure 4.7(a) Execution time of gRank and PageRank	94
Figure 4.7(b) Bar Graph of Execution time for gRank and PageRank	95
Figure 4.8 Number of iterations for gRank and PageRank	95
Figure 4.9(a) Result of iterations in gRank	96
Figure 4.9(b) Result of iterations in PageRank	96
Figure 4.10 Precision values for gRank and PageRank	97
Figure 4.11 Recall value for gRank and PageRank	97
Figure 4.12 Average Precision for gRank and PageRank	98
Figure 4.13 Mean Average Precision for gRank and PageRank	98
Figure 4.14 Cumulative Gain for gRank and PageRank	99
Figure 4.15 Discounted Cumulative Gain for gRank and PageRank	100

Figure 4.16 Idealized Discounted Cumulative Gain for gRank and PageRank	100
Figure 4.17 Representation of Normalized Discounted Cumulative Gain in gRank and PageRank	101
Figure 5.1 Pseudo-code of ABC Algorithm (Karaboga and Akay, 2009)	106
Figure 5.2 Proposed Architectural Model for BeeRank	108
Figure 5.3 Dataset of web pages	112
Figure 5.4 Queries tested on web pages	112
Figure 5.5 Structure of web page	113
Figure 5.6 Representation of Fitness value in different Iterations	117
Figure 5.7 Representation of BeeRank value in different Iterations	117
Figure 5.8 Ranking of Nodes in BeeRank Model	118
Figure 5.9 Ranking of Nodes in PageRank algorithm	120
Figure 5.10 Fitness Graph during Iterations	123
Figure 5.11(a) Execution time of BeeRank and PageRank	124
Figure 5.11(b) Bar Graph of Execution time for BeeRank and PageRank	125
Figure 5.12 Number of iterations for BeeRank and PageRank	125
Figure 5.13(a) Result of iterations in BeeRank	126
Figure 5.13(b) Result of iterations in PageRank	126
Figure 5.14 Precision values for BeeRank and PageRank	127
Figure 5.15 Recall values for BeeRank and PageRank	128
Figure 5.16 Average Precision for BeeRank and PageRank	128

Figure 5.17 Mean Average Precision for BeeRank and PageRank	129
Figure 5.18 Cumulative Gain for BeeRank and PageRank	129
Figure 5.19 Discounted Cumulative Gain for BeeRank and PageRank	130
Figure 5.20 Idealized Discounted Cumulative Gain for BeeRank and PageRank	130
Figure 5.21 Representation of Normalized Discounted Cumulative Gain in BeeRank and PageRank	131
Figure 6.1 Number of iterations in PageRank, scRank, gRank and BeeRank	134
Figure 6.2 (a) Bar Graph of Execution Time for PageRank, scRank, gRank and BeeRank	135
Figure 6.2 (b) Execution Time for PageRank, scRank, gRank and BeeRank	135
Figure 6.3 Precision values for PageRank, scRank, gRank and BeeRank	136
Figure 6.4 Recall value for PageRank, scRank, gRank and BeeRank	137
Figure 6.5 Average Precision for PageRank, scRank, gRank and BeeRank	137
Figure 6.6 Mean Average Precision for PageRank, scRank, gRank and BeeRank	138
Figure 6.7 Cumulative Gain for PageRank, scRank, gRank and BeeRank	139
Figure 6.8 Discounted Cumulative Gain for PageRank, scRank, gRank and BeeRank	139
Figure 6.9 Idealized Discounted Cumulative Gain for PageRank, scRank, gRank and BeeRank	140
Figure 6.10 Normalized Discounted Cumulative Gain for PageRank, scRank, gRank and BeeRank	140
Figure 6.11 Graphically Analysis of PageRank, scRank, gRank and BeeRank	141

LIST OF TABLES

Table Name	Page No.
Table 2.1 Comparative Analysis of Information Retrieval Model	19
Table 2.2 Comparative Summary of Ranking Algorithm	39
Table 2.3 Contingency notion of accessing Accuracy	42
Table 3.1 Frequency of terms in Web Pages	62
Table 3.2 Cumulative Weight for the nodes	62
Table 3.3 Weight of each term of query in (A)	63
Table 3.4 Cumulative weight of nodes for query in Cluster 1	63
Table 3.5 Cumulative weight of nodes for query in Cluster 2	64
Table 3.6 scRank of Cluster 1	64
Table 3.7 scRank of Cluster 2	65
Table 3.8 Ranking of the Nodes	65
Table 3.9 Iteration Required	66
Table 3.10 PageRank of all Nodes	66
Table 3.11 Ranked Nodes	67
Table 3.12 Cumulative weight of the URLs	68
Table 3.13 Cumulative weight of nodes in Cluster	68
Table 3.14 Processing steps of Different Iteration in scRank	69
Table 3.15 Ranking result in scRank	69
Table 3.16 Ranking result in PageRank	70

Table 3.17 Ranking result in scRank and PageRank	70
Table 4.1 Occurrence of terms on Web Pages	86
Table 4.2 Weight of the nodes	87
Table 4.3 Value of Fitness, Probability, Probability Percentage, Expected Count and Actual Count for the nodes	88
Table 4.4 1 st Iteration after applying Genetic Operators	88
Table 4.5 2 nd Iteration after applying Genetic Operators	89
Table 4.6 3 rd Iteration after applying Genetic Operators	89
Table 4.7 Fitness Values in Different Iterations	89
Table 4.8 Ranked Nodes along with Fitness Values	90
Table 4.9 PageRank of all Nodes	91
Table 4.10 Ranked Nodes	92
Table 4.11 Iteration Required	92
Table 4.12 Weight of the URLs	93
Table 4.13 Fitness values during Iterations	93
Table 4.14 Ranking result in gRank and PageRank	94
Table 5.1 Occurrence of terms in Web Pages	113
Table 5.2 I (W), O (W), Fitness and probability of the nodes	114
Table 5.3 Value of x_{ij} , N_{ij} , f (UFS) and p (UFS) of the nodes in 1st iteration	115
Table 5.4 Value of x_{ij} , N_{ij} , f (UFS) and p (UFS) of the nodes in 2nd iteration	116
Table 5.5 Fitness Values in Different Iterations	116

Table 5.6 BeeRank Values in Different Iterations	117
Table 5.7 Ranked Nodes along with BeeRank Values	118
Table 5.8 PageRank of Nodes	119
Table 5.9 Ranked Nodes	119
Table 5.10 Iteration Required	120
Table 5.11 Density of Nodes and Query	121
Table 5.12 Processing steps of Different Iteration in BeeRank	122
Table 5.13 Fitness values during Iterations	122
Table 5.14 BeeRank value during different iterations	123
Table 5.15 Final BeeRank values	124
Table 6 .1 Comparative Result Analysis for PageRank, scRank, gRank and BeeRank	141

LIST OF PUBLICATIONS

1. Shadab Irfan, D. Rajesh Kumar, (2021), "BeeRank A Heuristic Ranking Model to Optimize the Retrieval Process", International Journal of Swarm Intelligence Research (IJSIR), Volume 12, Issue 2, Article 6, IGI Global (Scopus & ESCI).
2. Shadab Irfan, D. Rajesh Kumar, (2020), "Ranking Algorithm to Optimize the Retrieval Process Using Genetic Algorithm", International Journal of Control and Automation, Vol. 13, No. 2, pp. 383 – 396, ISSN: 2005-4297 IJCA, SERSC.(Scopus)
3. Shadab Irfan, D. Rajesh Kumar, (2020), "Improving Ranking Procedure In Information Retrieval Process Using scRank Algorithm", International Journal of Advanced Science and Technology, Vol. 144, No. x, pp. 1 – 18, NADIA, ISSN(P): 2005-4238,ISSN(O): 2207-6360 IJAST (Scopus).
4. Shadab Irfan, Subhajit Ghosh, (2019), "Efficient Ranking Framework for Information Retrieval Using Similarity Measure", Springer - Advances in Intelligent Systems and Computing Series.
5. Shadab Irfan, Subhajit Ghosh, (2019), "Ranking Web Pages Using Cosine Similarity Measure", International Conference on Computing, Power and Communication Technologies (GUCON 2019).
6. Shadab Irfan, Subhajit Ghosh, (2018), "Web Usage Mining in Ranking Algorithms", International Journal of Innovative Research in Science, Engineering and Technology (IJIRSET), Volume 7 Issue 10, ISSN 22319-8753.
7. Shadab Irfan, Subhajit Ghosh,(2018), "Analysis and Challenges of Web Ranking Algorithms", 4th International Conference on Computing, Communication and Automation –ICCCA 2018.
8. Shadab Irfan, Subhajit Ghosh (2018), "A Review on Different Ranking Algorithms", International Conference on Advances in Computing, Communication Control and Networking –ICACCCN (ICAC3N'18).
9. Shadab Irfan, Subhajit Ghosh (2018), "Web Mining for Information Retrieval", International Journal of Engineering Science and Computing (IJESC), Volume 8 Issue 4, ISSN 2250- 1371.
10. Shadab Irfan, Subhajit Ghosh (2017) ,"Optimization of K- Means Clustering Using Genetic Algorithm", International Conference on Computing, and Communication Technologies for Smart Nation (IC3TSN2017).

11. Shadab Irfan, Subhajit Ghosh (2017) ,“Optimization of Information Retrieval Using Evolutionary Computation: A Survey” , International Conference on Computing, Communication and Automation (ICCCA2017).

12. Shadab Irfan, B.V. Babu (2016), “Information Retrieval in Big Data Using Evolutionary Computation: A Survey”, International Conference on Computing, Communication and Automation (ICCCA2016).

LIST OF ABBREVIATIONS & SYMBOLS

IRS- Information Retrieval System

IR- Information Retrieval

df- document frequency

idf- inverse document frequency

tf- term frequency

WWW- World Wide Web

URL- Uniform Resource Locator

WCM- Web Content Mining

WSM- Web Structure Mining

WUM- Web Usage Mining

AI- Artificial Intelligence

NLP- Natural Language Processing

HTML- Hypertext Markup Language

XML- Extensible Markup Language

OLAP- Online Analytical Processing

SCC- Strongly Connected Component

AP- Average Precision

MAP- Mean Average Precision

CG- Cumulative Gain

DCG- Discounted Cumulative Gain

IDCG- Idealized Discounted Cumulative Gain

NDCG-Normalized Discounted Cumulative Gain

PSO- Particle Swarm Optimization

GA- Genetic Algorithm

ABC- Artificial Bee Colony

DE- Differential Evolution

ACO- Ant Colony Optimization

CW- Cumulative Weight

IW- Inlink weight

OW- Outlink weight

EC- Expected Count

AC- Actual Count

CO- Crossover

MU- Mutation

dd- Document density of web page

qd- Query density

FS- Food source

UFS- Updated Food source

CHAPTER 1

INTRODUCTION

In this chapter, an interaction of the work is provided. The organization of the chapter is as under. The motivation for the proposed work is given in Section 1.1 and Section 1.2 provide the problem statement. In the next section 1.3, the relevance of the work is highlighted. In Section 1.4, the outline of the dissertation is specified.

Ranking is considered as a major task in the Information Retrieval procedure. Ranking is regarded as the backbone which helps in arranging the documents in relevant order as specified by the user query.

With the increased volume of documents on the web, the researchers are faced with the challenge to locate their documents of interest. Different algorithms are being proposed to accomplish this task. It has been found out that millions of web pages are added daily and this explosive growth of web pages places huge pressure on the search engines which uses different algorithms for classifying and retrieving the much-needed information in stipulated time duration.

In ranking, we associate web pages stored in the repository in order of preference, so that the pages which have high similarity index according to the user query are ranked in higher-order in comparison to other web pages. For any search engine, ranking algorithm is considered as an indispensable part which overall effect the processing result. In minimum possible time, the most beneficial results should be displayed by the ranking algorithm.

1.1 Motivation

Searching and retrieving information in a scenario is a quite difficult job where information keeps on changing frequently with time. The selection of an optimal ranking strategy is a challenging task, as the current ranking measure sometimes does not provide promising results according to the need of the user, so finding a ranking strategy that satisfies the current need motivated us to find a new ranking algorithm.

With the flow of online streaming of data at an exponential rate, the traditional Informational Retrieval techniques are becoming inadequate for handling the large volume of data so better strategies can be adopted which can also include some nature-inspired algorithms that assist in optimizing the procedure and enhance the competence of the retrieval process.

User generated query produces abundance of web pages having both relevant and irrelevant content. It is observed that in some cases irrelevant pages are given more importance and thus ranked higher in the hierarchy providing inaccurate results. Search engines normally use a content-based approach which cannot be considered as the only via media for web pages ranking as the content can be manipulated to enhance the ranking of the web page. Similarly focusing only on the link-based approach is also not appropriate as most of the ranking algorithms use only a link-based technique for web pages ranking as sometimes links are induced only for navigational purposes or to increase the weightage of a page. In this regard hybrid approach can be used for enhancing efficiency.

For retrieving information from the web, the search engine plays a vital role which helps to retrieve relevant web pages. In the process of retrieving, certain irrelevant pages are also retrieved which cannot be avoided, so new measures have to be employed that help in minimizing retrieval of irrelevant web pages. The activities on the web are evolving both in volume and in heterogeneity and in turn, increase the utilization of the search engine. It has also been observed that to perform ranking, the huge number of pages in the corpus has to be processed which occupy maximum time for computation and retrieval and thus lower the speed and time of retrieval. Clustering or grouping of web

pages based on certain conditions helps to make small clusters. This grouping of web pages will help in performing computation easily and enhance the retrieval rate.

The key motivation of my research work is to investigate different techniques and algorithms that ease the process of retrieval, as in the information retrieval system, ranking algorithms observed as the backbone that is the main focus of my research. By using a suitable ranking algorithm we can ease the process of retrieval that helps to boost the precision of the retrieval system thereby enhance the work of retrieving relevant documents for a particular query. The information present on the web is considered a useful resource for users to access distinct kind of information according to their preferences.

The proposed system first presents an outline of the various ranking algorithm, by analyzing their strength and weakness and propose a new ranking algorithm with certain new features for ranking web pages easily and satisfy user need in minimum time duration. Different measures have been employed for ranking the web page like Nature Inspired - Genetic Algorithm and Artificial Bee Colony Algorithm which assist in minimizing the computational complexity of the ranking procedure.

1.2 Problem Statement

With the increase of digital information at a rapid pace, there has been a tremendous change in the World Wide Web which urges the researchers to find different ways to manage this enormous flow of information and help in satisfying the need of the user. It has been observed that the core component of any search engine is the ranking framework that helps to improve the result quality based on user queries by ranking the web pages. A good ranking model should incorporate diverse measures to ameliorate the quality of the result and should not stick to a single measure. It can be pointed out that the Information Retrieval process can be improved by incorporating content mining and link analysis.

Various nature-inspired ranking algorithm is used which integrate different measures and help to optimize the quality of the result obtained. Evolutionary algorithms are exploration techniques inspired by nature and are regarded as a population-based stochastic algorithm. Different global optimization algorithms like Genetic Algorithm, Differential Algorithm, Particle Swarm Optimization, Ant Colony Optimization, and Artificial Bee Colony are used for searching and ranking purpose. To solve computer-related problems evolutionary computational models play a major role. They are used in many learning tasks and in many cases are considered as optimization techniques for solving the problems.

The work in the thesis aims to help the user in ranking their web pages according to the query generated efficiently in less time-bound. The proposed work helps in ranking the retrieved pages where the web pages are initially grouped on basis of certain condition so that irrelevant pages can be removed during the ranking process. It minimizes the computational complexity of the procedure and help in retrieving page in less time. Nature-inspired algorithms like the Genetic Algorithm and Artificial Bee Colony are employed to optimize the ranking algorithm which helps to rank the web pages. A hybrid formulation of both the content-approach and structural approach is used that not only consider the density of terms of the user query, but also handle the link structure among the web pages. The hybrid approach helps in ranking the web pages more efficiently.

1.3 Relevance of the Work

On the web, there is a great challenge to maintain the up-to-dateness of data so that the exact information can be retrieved easily. Most of the users on the web are laypersons that do not possess the correct knowledge to formulate their queries so that the correct information can be obtained. The information stored should be properly indexed and ranked so that the user can get the result of their query in a limited time constraint.

Users normally suffered a lot for accessing the desired information and it is the task of ranking algorithm to prioritize the result obtained so that the need of the user can be satisfied. Ranking algorithms that focus only on link structure does not provide relevant

results because they ignore the content within the pages and sometimes also suffer from proper feedback mechanism.

Users normally viewed 10-15 results based on their queries, so an efficient ranking system should produce a finite number of ranked pages that satisfy the quest of the user. Ranking has a major part in simplifying the interaction of the user with the obtain results by ordering the result of web pages. Relevancy is prime factor in the Information Retrieval System that helps in document ranking based on the needs of the users. Optimization techniques make use of available resources and help in finding the best solution without disturbing the desired conditions. For solving high-dimensional problem nature-inspired optimization techniques are more preferred in comparison to classical optimization methods. Various optimizations algorithms are present that helps in dealing with the complexity of the real-world.

For providing efficient results the ranking algorithms are tested on different datasets and it has been discovered that on using the proposed approach the number of iterations required to achieve the target is minimized tremendously. We can not only minimize the computational complexity of the procedure by limiting the number of iterations, but we can also achieve the result in less time-bound. By including certain clauses in the algorithms it removes irrelevant pages during the initial phase of the ranking process, which overall reduces the load and decreases the computational complexity of the process. It is observed that the given work when compared with other standard algorithms like PageRank, improves the mean average precision. The various evaluations measure for accessing the ranking quality is applied to the proposed work and it is observed that nature-inspired optimization technique used for web pages ranking help in increasing the Normalized Discounted Cumulative Gain value.

The first contribution of the thesis is the scRank algorithm that computes the cumulative weight of the nodes by considering the link structure and cumulative weight of web pages by considering the content of the web pages concerning the query before applying the scRank algorithm.

The second contribution of the thesis is the gRank algorithm that uses the Genetic Algorithm for web page ranking where fitness and probability of the pages is evaluated, and by monitoring the higher fitness value pages are ranked.

The third contribution of the thesis is BeeRank algorithm that considers the user query as bee which goes from page to page and collects its desired content from the web page which has the max quantity of the user query and thereby, calculates the rank value which is governed by the fitness of the food source.

1.4 Outline of the Thesis

Chapter 1 provides the Introduction, and define problem under investigation. The thesis includes six more chapters.

In **Chapter 2**, a survey of the current progress in the field of Information Retrieval and Ranking Algorithm is presented. A short outline of various algorithms of the Information Retrieval process is presented and the different evaluation measures used are also discussed. Various forms of Web Mining Techniques like Web Content Mining, Web Usage Mining, and Web Structure Mining is also elaborated. This Chapter describes the various ranking algorithm and also describe different evaluation metrics.

Chapter 3 contains an introduction and working on our first proposal. We propose a scRank algorithm that helps in clustering the web pages at the beginning of the process by checking the content of the web pages regarding the user query thereby eliminating the pages which lack proper content and also considers the structural links among web pages. The scRank filters the page based on different criteria as stated in the algorithm and computation is done for selected web pages that assist in minimizing the computational complexity of the process. The results are evaluated using dataset on Spyder platform.

Chapter 4 contains an introduction and working on our second proposal. We propose a gRank algorithm that uses Genetic Algorithm for web page ranking and consider both

structural characteristics and content of the web pages. Fitness and probability of the pages are evaluated, and by monitoring the higher fitness values, pages are selected and genetic operators, single-point crossover, and flip mutator are applied for getting better results. Roulette Wheel is applied for selection which is directly proportional to the fitness value. The results are evaluated using dataset on Spyder platform.

Chapter 5 contains an introduction and working on our third proposal. We propose the BeeRank algorithm that considers the user query as bee which goes from pages to pages and collects its desired content from the web page which has the max quantity of the user query, the pages are prioritized in order of higher fitness value and are arranged in BeeRank order. The given approach undertakes the structural linking and content of the web pages for ranking and thus enhances the efficiency of the information retrieval process. The results are evaluated using dataset on Spyder platform.

Chapter 6 provides a comparative evaluation of all the three proposed algorithms with the basic PageRank algorithm and the results are discussed and compared using dataset on Spyder platform.

Chapter 7 sketches the conclusion and Future Scope of the dissertation.

1.5 Summary

In this chapter, the fundamental issues and challenges faced for ranking the web pages are introduced. It also focuses on the proposed technique for extracting relevant pages by grouping them initially and later applying evolutionary techniques which are further implemented in this research work.

CHAPTER 2

LITERATURE REVIEW

This chapter provides a comprehensive background of the Literature Survey conducted during the thesis work. The chapter is arranged as follows. Section 2.1 gives an overview of Information Retrieval, Section 2.2 throws light on Information Retrieval and Web search and Section 2.3 presents Challenges faced in Web Information Retrieval. Section 2.4 provides information about Information Retrieval Process and Section 2.5 provides a comprehensive survey of Information Retrieval Models. Section 2.6 deals with Web Mining and Section 2.7 deals with Web Mining Process. Section 2.8 presents an overview of Document Clustering and Similarity Measure. Section 2.9 outlines about Ranking Algorithm and Section 2.10 outlines about Connectivity-based ranking. Section 2.11 provides a comprehensive survey of Various Ranking Algorithms while Section 2.12 provides an overview of various Performance Evaluation Measures. Section 2.13 outlines Optimization Techniques and gives a brief overview of the Nature Inspired Algorithm.

2.1 Information Retrieval

Information retrieval broadly deals with retrieving different data objects from a vast collection of data. It is an interdisciplinary science that deals with mathematics, computer science, statistics, etc. It mainly comprises of three subsystems that play a major role in the retrieval process. First is, how documents are represented, second deal with the user need i.e. formulation of the query part, and lastly a suitable algorithm that helps to match the user query with the documents being stored so that top highly relevant documents can be retrieved. Ranking of the documents are focused on the relevance of the documents that consider user query and form an integral part of Information Retrieval systems [1]. The process of Information Retrieval is shown in Figure 2.1[2].

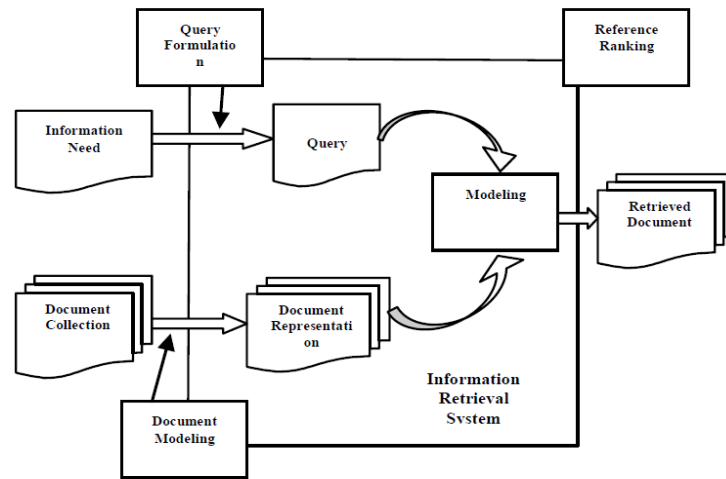


Figure 2.1 IRS Structure [2]

Information retrieval not only deals with retrieving relevant documents for the user but also helps in predicting the relevancy of the content. It overall helps the user in getting access to the top- rated documents by properly indexing and ranking them. The broad area of Information retrieval covers every corner of the globe. It can be applied to digital libraries where the repository of work is stored and can be accessed according to the need. Recommender systems help the user to predict certain items of relevance based on the rating given for an item and normally based on content approach or social environment and regarded as collaborative filtering.

Information Retrieval is normally associated with the extraction of relevant as well as non-relevant information. Information extraction performs the tasks of extracting relevant facts from documents while Information retrieval selects relevant documents. Information extraction works at a finer granularity level concerning Information retrieval [3]. Various optimization techniques like the Genetic Algorithm, Differential Algorithm, etc [4] are being employed that ease the process for getting an optimum value for the information retrieval process.

Fuzzy systems are required to handle uncertainty as the information required by the users is imprecise and on many occasions the query change during the retrieval process. The basic procedure of Information Retrieval generally starts using a simple query that is

generated by the user. Based on the formal statement of the query the items are ranked properly, and the top-ranking items are revealed to the users [5].

2.2 Information Retrieval and Web Search

The search engine has a key role in the application of information retrieval where information is drawn from web pages and satisfies the needs of the user. Helping users to retrieve documents based on relevance feedback allows easy access to information which is based on rating criteria. World Wide Web is a platform where users from different spheres share their thoughts, feelings, ideas, and memories. It is a place where user can access documents/pages which is based on various areas of importance. It is a repository which constitutes information from different fields and it keeps on updating.

The web is constituted by a large volume of data that is semi-structured and unstructured. Apart from this, the users who try to access information from this unstructured repository also create a problem using improper formulation of query and this creates a challenge for the search engines to maintain a balance between these two so that the need of the user can be satisfied. Satisfying the need of the users is the foremost role of the search engines which employ different tools and algorithms so that the quest of the user can be fulfilled. It facilitates extracting useful information according to the need of the user and possesses various challenges to retrieve information that is semi-structured [6].

Web is a repository where different sets of activities persist of varied nature like shopping, transaction, searching, weather report, etc. It facilitates the exploration of interesting facts and figures which is useful in many cases and helps the user satisfy their quest for knowledge. Web Information retrieval is regarded as an application of IR where the need of the user is not static, but dynamic and keeps on changing. It is related to web mining where a specific type of data is mined on basis of content, link and log data and retrieve user needed information.

Web Informational Retrieval is prescribed as an application of Information Retrieval to the web and is related to web mining which helps to discover interesting and valuable information from the web [7]. To disseminate a huge amount of information the World Wide Web is one of the interactive and accessible approaches. The information is

overloaded following the current scenario and users are drowning in the pool of information. Due to the flood of information, the users are facing various difficulties in extracting the relevant and pertinent information according to their needs. For accessing information, search service is employed where a simple query provides a list of pages by considering similarity measures. Searching create problem in providing low precision and irrelevant results, due to the inability to index the information the recall is found to be low [3]. For efficient searching and browsing, the clustered documents help to provide a structured format. A good model of Information retrieval helps in explaining and predicting relevant user needs according to user query [8].

The processing of retrieving web pages progresses through a series of steps, starting from crawling, extracting, matching, and finally ranking the pages as per the need of the query, generated by the user. Different researchers from across the globe derive various ways and measures, which help the user in getting their information in minimum time duration. In this regard, many ranking algorithms are proposed which assist to rank the relevant informational pages at their disposal, and irrelevant pages are discarded thus overall reducing the processing time also. Search engine performance is hugely dependent on the ranking algorithm being used [9]. In this work, ranking algorithms are proposed that help to filter the irrelevant pages at the starting phase of filtering and thereby uses different approaches to rank the retrieved result as per the need of the query.

2.3 Challenges faced in Web Information Retrieval

It is one of the great challenges to retrieve the relevant pages from the web so that a needless amount of time can be eliminated. The queries that are generated in IRs are usually vague and the algorithms are deficient in generating relevant documents. The documents have to be ranked based on relevance. There is a need to evaluate the result of the query in information retrieval, by judging how precise the answer of a query is. The process of evaluation is referred to as Retrieval Performance Evaluation and it is required to judge which algorithm is best [10].

2.4 Information Retrieval Process

In recent years, the advancement in various fields of information retrieval has reflected a sudden rise and many research activities are bringing ease to the retrieval process. Information Retrieval has three basic processes [1] content representation for the document, depiction of user information, comparing both representations. Manning et al. in [11] defined IR as the process of searching for information from a large collection [12]. A perfect system should be able to satisfy the quest of the user by providing relevant documents at his disposal. Three basic processes comprise the information retrieval system-how contents of the documents are represented, how user needs are interpreted, and how a comparison is made between the two. The process can be visualized as shown in Figure 2.2. [13]-

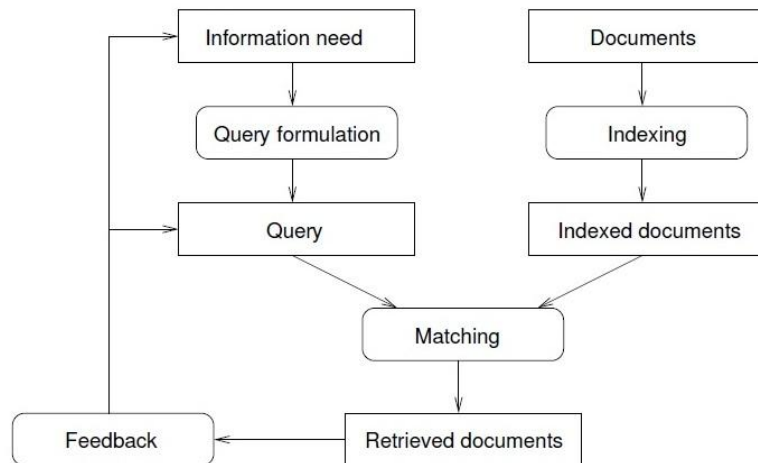


Figure 2.2 Information retrieval processes [13]

Information Retrieval has a vital part in fulfilling the quest of the user no matter where the information is stored. It focuses on what the user want, try to acquire the content using different measure and then properly matching the retrieved content with the user need so that relevant content can be provided in a short duration. They help in extracting interesting information from the vast collection of documents and there has been a rapid

change in the retrieval system which helps the user in managing and retrieving the information.

The indexing process normally deals with representing the documents which is an off-line process and result in the representation of documents. The query formulation process helps the user in representing the needs so that the system can better understand it and provide valuable results. After this, the documents are matched with the query and provide a ranked list of documents that save user time to a greater extent in checking all the retrieved results. In information retrieval process ranking algorithms plays a crucial role.

In the process of information retrieval, tokenization performs the task of chopping the character streams into tokens for further processing. Stopwords are words that are of little value in the document collection and are removed from the documents by computing the frequency of the terms that appear in the document [14]. The process of stemming & lemmatization helps in reducing the inflectional forms, on one hand stemming chops off the ends of words regarded as a crude heuristic process using Porter's algorithm while lemmatization performs the task by using vocabulary and morphologically analysis of words by providing dictionary form of word lemma. Term weighting is the final phase of text preprocessing where index terms are weighted differently on account of their significance in the documents. It is one of the basic needs of the search engine to rank-order the documents that matched a particular query term, and to perform this task a score is computed for the matching document.

2.5 Information Retrieval Models

Information Retrieval Models provide a way to represent documents in various ways so that user-specific information can be retrieved easily [15]. The models help to facilitate the procedure of information retrieval and act as a blueprint for outlining various activities. Different models for Information Retrieval process are-

2.5.1 Boolean Model

In IR, Boolean model is the basic model that expresses a query in form of Boolean expression which is combined with Boolean operators and is referred to as the classical model by Van Rijsbergen [16] which is based on Boolean logic theory. The basic operators which are used are 'AND', 'OR' and 'NOT' that are used for the construction of query and retrieval of documents. This model does return too many or few documents which are not properly ranked [12].

The underlying issue with the Boolean model is that it generate high precision & low recall while employing AND operator and similarly low precision & high recall with OR operator so it is really hard to maintain a balance between the two [14]. It does not give a ranking of the retrieved documents which is a major flaw of this model and it undertakes unambiguous set of documents as query term [17].

2.5.2 Vector Space Model

Vector Space Model is a well-known models proposed by Salton et al. [18] where multidimensional space is considered to view documents and queries which are treated as vectors. In a high dimensional Euclidean space, queries are embedded which are treated as vectors in the Vector Space Model. The problem arises when frequency of the document is changed while adding a new document [19]. Between the document and query the similarity is measured using various similarity measure approaches which help to rank the document based on how close it is with the user query. Here the terms are treated independently as a result of which structure of document is ignored [12].

Vector model allot real non-negative weight to index terms in documents and queries, where documents are vectors in multidimensional Euclidean space. The index term weight is a function of two factors- frequency factor, Term Frequency that is intra cluster similarity, and IDF which is a measure of inter-cluster similarity. Due to term weighting, there is improvement in retrieval performance [20]. One of the fundamental hosts for

various operations like scoring document, document classification, and document clustering in information retrieval is the vector space model.

The score between the query term 't' and a document 't' in 'd', is evaluated on weight 't', and is termed as term frequency $tf_{t,d}$. Number of documents in the corpus having term 't' is referred as the document frequency 'df_t'. In this regard the entire documents is represented by N and inverse document frequency is represented as shown in 2.1.

$$idf_t = \log \frac{N}{df_t}. \quad (2.1)$$

It is found out that idf of the rare term is found to be high & low for frequent terms. A composite weight is evaluated by considering both tf and idf and is represented as shown in 2.2.

$$tf-idf_{t,d} = tf_{t,d} \times idf_t. \quad (2.2)$$

It is used to allot a weight to a term. The score of a document 'd' is evaluated which is sum of all query terms found in document 'd' [14] as shown in 2.3.

$$Score(q, d) = \sum_{t \in q} tf-idf_{t,d}. \quad (2.3)$$

For quantifying the similarity between the documents, the cosine similarity measure is computed between two documents d_1 and d_2 for vector representations $V(d_1)$ and $V(d_2)$ as shown in 2.4.

$$\text{sim}(d_1, d_2) = \frac{\vec{V}(d_1) \cdot \vec{V}(d_2)}{|\vec{V}(d_1)| |\vec{V}(d_2)|} \quad (2.4)$$

The nominator represents the dot product while the denominator is considered as the product of Euclidean lengths.

The Euclidean length 'd' is defined as $\sqrt{\sum_{i=1}^M \vec{V}_i^2(d)}$. and thus similarity is specified as shown in 2.5.

$$\text{sim}(d_1, d_2) = \vec{v}(d_1) \cdot \vec{v}(d_2). \quad (2.5)$$

2.5.3 Probabilistic Model

The probabilistic model that focus on the probability of ranking principle is defined by Roberston [21], where documents are ranked on the probability of relevance. In the probabilistic model, the ranking is done based on feedback that is gathered. The documents are arranged in the decreasing order of their probability but it creates problems in estimating the initial separation of documents into relevant and non-relevant and index terms are supposed to be independent [20].

Cooper et al. [22] stated that this model has powerful statistical indicators that help to judge the goodness of a model and less importance is given to the trail & cure method of retrieval [12].

2.5.4 2-Poisson Model

The probability of k number of events is modeled using Poisson distribution that occur in a fixed interval of time and space. The "2-Poisson model", considers that the distribution is different and depends on whether the term is elite or not [14]. It is presumed that documents are produced using an arbitrary stream of term occurrences and divided into two subsets. If documents are selected at random from one subset then its probability of relevance is supposed to be identical to or higher than the probability of relevance of document from the second subset. The subsets have different Poisson u_1 and u_2 . The advantage of the 2-Poisson model lies in that it does not require an additional term

weighting algorithm to be implemented but estimating parameters create problems and it is quite complex[13].

2.5.5 Bayesian Network Model

In the IR environment for managing the uncertainty, the Bayesian network model is considered that are graphical models which efficiently manipulate n-dimensional probability distribution [15]. Turtle and Croft model helps to understand the complexities connecting document and user's information need. The model is decomposed into a query network that is relatively small and built a new network whenever a query comes in and the other is a document collection network that can be precomputed and the query is attached to the document network [14]. It is an acyclic directed graph that encodes the probability dependency relationships among random variables. By utilizing a graph theoretical approach it helps to analyze complex conditional assumptions and require too much space [13].

Knowledge can be expressed in a graphical format that helps to express dependency and independency relationships. The knowledge can be revealed in terms of dependence and independence relationships between variables using graphical components. The relationships help to encode the existence or absence of links between nodes of a graph and help to introduce uncertainty within the model. The available knowledge can be represented intuitively using graphical tools and probabilistic inference can be performed efficiently [23].

2.5.6 Language Model

The language model proposed by Zhai & Lafferty [24] utilizes probabilities of term occurrence it utilizes smoothing techniques for providing non-zero probability for a specific document that has the query term, and has better performance concerning the probabilistic model of retrieval [8]. It is inherently probabilistic and puts a probability

measure over strings that are extracted from some vocabulary. Language model provides a distinct perspective to score match between queries and documents and uses simple models of language, and provide good result in the retrieval process. For each documents, these models are built separately [14].

Language models in information retrieval are designed for each document and are found to be originated from probabilistic models of language generation and are created for automatic speech recognition systems [13]. Indexing function is used in the Fuzzy model to map document, where a term is mapped on the document using an indexing function. Here the fuzzy queries are based on weighted terms that are expressed using a query language [25].

2.5.7 Google's Page Rank Model

This model aims to return high quality documents and uses a hyperlink structure for accessing the quality of the pages. The model focuses on returning high-quality pages from trusted sites rather than returning pages that match closely with the query terms. By employing hyperlink structure it assesses the quality of the page referred to as page rank. Pages that are linked with high-quality pages are also considered important and worth looking at. It does not depend on user query and therefore referred to as static ranking function and computed off-line during indexing [13]. If the Pagerank of each page is computed then it can be employed throughout querying.

A comparative analysis of the Information Retrieval Models is presented in Table 2.1 [26]

Table 2.1 Comparative Analysis of Information Retrieval Models [26]

S.No.	Model	Strength	Weakness
1	Boolean Model	Flexibility and exuberance. Allow complex search requirements.	No ranking of retrieved documents.
2	Vector Space Model	Relevance scoring and relevance feedback	Implementation problem and vector components values undefined. Computational expense.
3	Probabilistic Model	Used to retrieve documents based on probability of relevance. Operates recursively.	Hard to build and program. Complexity grows quickly.
4	2-Poisson Model	Not require an additional term weighting algorithm.	Problem in estimation of the parameters. Require probabilities conditioned on relevance.
5	Network Model	Include terms dependence relationships. Diminish computation cost by using simplifications.	As query nodes increases, use of the network is impractical. Node growth is exponential with number of parents.
6	Bayesian Model	Analyze complex assumptions.	Time required assessing the distribution and space requirement is NP hard problem .
7	Language Model	Novel way of looking at the problem. Probabilistic language modeling.	No notion of relevance in the model: everything is random, query expansion is not part of the model, does not directly allow weighted or structured queries.
8	Google's Page Rank Model	Less Time consuming, computes rank score at indexing time not at query time.	Favors older pages. Rank sink problem. Dead Ends. Reduce front Page's page rank if circular reference. Spider traps if no links from within the group to outside the group. Dangling links problem.

2.6 Web Mining

The analysis of pertinent information from the World Wide Web (WWW) is referred as Web Mining. It is a converging research that includes research from different areas like Artificial Intelligence, database, Information Retrieval, Machine learning, Cloud

Computing, Human Interaction, Natural Language Processing, etc [3]. It is viewed as data mining usage that helps to retrieve, extract and analyze information. As most of the concepts are approximate so web mining sometimes provides poor mining results [27]. The existence of search engines came into the limelight with the fast growth of information on the web. They are almost used by billions of people daily and regarded as one of the most widely implementation of information retrieval techniques [28]. The main focus of Web Mining is to enhance the structure, content, and usage of web documents through data mining algorithms. Web Mining techniques are affected by the presence of irrelevant, redundant & insignificant features that affect the clustering, classification & prediction features [29].

Web mining encapsulates areas governing with Data Mining & World Wide Web. It helps to discover interesting information and patterns from the web data that ameliorate the power of the web by classifying and helping to manage the pages. It follows the navigational pattern of the user to collect different information. Web Mining help to extract useful patterns and implicit information related to World Wide Web. It is a challenging task to efficiently search and extract information from the web. Many of the search tools try to extract innumerable documents, though only a fraction of them is relevant. The search engine assists the user in accessing the relevant links of URLs as per their query requirement.

The search engine comprises of crawler, indexer, query processor, and ranking algorithm. The crawler job is focused on the hyperlink structure of the websites and extract web pages which are then passed to the Indexer that analyze the documents and help in creating the index for fast and easy retrieval. After this, the query processor retrieves the related documents by referring the database, and before delivering the result to the users they are first ranked by following a suitable ranking algorithm which accesses the relevant documents [30]. The application areas of Web Mining comprises of- Robot Detection (navigate hyperlink structure), Extracting User Profile (User Information), Finding Significant Pages on the web, Google Search Engine Application, etc.

2.7 Web Mining Process

Web Mining helps to extract interesting, relevant patterns from web documents that are required in the knowledge discovering process. It is regarded as the evaluation of information which is mined and it helps the user to perform different sets of activities. The mining of information can be performed by analyzing the link structure of the pages or by observing the content found within the pages. It can also be accessed from log files which are maintained on the server [30]. Web Mining by employing the benefit of content, link, and usage statistics helps the user in assisting them, in making their needs fulfill and helping to trace the useful information quickly and efficiently. The process of Web Mining normally incorporates content, usage and structure. It encapsulates information extraction which then follows analysis. The content includes the text and the multimedia part while structure analyzes the link of the web and log handles the log files [31].

Based on mining data they are categorized into Web Content Mining (WCM), Web Structure Mining (WSM), and Web Usage Mining (WUM) [3]. Some researchers classified Web Mining into four subtasks. First is Information Retrieval also termed as resource discovery where the aim is to extract relevant documents while at the same time filtering non-relevant documents. Second is information extraction which helps in extracting specific fragments of documents from web resources and uses wrapper technique for extraction as the content on the web is diverse and dynamic. The Third is the Generalization where user behavior and interest are analyzed by observing information patterns with the help of techniques like clustering and association rules. The final is the analysis and validation part which analyzes the information pattern and helps in validating web information.

The area of Web Mining is huge as it indulges in various research activities which incorporate sources of different areas. Web Mining encapsulates different sub-tasks that include activities of retrieving independent web documents, selection and pre-processing of specified information from web resources, discovering patterns from different sites and validation, and interpretation of mined patterns. Web Mining is regarded as an extension

of Knowledge Discovery in Databases (KDD) which is applied to the web data, and is often associated with information retrieval or information extraction [3]. It incorporates various efficient techniques like AI, Machine Learning, NLP so that the user can find the information efficiently [32].

2.7.1 Web Content Mining

WCM handle the content of the pages and focus on the result obtained. It deals with extracting and integrating useful data that help in performing knowledge discovering from the page content that represents structure, semi-structured and unstructured documents [29]. It is one of the important parts of any page which shows the relevance of the material and normally is a collection of facts which help to provide a means of communication to the user. It not only comprises of text and images but also can include audio, video, records, etc. Web content mining normally work with the factual content which is the core part of any page and can be categorized into-

1. **Agent-Based Approach:** - Help in filtering the information using intelligent agents.
2. **Data-Based Approach:** - Deals with a structural form of data by connecting the multilevel database and query systems [30].

WCM retrieve content of the web pages which are indexed for proper tracking. The content on the web can be unstructured (plain text), semi-structured (HTML document) or structured (accessed from database). Some researchers conclude that content mining is text mining and in today scenario is strongly related to web structure mining which helps in extracting and organizing information from the web. It enables retrieval, organization, automated discovery, and systematic management of resources available on the web. Cooley et.al [33]emphasized research in the area of Content Mining in two approaches- Information Retrieval and Database Approach, while the former presume the development of a sophisticated AI system that work on behalf of the user, the later help in organizing web-based information. Web Content Mining outlines the discovery of

valuable information from web data. Some of the web content data are hidden and some data are dynamically generated through queries. The data can be in an unstructured format like HTML data or data found in tables or database that is regarded as structured. [3].

For extracting and managing web data in a formal way, web content mining is furnished with different tools. For easy retrieval of the web content, various algorithms are employed. Different algorithms are- K-nearest neighbor, Decision Tree, Naïve Bayes, SVM, and Neural Network. The utilization of Web Content Mining span to a larger area from grouping to classifying information present on the World Wide Web. In online marketing it quite helpful in tracking the behavior of the users which overall assist in market analysis. The online shopping sites can be reframed as per the need of the user based on customer feedback and reviews. The online content is clustered together from different social sites which help to maintain personalized data and it assists in decision-making tasks [34].

2.7.2 Web Structure Mining

Web Structure Mining involves the mining of links and the structure of web pages. It is quite useful in accessing the linkage path, hyperlink structure that helps in improving the result of the search engine [29]. The graphical structure which comprises nodes and edges constitute the structural mining part of web mining. It helps to establish the relationship between different web pages utilizing link structure which provides interconnectivity among the web pages. It can be categorized into-

- Hyperlinks
- Document Structure

For connecting various location of the web page hyperlink structure is used, similarly document structure encompasses the content that lies within the page and based on HTML and XML it can be structured in tree form [35]. In social network analysis hyperlink structure is quite useful that uses Web Structure Mining [3]. For web sites and web pages the significance lies in finding the structural summary. For connecting web

pages world wide web contains web pages, links (inlinks and outlinks) and co-citation that link two pages by same page. Link based mining techniques assists in performing various tasks like-

- Category prediction is possible based on words found on pages and the links among the pages.
- For discovering hidden patterns similar types of links are clustered together.
- The number of links, purpose and weight of the links can be allocated on work performed [36].

For inferring knowledge the links in web structure mining are employed that relate to web documents from where content is extracted based on query assigned. The main objective of web structure mining lies in analysis of the graph and for ranking the web pages based on links different algorithms are proposed like-Page Rank Algorithm, HITS, Weighted Page Rank Algorithm, and Weighted Link Rank Algorithm [37].

The focal point of Web Structure Mining lies in generating information like similarity and relationship among web pages and identifies authorities that are important sources of information. Undiscovered link structures of the web are established on link topology and it is quite useful for categorization of the web pages and help in generating similarity relationship among the web pages [3].

2.7.3 Web Usage Mining

Web log mining usually handle data related to user behavior, accessing time, etc that are stored in log files, and helps in accessing user pattern of web pages. Web Usage Mining helps in learning & predicting human behavior & navigating patterns of uses by analyzing the server access log files, session, or transactions [29]. It deals with discovering interesting and useful patterns and helps in accessing pages that can assist to predict user navigation or improving web site structure. By analyzing the statistics of the log files we can get various information and solve queries about the most visited pages, recent visits, which browser search engine used, and many more. The usage mining deals with extracting the log file which is stored on the server. Log files are useful in analyzing user browsing patterns and can ease the process of retrieval [38]

Web Usage Mining is comprised of Data Collection, Pre-processing, Pattern Discovery, and Pattern Analysis steps. The data which is collected is not appropriate for the mining process therefore pre-processing is needed for converting data into an acceptable form for further processing and produce precise and concise data. Various methods like path analysis, association rules, sequential patterns, clustering, classification, etc are applied for discovering novel, interesting and useful information. For detecting useful pattern various pattern discovery techniques are used like path analysis where graph models are used for defining the relationship among web pages, the association that relates pages and help in predicting correlation among a set of items, sequential patterns that record time, the sequence of transactions, clustering which assist in grouping similar browsing pattern. Pattern Analysis help in eliminating uninteresting patterns with the help of techniques like Knowledge Query Mechanism, OLAP, and visualization. It helps in knowing user needs its effectiveness, assist in detecting intrusion and website modification [39].

To satisfy the need and requirements of the user and help them serve better, log files play an interesting role. Web Usage Mining help to extract user information from the log files by analyzing their browsing pattern and history, time spent on a particular page, and other factors that not only help to enhance the working of the websites but also help the user in finding relevant information. By analyzing the data it can be interpreted that certain pages are of interest to a particular group of user or certain items are favored by a group of user. Such types of activities help in increasing the business and enhancing the structure of the sites. By applying association rules, classification and clustering on the web pages the most frequent visited and associated pages can be traced and grouped together. These process can overall assists in improving the ranking of the web pages so that the needed information can be traced easily. The Recommender System can be employed which help the lame user in recommending certain pages of interest based on search query. By analyzing the navigational behavior of the user and pages of interest more similar pages can be recommended which help in the searching process [40].

Web Usage Mining focuses on web server sessions and behavior that generate data and helps in mining secondary data which is received during the interactive sessions of the users. It consider the data that is generated from proxy server logs, web server access logs, user profiles, user session, cookies, transactions, mouse clicks etc[3].

2.8 Document Clustering and Similarity Measure

Clustering is referred to as the grouping of similar documents for accelerating the information retrieval. New approaches are developed for clustering documents and are exclusively planned for processing extremely large data sets which are appropriate with some minor modifications in Web-based information retrieval systems [14]. It is an unsupervised learning method, while classification is a form of supervised learning. It facilitates user in navigating a collection of docs [41]. The clustering approach is employed that help to partition datasets into smaller groups that are coherently related to each other. The process of clustering is employed for partition data sets into groups based on predefined patterns [42]. Cluster-based navigation is quite useful if users prefer browsing instead of searching as they are unaware of what they want to search and is usually preferred for keyword searching. It is considered for news stories where the need is to select a subset of stories or news. It also helps in improving the search results and speed up the process. Clustering is seen as essentially a search problem where the target is to optimize an objective function. The objective function tries to attain its goal of maintaining high intracluster similarity, where documents inside the clusters are similar and lowering inter-cluster similarity where documents from different clusters are dissimilar.

The clustering algorithm main input is the distance measure in the 2D plane and is regarded as Euclidean distance. Approach can be used for finding the clusters that are close to the query term and examine the documents from these clusters only. For this smaller subset similarity can be computed easily and documents can be ranked easily. In hard clustering, the document belongs to only one cluster while in soft clustering it is distributed over clusters. Cluster not only helps to identify topics but also support user navigation. It also helps in enhancing the performance of information retrieval process thus optimizing overall search engine [43], [44]. The search result in clustering approach helps to create a cluster of the search results, as such similar documents will appear together [14]. Ranking for retrieving the document is closely associated with clustering

the document having a high similarity index. Performance and efficiency of ranking can be enhanced if pre-grouping documents are being employed [45].

2.9 Ranking Algorithm

The World Wide Web is becoming an indispensable part of human life across the globe and it has been estimated that the number of online users on an everyday basis is increasing in manifold amount over the past few years. This sudden rise is due to the result of social media usage and internet technology. Different ways of social media have gripped the human life from every hook/corner like Facebook, Twitter, Instagram, and YouTube where users share their thoughts and feelings with their friends across the remote areas. The e-commerce and other sites are creating more digital data than before and in such a scenario the need for good and advanced technology that can manage and retrieve the user needed data in minimum period becomes a necessity.

Ranking is one of the core ingredients of any retrieval system as it delivers the much needed relevant information based on the query generated by the user. A good ranking system should help the users in managing the most pertinent documents at the top level of hierarchy concerning irrelevant documents, as due to lack of time users are interested in retrieving only a limited number of documents to all retrieved documents. Users normally seek the ranked results that are found in first pages i.e. few limited documents are retrieval for a particular query [46], [47]. Relevance feedback and clustering also play a key role in the ranking procedure. The algorithm must be perfect not only in respect of time but space also. A good ranking model should incorporate some of the basic characteristics like-

- Efficiency- A ranking algorithm should perform the computation efficiently with the set of data provided and giving good results.
- Effectiveness- The efficiency of the algorithm should exceed in terms of evaluation parameters when compared with other algorithms and thus focus on the desired result.
- Robustness- The algorithm should be robust with the data set in providing the relevant result based on query generated [48].

In simplifying the interaction of the user with the obtain results a major role is played by the ranking algorithms that provide ordering of the web pages. Relevancy is an important parameter in the IR system that helps in ranking the documents. Any system is effective only, if it provides the content as required by the user [49], [50]. For a given a set of queries which is applied over document sets, if the query terms are found in the documents it is termed as relevant, otherwise non-relevant. It can be illustrated as-

Given

$$Q = \{q_1, q_2, \dots, q_n\}$$

$$D = \{d_1, d_2, \dots, d_n\}$$

If $(q_i, d_j) = 1$: relevant

If $(q_i, d_j) = 0$: non-relevant

2.10 Connectivity-Based Ranking

Web crawling gather useful web pages as efficiently and quickly as possible. It possesses features like robustness and politeness. The crawler can be executed in a distributed fashion, scale the crawler rate with extra bandwidth, efficient, access quality pages with freshness of fetched pages, and should be extensible.

User queries are categorized into informational, navigational, and transactional. Informational queries seek to cover a broader area of specific topics and assemble information from different pages. Navigational queries try to find a single entity that contains the user expectation and avoid in plethora/number of documents. The best measure considered is user satisfaction having a precision value of 1. The transactional query deal with user transactions like purchasing and provide all services related to a particular operation.

Link analysis helps the user by performing the work of an indicator and helps to provide a direction as to which page to crawl next. The links on the web page follow the power law, where the total number of web pages with in-degree 'i' is proportional to $1/i^\alpha$, and ' α ' is 2.1. If there is a hyperlink from page 'i' to page 'j' then $A_{ij}=1$, and $A_{ij}=0$

otherwise. If 'V' links to a page having a high authority score, its hub scores increases while if page 'V' is linked to by good hubs, its authority score increases. Some studies recommend that the directed graph of the web page has 'bowtie shape', where three categories are mentioned as IN, OUT, and SCC. A user can switch from any page in IN to any page in SCC by succeeding certain hyperlinks and similarly, switching can be performed from page in SCC to any page in OUT and from any page in SCC to any other page in SCC. Remaining pages form into tubes that are a small set of pages link IN to OUT outside SCC, and tendrils that guide now wherefrom IN or from OUT. Figure 2.3 represents the Bowtie Structure of the web having one tube and three tendrils [14].

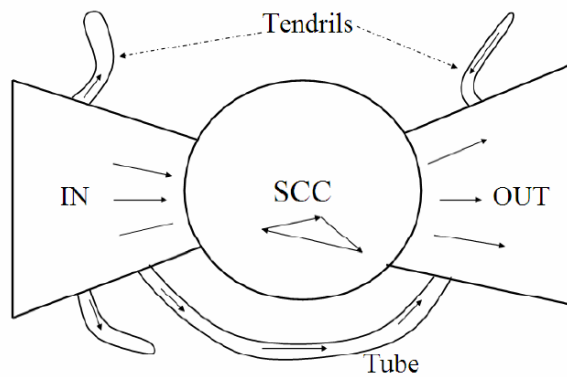


Figure 2.3 The Bowtie Structure of the Web [14]

Search engines based on accessing information are regarded as content-based and link-based. Search engine performances are greatly affected by the resulting quality of the documents that are retrieved. The result of the search is directly proportional to the ranking algorithm being used. A good ranking system provides efficient results for the user [51]. Two ranking factors govern the ranking algorithms that help in retrieving relevant information as per user query and are referred to as Query-dependent approach and Query-independent approach.

2.10.1 Query-Independent Ranking

The factors that are considered in Query-independent approach are not only the link measure of the pages but also take into consideration number of clicks a documents receive, the document length, the up-to-dateness of the document, what is the age of the document, what type of file is preferred, whether it is HTML document or pdf file, etc. It considers link popularity among the web pages regardless of the user query. Regarding the link based approach, two important algorithms are- Page Rank and Weighted Page Rank [45].

2.10.2 Query-Dependent Ranking

The Query-dependent consider various factors for ranking the web pages like word-document frequency, the position of the query term found within the document, the emphasis laid on the anchor text found, the language of the user query concerning the language of documents. It is based on user query term and regard word document, frequency, Inverse Document Frequency as a measuring unit. [45].

2.11 Various Ranking Algorithms

The web information is not structured like information that is found in the document collection and databases. The information is volatile and heterogeneous and exists in multiple forms. Different tools are required to manage the increasing flow of information so that from hundreds and thousands of documents, users can get only relevant documents having a high precision rate. Various ranking algorithms are proposed that help in finding user-relevant information.

2.11.1 Page Rank Algorithm

The increased number of inexperienced users on the web is rising, and the volume of information on the web is also rising exponentially, which makes a challenge for the

researches in the field of information retrieval. Link structure play a major role in accessing web pages and all the pages are connected through a link by which user access information. PageRank is one of the popular ranking algorithms used by Google search engine. Here, hyperlink to a page is regarded as a vote that counts and a page having a higher number of links as votes is weighted high in order of relevance [45].

The PageRank algorithm is drafted such a way that it can accommodate large data set and make systematic use of storage space for storing indexes. The data structure is maintained in a way so that the users can access the required information efficiently. It takes into consideration link structure and anchor text. A page is of prime importance in PageRank, if it is cited by many pages or is linked to an important page. It examines the back link for deciding the page score [52].

PageRank focus on the “random surfer model” where the user clicks on the pages randomly to search required content and when they get bored surf other pages. The ‘d’ damping factor is considered as the probability when the user switches to another page when he gets bored. PageRank considers that the sum of all the web pages should be one by considering the probability distribution over web pages and count links of all pages equally and normalize it.

The working of PageRank algorithm focuses on web page link structure and observe back link in determining the page score. The PageRank algorithm is evaluated as shown in 2.6.

$$PR (A) = (1-d) + d (PR (T1)/C (T1) + \dots + PR (Tn)/C (Tn)) \quad (2.6)$$

where-

A = a given page

T1 ... Tn = the pages that point to page A (i.e. citations)

d = damping factor which lies between 0 and 1

(usually we set d = 0.85)

C (A) = number of links going out of page A

PR (A) = the PageRank of a page A

The ranking is assessed using an iterative algorithm that corresponds to the eigenvector principal [51]. Page Rank suffers from topic drift problem where it sometimes points the user to a site that are off-topic and of no use. As it is query-independent measure so it ranks a site based on a vote that it receives. i.e popularity. PageRank has a problem in distinguishing between authoritative pages in general and that are authoritative specifically to query topic [53]. Sometimes in the PageRank algorithm, random jump from pages results in ‘dead-ends’ as many authorities provide no-way to jump to a node.

2.11.2 Weighted Page Rank Algorithm

Weighted Page Rank considers both Inlinks and Outlinks for accessing the popularity of page based on rank score and is an expansion of the Page Rank algorithm. It allocates a larger rank value for a popular page and the value of its score is directly proportional to its popularity [45]. Based on the popularity, the value changes correspondingly.

$W_{(v,u)}^{in}$, is the weight for link(v, u). Based on the number of inlinks for page u and number of inlinks for reference pages of page v it is measured using 2.7.

$$W_{(v,u)}^{in} = \frac{I_u}{\sum_{p \in R(v)} I_p} \quad (2.7)$$

where I_u is regarded as the number of inlinks for page u and similarly I_p for page p. The reference page list for page v is indicated by $R(v)$.

$W_{(v,u)}^{out}$, is said to be the weight for link(v, u) . Based on the number of outlinks for page u and the number of outlinks for all reference pages of page v it is evaluated using 2.8.

$$W_{(v,u)}^{out} = \frac{O_u}{\sum_{p \in R(v)} O_p} \quad (2.8)$$

where O_u is number of outlinks for page u and O_p is number of outlinks for page p . $R(v)$ is the reference page list of page v . The original PageRank formula is modified and elucidated in 2.9 [54]

$$PR(u) = (1 - d) + d \sum_{v \in B(u)} PR(v) W_{(v,u)}^{in} W_{(v,u)}^{out} \quad (2.9)$$

2.11.3 HITS Algorithm

The main concern of the present era is to enhance the quality of search techniques which is concerned with human satisfaction. The searching process is one of the initial steps of the user-generated query. It is a strenuous task for the users to digest the result of the query as an abundance of pages is retrieved as a result of the query. Huge numbers of links are produced and many of them are just for the navigational task which does not provide relevant sources so it becomes quite a difficult judgment to maintain a balance between the relevance of the document and their popularity. Not all the popular documents are relevant to some extent and the problem arises for filtering the relevant documents.

HITS algorithm works on the link structure of the graph and attempt to maintain a balance between hub and authority. Pages that are connected to maximum nested authority pages are referred as Hubs. In HITS first searching is performed based on text and the numerical score is calculated for a small sub graph. In HITS the approach followed is to try to identify the social organization in which hubs are related to authorities [55].

The ranking of the page is calculated at query time and often the page operate both as Hub and Authority [56]. The Hub weight (H_q) and the Authority weight (A_q) is evaluated using the formulae as given in 2.10.

$$H_p = \sum_{q \in I(p)} A_q \quad A_p = \sum_{q \in B(p)} H_q \quad (2.10)$$

Where,

H_q = Web page score of hub

A_q = Web page score of Authority

$I(p)$ = For page p the set of reference pages B

$B(p)$ = set of referrer pages of page p

For a web page the hub weight is equal to sum of authority weights of pages that it links to, similarly authority weight is equal to sum of hub weights of all the pages that it links to.

2.11.4 Weighted Link Rank Algorithm

There are three parameters which are the focal point of this algorithm- tag where the link is found, relative place for the page and anchor text length and it give weight value for the links. In search engines the precision value is enhanced by the weighted links and length of anchor text is reviewed as the finest characteristic while relative position is not very useful. By modifying the weight factor for every term additional modification can be accepted. [57], [58].

2.11.5 EigenRumor Algorithm

This algorithm performs eigenvector calculation of adjacency matrix for links by weighting hub and authority score of blog entries. Blogs are a mean of communication tool where new topics can be discussed by the blogger and those entries which comprise of new topics of relevance should have a high score. Eigenvector calculation is performed by scoring each blog entry. The weight of the hub and authority score of bloggers is

considered for calculating the score and it assigns a high score to a good blogger [57], [67].

It has been observed that PageRank and HITS algorithms utilized for ranking web pages do not provide correct ranking to the blog entries as sometimes blog entry is very small and even there are few links to a particular blog. In such cases, many of the blogs which have high score in the past suffer from ranking mechanics. These problems can be handled by EigenRumor algorithm. The algorithm helps to rate new blog entries using structural links even though it has no in-links based on past behavior. Here the concept of agent and object is used to represent a relationship where an agent is designated as a blogger and an object is the blog. To calculate the weighted Score of object, EigenRumor algorithm, consider hub score and authority score regarded as ‘Information Evaluation and Information Provisioning’ Figure 2.4 depicts EigenRumor Community Model [59].

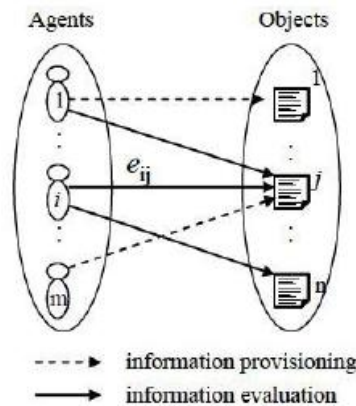


Figure 2.4 EigenRumor Community Model [59]

2.11.6 Distance Rank Algorithm

The distance among the pages is considered as punishment and is focus on reinforcement learning. It focuses on the amount of average clicks among the pages which is regarded as distance. The focus is to have a high ranking which is achieved by minimizing the distance between pages. There has been an exponential growth in the number of web pages and web servers and to process these pages is a hectic task.

It is a challenge for the search engines to retrieve quality pages from the web servers which is defined by user preferences and requests and an efficient ranking algorithm is needed to accomplish this task. For sorting pages in order of relevance, good ranking criteria are required and the ranking algorithm should have high precision and low complexity. The Distance Rank algorithm focus on maintaining low distance between the pages so that the pages will have a higher ranking i.e. distance is inversely proportional to ranking. To travel from a page 'i' to page 'j', minimum numbers of average clicks are considered. The algorithm helps in retrieving high- quality pages in minimum time duration by considering the minimum distance among pages. It is considered that having many inlinks to a particular page increases its ranking, similarly if a page has many inlinks the distance should be minimum which overall increases the ranking score. It is an iterative algorithm with high convergence speed and having less iteration [60].

2.11.7 Time Rank Algorithm

It employs a time factor for refining the precision of the ranking algorithm. The accuracy of ranking can be enhanced by using the time factor which depicts the interest of the user on a particular page and long stay on a page increases the ranking score. It is considered as a combination of both context and link structure.

Time Rank is considered a topic related algorithm where the user visits a page of his interest and spent time on a particular page which can be accessed from the search engines. The longer the duration user spent on a page, the more important the page is and so its ranking will be increased. Longer visited pages will get a higher score, which is accessed from the server log files that are processed to retrieve the desired fields [61].

2.11.8 TagRank Algorithm

It is established on the concept of social annotations for page ranking. Users can annotate web resources of their interest and tag their interest of web resources. It is estimated that the more time bookmarks are annotated the more self-identity corresponding web page will get. The more annotation the page gets the more superior it is, and the new

information resource is indexed more effectively while using the TagRank algorithm. The method helps in increasing the user satisfaction for obtaining the information. The algorithm can be improved by observing the semantic relationship which is reviewed among co-occurrence tags. The annotation factor and time factor of the tag are considered for achieving better results [62].

2.11.9 Relation Based Algorithm

The relation based ranking algorithm is utilized in conjunction with the Semantic web search engine and rely on information taken from the user query. Due to the massive growth of the web, it is very tough to extract information from the repository and required resources using only the hyperlink structure. The presence of unwanted page by the search engine iterates the procedure of refining the query and reprocessing of retrieved information. The refinement of the result is a time-wasting process and normally search engines do not process relation-based infrastructure to handle semantic annotations for web pages. In association with other established ranking strategies, semantic web search engines ranking principle work at the inner level and enhance the query result and minimize the complexity of the ranking task.

The semantic web help in enhancing the search result of the user query without the need for tiresome manual refinement. The ranking strategy provides a significant score for the web page by considering –user query page annotation and ontology used, which achieve promising output in terms of time and accuracy [63].

2.11.10 Query Dependent Ranking Algorithm

A structure for Query-dependent ranking is proposed that calculates similarity between queries and ranking model for training query with documents is considered. In the framework, a ranking model is build for the training query and its allied documents in the ranking model construction phase where queries are ranked based on score. After this model, selection of high performance is done in model selection mechanism and for model combination, a weighting procedure is described. Between two queries- cosine,

distance between starting points, and the proportion of lengths is considered for measuring similarity [64].

2.11.11 Ranking and Suggestive Algorithm

It considers the problem faced in ranking the popular items and how the suggestion is provided based on user feedback for popular items. It has been found that giving suggestions for items can overall affect its popularity by raising its level or lowering its ranking. These types of problems are faced in query searching and tag suggestions.

The proposed algorithm helps the user in suggesting popular items that help in learning of user's true preference over items. The randomized algorithm focuses on understanding the limitations of ranking and helps in assessing the quality of the suggestions which is measured by the popularity of suggested items [65].

2.11.12 Comparison Score Based Algorithm

Ranking of items is one of the frequent tasks where pairwise comparison can be operated. The items are provided a score based on the order in which they appear and then finally comparison is done on the total score reached between a pair of items.

A metric referred to as Kendall distance is specified for comparing score vectors, which also consider the space found between the pair of scores. The focus is done on studying the ranking of items in two ordered lists and a measure is provided for analyzing how the position of items are differed in the provided list and is zero when ranking is the same, and maximum when found in opposite direction [66].

2.11.13 Algorithm for Query Processing in Uncertain Databases

It considers the ranked queries in uncertain data management which are found in various applications like decision making, recommendation system, and various data mining tasks. It can be used in distinct fields like monitoring of sensor data and tracking of mobile objects [67]. The probabilistic ranked query (PRank) approach is accelerated by

using a novel approach and to reduce the search space-effective spatial and probabilistic pruning is introduced. The two pruning heuristics is integrated into the query procedure. It has been found that the query extract 'k' uncertain objects having the highest score as expected. The J-PRank query on two join uncertain databases is proposed and ranking is conducted. The given approach is effective and efficient in terms of wall clock time [68].

Table 2.2 gives a comparative summary of ranking algorithms, which are compared based on their excellence, restriction, and appropriateness. The order of the algorithm is defined on the procedure used to rank the web pages [69].

Table 2.2 Comparative Summary of Ranking Algorithm

Ranking Algorithm	Author	Technique	Relevancy	Merits	Limitations
Page Rank Algorithm	Sergey Brin and Larry Page	WSM	>HITS	Efficient, fast, ranking based on importance of page	Result at indexing time not at query time.
Weighted Page Rank Algorithm	Wenpu Xing and Ali Ghorbani	WSM	<PR but >HITS	Accuracy high	Based on popularity of page, efficiency problem.
Distance Rank Algorithm	Ali Mohammad Zareh Bidoki, Nasser Yazdani	WSM	Moderate	Pages found quickly with high quality	Problem to calculate distance if new page inserted
Relation Based Algorithm	Fabrizio Lamberti, Andrea Sanna and Claudio Demartini	WSM	High	Efficiently manage search page as it is keyword based	Every page have to be annotated which is complex task
EigenRumor Algorithm	Ko Fujimura, Takafumi Inoue and Masayuki Sugisaki	WCM	High for Blog	Useful for ranking blog	Suited for blog ranking
Tag Rank Algorithm	Shen Jie, Chen Chen, Zhang Hui, Sun Rong Shuang, Zhu Yan and He Kun	WCM	Less	Ranking result are exact	Comparison based and require more site as input
Query Dependent Ranking Algorithm	Lian-Wang Lee, Jung-Yi Jiang, ChunDer Wu, Shie-Jue Lee	WCM	High	Give result for user query	To calculate similarity limited number of characteristics is used
Hypertext Induced Topics Search	Jon Kleinberg	WSM & WCM	<PR	Highly relevant pages are searched	Less efficient and topic drift problem

2.12 Performance Evaluation Measures

For judging the functioning of the IR model various evaluation measures are used which stick to a particular query or can include the whole session [12]. A document is considered as appropriate or inappropriate and this notion is examined for evaluating the Information Retrieval System. A document is considered appropriate if it addresses the given information and meets the requirement of the user.

For accessing the usefulness of the IR system precision and recall are frequently used as an evaluation measure. Precision, Recall & F-measure are the most commonly used performance evaluation measures. It is a means by which we can judge which model or algorithm is best in the current scenario for solving a problem [10]. Relevance feedback for Information Retrieval help in improving the final result set based on feedback collected for relevant documents. It can be used to refine both recall and precision. The ranking qualities of the algorithms can be evaluated using Cumulative Gain (CG), Discounted Cumulative Gain (DCG), Idealized Discounted Cumulative Gain (IDCG), and Normalized Discounted Cumulative Gain (NDCG).

2.12.1 Precision

Precision and Recall are utilized for accessing the quality of unordered retrieved documents. Both are considered as a set-based measure for ranking documents. For evaluating ranked lists precision is plotted against recall. The precision for a relevant document is assumed to be zero if a relevant document is never gets retrieved. It helps to specify what fractions of returned results are relevant to the information need. It is determined as the fractional value of the recovered documents that are significant [14]. Precision help in presenting only the relevant items and is evaluated as shown in 2.11.

$$\text{Precision} = \frac{\#(\text{relevant items retrieved})}{\#(\text{retrieved items})} = P(\text{relevant}|\text{retrieved}) \quad (2.11)$$

Precision at k is evaluated for estimating precision at set low levels of recovered results and does not consider any approximation of the size of the set of relevant documents. For a good system, it is considered that precision tends to decrease as the number of documents retrieved keeps on increasing. Most of the surfers of the web want relevant results (high precision) on the first page.

Precision@k (Precision at k) is evaluated up to the kth prediction and can be processed at a different level as shown in 2.12.

$$\mathbb{P}recision@k = \frac{true\ positives\ @k}{(true\ positives\ @k) + (false\ positives\ @k)} \quad (2.12)$$

2.12.2 Recall

It specifies what fractions of significant documents in the collections were retrieved by the system. It is described as the fractional value of relevant documents that are retrieved [14]. Recall is referred to as the capability of the system which presents all the appropriate items but is difficult to measure on the web and is evaluated as shown in 2.13.

$$Recall = \frac{\#(relevant\ items\ retrieved)}{\#(relevant\ items)} = P(\text{retrieved}|\text{relevant}) \quad (2.13)$$

Similarly, Recall@k (Recall at k) is estimated only up to the kth prediction level and can also be processed at different levels as shown in 2.14. Search engines frequently use precision at top k, e.g., k = 10.

$$Recall@k = \frac{true\ positives\ @k}{(true\ positives\ @k) + (false\ negatives\ @k)} \quad (2.14)$$

Table 2.3 displays the contingency notion of accessing the accuracy of the Information Retrieval System by using two actual classes relevant and non-relevant and specifies recall and precision in these terms and as shown in 2.15.

Table 2.3 Contingency notion of accessing Accuracy

	Relevant	Nonrelevant
Retrieved	true positives (tp)	false positives (fp)
Not retrieved	false negatives (fn)	true negatives (tn)

$$\begin{aligned}
 P &= tp / (tp + fp) \\
 R &= tp / (tp + fn)
 \end{aligned}
 \tag{2.15}$$

For a good system, the value of precision normally decreases, if the number of documents recovered is increased. Some amount of recall can be tolerated under a definite percentage of false positives.

2.12.3 Harmonic Mean (F-Measure)

F-measure is regarded as a tradeoff between Precision and Recall and is a weighted harmonic mean rank based metric that is also referred as a mixed metric that takes into consideration precision and recall and measures the portion of recovered elements which are appropriate and is summarized as shown in 2.16.

$$F = 2 * \frac{p * r}{p + r},
 \tag{2.16}$$

2.12.4 Average Precision (AP)

This metric evaluates how much relevant documents are concentrated in the highest-ranked predictions. It notifies about how accurate a model's ranked predictions are for a single example and is shown in 2.17.

$$AP = \frac{\sum_{n=1}^N (p@n * rel(n))}{\text{Number of relevant document for Query}} \quad (2.17)$$

2.12.5 Mean Average Precision (MAP)

It tells how much a model's ranked predictions are correct, on average, over a whole validation dataset and is regarded as a macro-averaging function where each query counts equally. It is a simple average of AP over all examples in a validation set and is shown in 2.18. It is found to be adequate for web search and suppose that users are concerned in discovering different relevant documents for a particular query and requires numerous relevance judgments in a text collection.

$$MAP = \sum_{n=1}^N AP_n \quad (2.18)$$

2.12.6 Cumulative Gain (CG)

Gain is normally collected starting at the top of the ranking and can be decreased, or discounted, at lower ranks. It simply adds the ratings up to a specified rank position and is shown in 2.19.

$$CG_n = \sum_{i=1}^n rel_i \quad (2.19)$$

2.12.7 Discounted Cumulative Gain (DCG)

It is a general measure for assessing web search and related tasks and emphasize on retrieving highly relevant documents by using graded relevance as a standard of usefulness, or gain, by investigating a document. DCG is regarded as a weighted sum of the degree of relevancy for the ranked items. It is known as discount as weight is a

decreasing function of rank and the probability of viewing a document decreasing with its rank value.

In DCG the total gain is assembled at a particular rank p and the queries that return a huge result sets will probably always have higher DCG scores than queries that provide small result sets. It considers two assumptions – First, the highly appropriate documents are considered to be helpful than marginally relevant documents, and second the lesser the rank of a relevant document, the less likely it is examined by the user.

The DCG help in returning absolute values and does not allow comparison between the given queries. It is ineffective in gauging the performance of a ranking model on a whole validation dataset. It is represented as shown in 2.20.

$$DCG = \sum_{i=1}^n \frac{rel(x_i)}{\log_2(i + 1)}, \quad (2.20)$$

2.12.8 Normalized Cumulative Gain (NDCG)

NDCG is highly approved in evaluating Web search and provide a great advantage in comparing various ranking evaluation measure. NDCG is a normalization factor of Discounted Cumulative Gain (DCG) measure which focuses on discount function while others consider uniform weight. It helps in depicting the graded relevance for retrieved documents that show the degree of relevancy while other measures allow only binary relevance where documents are considered as relevant or non-relevant. It presents a discount function over rank that helps to rank documents concerning other measures that weight all positions uniformly. NDCG is used to normalize DCG by IDCG which is the DCG of best possible results based on best ranking measure and NDCG is always between 0.0 and 1.0 [70],[71],[72] and is shown in 2.21.

NDCG is utilized when we want to differentiate the ranking of one result set with another ranking, with potentially fewer elements, different elements, etc. which is not possible using DCG as the query result may vary in size and it unfairly penalizes queries that return long result sets. NDCG help in normalizing a DCG score, by dividing the finest possible DCG at each threshold [14].

$$\text{NDCG}_n = \frac{\text{DCG}_n}{\text{IDCG}_n} \quad (2.21)$$

2.13 Optimization Techniques

Optimization is the process that helps in making a system more efficient by using minimum resources. We can choose different algorithms and methods which help in making the process more efficient in terms of both space and time. Optimization helps in decisions making and analyzing a system. To find the best solutions from a set of feasible solutions is the main goal of optimization. It is the process to achieve the best-desired result on given constraints under different circumstances to acquire maximum benefit in minimum effort. There are different ways for solving the optimization problems efficiently-

- Mathematical programming methods which under a prescribed set of constraints are useful for finding minimum function that comprises of several variables.
- Stochastic techniques help in examining the problems that are specified by a set of random variables of known distribution.
- Statistical methods are utilized for the construction of empirical models and are used in the analysis of experimental data [73].

Due to rapid emergence in science and technology, optimization has received enormous attention. The optimization problem comprises of objective function, constraints, and a choice variable which overall help in the improvement process [74].

2.13.1 Nature Inspired Algorithm

Nature is always a source of inspiration for all living beings and it inspired the researchers in different ways to draw efficient and effective results of hard problems in a constrained time interval. There is a great drift from nature to technology and nature in these algorithm acts as the sole component for solving complex problems, thus giving good results by using a simple process that is completely drawn from nature. Some of the

problems that are found in the real-world are NP-hard problems that are difficult to deal with, so in this scenario optimization algorithms can be used to handle the complex problems [75]. It is a new era of computation that solves natural meta-heuristics problems. The Nature-inspired algorithms are classified into-Stochastic algorithm, Evolutionary algorithm, Physical algorithm, Probabilistic algorithm, Swarm algorithms, Immune algorithms, and Neural algorithms as shown in Figure 2.5. These entire algorithms cover a broad area ranging from random search to genetic algorithm, from memetic algorithms to bees algorithms and also span to back-propagation techniques.

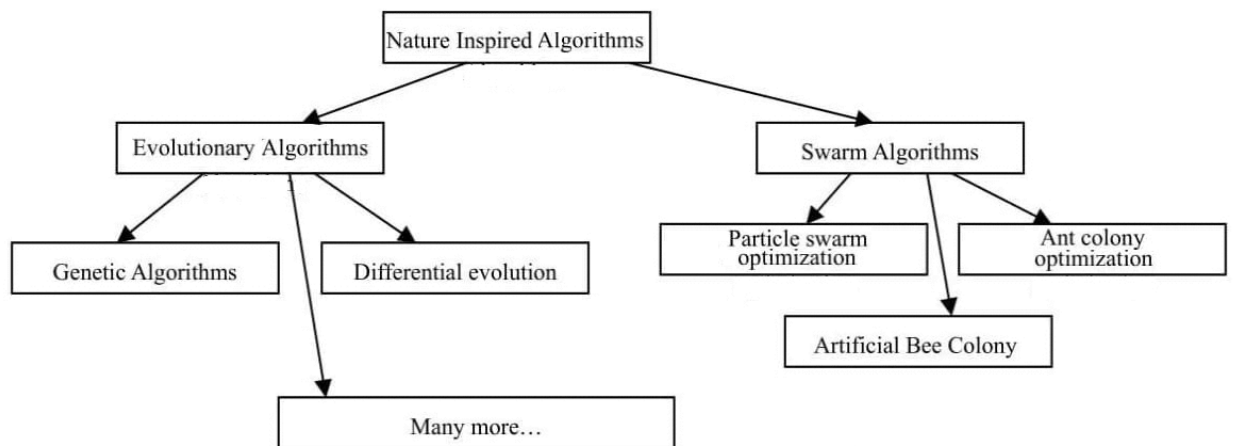


Figure 2.5 Nature Inspired Algorithms

The Nature-inspired algorithms are reviewed in a variety of fields and help to solve different problems where conventional optimization algorithms usually fail to produce a result within the search space and stuck within the local optima by not observing the global perspective. The real-life scenario is imposed with not only timing constraints but also other constraints, and in such a scenario these Nature-inspired algorithms help to produce efficient results for multidimensional problems. These are inspired by the behavior of swarm, bees, ants, moth, bat, firefly, etc. The optimization algorithm helps in providing better results from the number of given choices by following a stochastic approach.

Evolutionary Algorithms are iterative based algorithms that select an initial population from the search space and perform crossover and mutation to reproduce a novel population that is better than the parents. The fitness of the individual is calculated in each iteration and by considering the fitness value that are generated the best individuals are chosen for the mating process. Genetic Algorithm is a stochastic optimization algorithm that follows the Darwin theory of evolution for generating offspring that are better than the parent chromosomes. Selection, crossover, and mutation are basic genetic operators that are employed [76]. Differential Evolution is also a population-based algorithm that helps in solving global optimization problems having a few control variables that help in effortless implementation. It can also be used in information retrieval process and considered an ideal for learning a ranking function [77].

Swarm Algorithms collectively capture the behavior of organisms like birds, fish, ant, bees, firefly, etc. The movement of the species in a particular direction for finding the target source helps to evaluate a possible solution in the search space. Swarm intelligence is considered a sub-field of Artificial Intelligence. Particle Swarm Optimization (PSO) follows the behavior of the birds and each bird is treated as a particle that searches for the food particle and moves in the direction of food with a specific velocity for finding the target solution. During each iteration, the particle moves closer to its destination which is calculated by using an objective function, that tells about the fitness of the solution in the search space, pbest and gbest values are evaluated during each iteration [78]. Ant Colony Optimization is also a search-based technique for finding the solution which focuses on the behavior movement of the ants. They help in discovering the shortest path to reach the desired goal by interacting and coordination among themselves using pheromone that helps to trace the straight path to the food source [79]. Artificial Bee Colony is established on the intelligent behavior of the honey bees, which look for the food source in the local and global search space using employed bees, scout bees, and onlooker bees. The employed bees execute the task of collecting nectar from the food source while onlookers search for other food sources. Tracing the position of the food is evaluated by the fitness value which keeps on updating when the new source is found and the direction of the target source is communicated to others bees using the waggle dance [80].

These entire nature-inspired algorithms use computational intelligent techniques for optimizing the problem. Other optimization algorithms are designed for a specific task as all algorithms are ineffective in providing a feasible solution according to the problem domain and can sometimes increase the complexity of the solution space by sticking on the local optima.

2.13.1.1 Evolutionary Algorithm

Computational models of the evolutionary process are regarded as the key elements of evolutionary computation, which help in designing and implementing computer-based problem-solving systems. Evolutionary computational models simulate the evolution of individual structures using the process of selection and reproduction and rely on the performance (fitness) of the individual structure. Selection of the individuals depend on their fitness value thereafter recombination and mutation perturb individuals which provide a common heuristics for exploration [81].

Evolutionary algorithms progresses from generation to generation using evolution, selection, recombination and mutation and is captured as-

```
Procedure EA; {  
    t=0  
    initialize population P(t);  
    evaluate P(t);  
    until (done) {  
        t=t+1  
        parent-selection P(t);  
        recombine P(t);  
        mutate P(t);  
        evaluate P(t);  
        survive P(t);  
    }  
}
```

The procedure of selection is a two-step process- parent selection and survival. On one hand, the parent selection decides who becomes parent, and children are generated via recombination that exchange information between parents and mutation, and finally children are estimated. The survival step helps in deciding who will survive in the population. Fitness is the central concept of all evolutionary algorithms and for maximizing fitness the focal point will be on focusing the concentration on higher fitness and this concentration is referred to as exploitation and is the task of selection. In ranking selection, the parent is uniformly randomly selected and does not consider the relative ordering of fitness and only the rank of the individual is important [81].

2.13.1.1.1 Genetic Algorithm

For solving search and optimization problems the most common adaptive procedure is the Genetic Algorithm, that focuses on the genetic processes of biological organisms. By mimic the process of natural selection, Genetic Algorithm help in evolving solution for the real-world problems. The basic principle of the Genetic Algorithm was laid by Holland and helps to simulate those processes that are subjective to evolution. Genetic Algorithm help in handling a wide range of problems in different areas and is a robust technique that does not guarantee an optimal solution, but is good at finding ‘acceptable good’ solution. It is appropriate for extracting information from the search space that has robustness and fast search capabilities [82].

The underlying notion of Genetic Algorithm (GA) is planned in a way that it helps in simulating the process of natural systems and is vital for evolution that follows the principles of survival of the fittest. It is normally used where classical search methods cannot be used easily and the search space is bound to be huge [37]. As in nature, the species compete with each other for accessing resources and also compete to attract a mate. The fit species will spread to successive generation and help in producing superfit offspring having high fitness value suited to the environment concerning weak individuals whose offspring will decline in the coming generation.

Genetic Algorithm work with a set of the population of individuals which are allotted a fitness score accordingly. The solution is provided for a given problem, and the highly fit

individuals are crossbred with other individuals for producing a new population which carry forwarded some traits of their parents and the weak ones will be excluded in the coming generation [83]. The process will lead to producing good individuals and this proportion of good characteristics will be forwarded to further generations and new good characteristics are spread throughout the coming population. This process helps in exploring new search space and an optimal solution to the problem is achieved. A good random search considers exploration by excluding exploitation, and the Genetic Algorithm try to maintain a balance between the two.

Standard Genetic Algorithm comprises of following steps-

```
BEGIN
    Generate initial population
    Compute fitness of each individual
    While NOT finished DO
        BEGIN
            FOR population-size
                BEGIN
                    Select two individual from old generation for mating
                    Recombine the two individuals to give two offspring
                    Compute fitness of the two offspring
                    Insert offspring in new generation
                END
            IF population has converged THEN
                Finished=TRUE
        END
    END
END
```

(A traditional Genetic Algorithm)[83]

Solution comprises of a set of parameters which are referred as genes attached mutually to form string known as chromosomes. The set of parameters comprised of the chromosome is referred to as genotype, which has information for constructing an

organism and is referred to as phenotype and fitness depends on the performance of phenotype. The fitness function returns a numerical value and is devised for each problem which is proportional to the ability of the individual. During the reproduction, individuals are chosen from the population and re-synthesize to produce offspring for the next generation. The parents are randomly chosen which comprises of good individuals and the poor ones are left out. The parent chromosomes are re-synthesize following the mechanism of crossover and mutation. During crossover, individual chromosomes are cut at random position and then swapped to generate new chromosomes. It might not be applied to all chromosomes, a random choice is taken where the likelihood of crossover is between 0.6 and 1.0. After performing crossover, mutation is applied that randomly modify gene with small probability (0.001) [83].

Genetic Algorithm focuses primarily on the crossover operator and most common among them is one-point, multi-point and uniform. In one-point crossover, a cut-point inside two parents is inserted and information before the cut-point is swapped among two parents. In multi-point, a higher number of cut-points are made and information is then swapped. Cut-points are not used in uniform crossover, instead a global parameter indicates the likelihood that every variable should be interchanged among two parents [81]. In Tournament selection, a pair of individuals are picked randomly and the pair having high-level of fitness is copied into the mating pool and the process is recurred until the mating pool is full [83]. Mutation introduces a new material into the population by randomly changing a gene in the chromosome. The final termination is made on the process when the value remains the same over successive iterations or when a fixed number of iterations are reached. GA is a computational effective and adaptive method and is usually employed where search space is of high-dimensionality and mathematically computed. Clustering is an unsupervised problem that handles a large volume of data and GA can be employed in this area [84], [85], [86], [110].

2.13.1.2 Swarm Algorithm

Swarm Intelligence has been used for the collective behavior of self-organized systems which are natural or artificial. It normally comprises of agents that interact with each other, by following simple rules that are inspired by nature. Swarm Intelligence extracts collective intelligence as a behavior and is assessed as an adaptive strategy that specifies how an individual should behave. It follows self-organizing, co-evolution, and simple rules which are applied in different domains like searching, optimizing, planning, and research work. [74].

2.13.1.2.1 Artificial Bee Colony

For solving high-dimensional problem nature-inspired optimization techniques are more preferred in comparison to classical optimization methods. Various optimizations algorithms are present that helps to intercept with the complexity of the real-world. The social behavior of various species like birds, fish, and insects leads to the rise of swarm intelligence algorithms that help in solving various optimization problems. Artificial Bee Colony (ABC) depends on the foraging behavior of honey bee and is considered as an optimization algorithm.

ABC is inspired by the intelligent foraging behavior of honey bees and is a population-based search optimization process where artificial bees search for places that comprise of different food sources and try to achieve the source that has high nectar content. The Artificial Bee Colony (ABC) is a swarm-based meta-heuristics algorithm inspired by the intelligent behavior of honey bees and is a recent nature-inspired algorithm like Particle Swarm Optimization (PSO), Differential Evolution (DE), and Ant Colony Optimization (ACO) that is considered for optimization purpose. Tereshko Artificial Bee Colony (ABC) specifies two basic phases for the food source- Discovery of food source and collecting food from the food source.

The first phase comprises initializing the scout bees to look for the food source, discovering the food sources within their reach, and evaluating their fitness so that the best food source can be traced easily. The second phase deals with the recruitment of the

scout bees who search the local area for the food source, evaluating their fitness and rejecting the site if required and then search the global area for new good sources. The food sources are searched in a multidimensional search space where employed and onlooker bees search for food sources in local search areas while onlookers and scouts cover the global search area and by this, a balance is maintained between the exploration and exploitation process. In ABC the number of food sources (solutions) equates to the number of employed bees and the association of employed bees is with only one food source.

ABC model has three main components - Food Sources where the bees evaluate the richness of food, taste, and ease of extraction as major parameters to choose the best food source. The distance from the hive is also considered for accessing the desired target, Employed foragers who associate themselves with the current food source being utilized and inform other bees about the source distance and its profitability and Unemployed foragers which can be scout or onlooker that search for the food source to exploit and randomly explore the environment for finding new food sources [87].

ABC randomly tries to discover a good solution by iteratively improving the strategy of searching the neighbor and moves towards a better solution goal by abandoning poor solutions. Exchange of information among the bees is normally communicated through waggle dance.

In relation with other swarm intelligence-based algorithms, ABC is an iterative approach where the fitness of the food source is judged by the quality of nectar in the food source. Four phases are required in ABC- initialization, employed bees phase, onlooker bees phase, and scout bee phase to perform its desired goal. The food source is produced randomly during the initialization phase using a specified equation, in employed bees phase the new position of the food source is searched and the value is updated if it is found to be better. In onlooker bees phase the fitness information of updated food source and its position is shared with the onlooker bees which then select food source based on probability. Finally, the Scout bees phase started if it is found that there is no change in the food source, which is then discarded and a new search is started [88].

The general scheme of the ABC algorithm is as follows:

```
Initialization Phase
REPEAT
  Employed Bees Phase
  Onlooker Bees Phase
  Scout Bees Phase
  Memorize the best solution achieved so far
UNTIL (Cycle=Maximum Cycle Number or a Maximum CPU time)
```

The algorithm tends to produce improved results by increasing the colony size, but after reaching a sufficient point any improvement does not make any effect in overall performance. ABC handle smoothly the increasing dimension of the problem and help in optimizing numerical functions, it becomes more efficient on increasing number of variables.

2.14 Summary

In this chapter, a survey for related work is conducted that is relevant for this thesis. The chapter depicts information about the Information Retrieval Process and highlighted the role of ranking algorithm in the retrieval process. Various ranking algorithms are studied and a brief overview of the Nature-inspired algorithm is depicted which are considered for optimizing the ranking process.

DESIGN AND IMPLEMENTATION OF SCRANK ALGORITHM FOR RANKING WEB PAGES

3.1 Introduction

Owing to sudden surge of information on the World Wide Web there is difficulty in finding out the relevant web pages that focus on the user query and maximum time is spend in retrieving the needed information. Internet normally comprises of an abundance of information and people explore their areas to get the relevant information. It has been found out that the core component of any search engine is the ranking framework which work on user queries and rank web pages. A good ranking system should not be based on single criteria but can take input from multiple points to create a comprehensive ranking system. It can be pointed out that the Information Retrieval process can be improved by incorporating the process of both content similarity and link analysis. The aim of the dissertation is to explore the ranking algorithm. This chapter proposed the scRank algorithm that helps in web pages ranking quite efficiently. Based on user query, the web pages are clustered that undergoes certain conditions and thereafter ranking is applied to them. The illustrated example given demonstrates the working of the model scRank. The proposed approach helps in reducing the complexity by reducing the number of iterations and rank documents in minimum time duration.

3.2 Related work

Different works by various researchers have pointed out that various measures are used for ranking the results of the query. Various approaches have been adopted for web page ranking that satisfies user yearning for appropriate information.

Shen et al., (2012) [89], presents a ranking method that focuses on query words, history, and search results. The relevancy of document content can be expressed on user performance. Anjusha and Nizar, (2019) [90], proposed a search technique that produces high-quality and relevant web pages by not only combining link structure but also the content of the pages. Better search results are provided concerning the generic search engine. Koo et al., (2019) [91], proposed a C-Rank algorithm that integrate collectively link information and content of a web page and helps in providing quick and accurate responses for user-generated queries. Based on the above works the proposed work also tries to focus on query words and combine both content and link of the pages, the work also tries to enhance the ranking work by pre-filtering the web pages. Gao and Xu, (2018) [92], presents an aggregation framework pRankAggreg that is based on unsupervised ranking and uses agglomerative hierarchical clustering which manages to cluster users that have related partial rankings. For computing joint ranking a weighted linear combination method is used that helps in assigning user weight. We also try to segregate the query results by clustering the web pages based on certain conditions that help in reducing computational work. Choudhary et al., (2018) [93], manages user profile using web search personalization method and help to personalize the search results. Static and dynamic preferences of the user are incorporated while Goyal et al., (2019) [94], proposes sNorm(p), which is a variant of SALSA, rank web pages efficiently. The work efficiently reduces the low authority hub weight calculation by combining the p-Norm of vector norms and SALSA. Sethi and Dixit, (2019) [95], proposed a new ranking mechanism for pages that undertake the browsing patterns of the user and focus on the links that are visited by the users. Like, Sheetal et al., (2019) [96], focus on closed pattern technique that helps in reducing the number of rules using frequent pattern mining technique which assist the user in accessing web pages based on certain patterns, the proposed work tries to group web pages based on query words. Guo et al., (2019) [97], tries to summarize the neural ranking models present status that consider raw text inputs and learn from it, and help in solving the ranking problem, while Overland and Juraev, (2019) [98], develops an automated algorithm for segmented string relative ranking that

create rankings of countries based on expert opinion survey using network theory and link analysis.

3.3 Proposed Methodology

The preferred approach scRank Algorithm is based on the ranking algorithm that is used to prioritize the web pages based on user queries which are generated by the search engine. It is proposed with the motivation to ease the task of retrieving the relevant web pages. Some ranking algorithm uses only the structural links for ranking purpose, but the motivation behind the given approach is to incorporate both structure and content of the nodes for formulating ranking score so that high precise results can be obtained and the user will get the pages that are rich in both content and link.

The main implication for the proposed work is to assist the lame users in finding the preferred information from the pool of data. Most of the users are unable to get the desired result of their query and keep on reframing the query to get the result, so the scRank ranking algorithm helps the users in obtaining their desired pages in less amount of time. It also employs both a content approach and structure approach and eases the task of information retrieval. The given work computes the cumulative weight of the nodes by taking into account the link structure and cumulative weight for the web pages. It undertakes the content of the web pages concerning user queries before applying the ranking algorithm. The given Query Q extract N pages that are filtered using different criteria and later ranking is applied to them.

The given approach helps in clustering the website pages toward the start of the procedure by checking the substance of the pages concerning the user query thereby eliminating the pages which lack proper content. It also considers the structural links among web pages. The PageRank consider structural links only and performs the computation for all the web pages while the scRank filters the page based on different criteria as stated in the algorithm and computation is done for selected web pages only. It helps in limiting the computational intricacy of the procedure and lastly helps in ranking of the nodes that are recovered.

3.3.1 Architecture

The architecture of the given approach is represented in Figure 3.1.

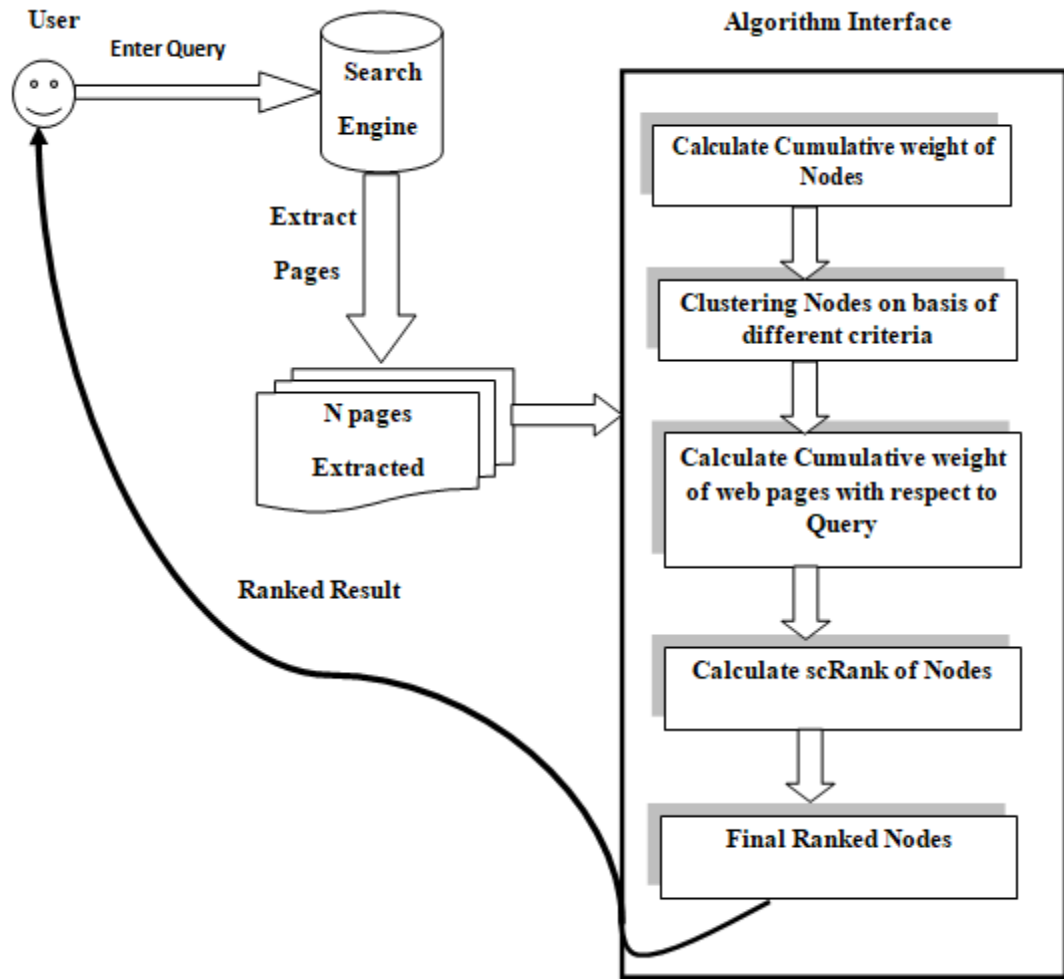


Figure 3.1 Architectural Model for scRank

3.3.2 Algorithm

The proposed scRank accepts user query and helps in ranking the web pages by following certain algorithm steps and is portrayed as-

Algorithm scRank:

Input: User Query

Output: Ranked Web Pages (WP)/documents

Steps:

1. Enter query Q consisting of 'n' terms that extract 'N' pages by removing stopwords.
2. Count number of terms 'n' of query 'Q'.
3. Calculate cumulative weight of the nodes using the formula-

$$CW(n) = I(w) / (I(w) + O(w))$$

where

CW(n)- cumulative weight of node

I(w)- Indegree weight of node

O(w)- Outdegree weight of node

4. Webpages/ documents are clustered on basis of following criteria-
 - Case I: If all term 'n' of 'Q' appear in WP_i where i=1, 2, 3,...M group them in C_i, where i=1, 2, 3.
 - Case II: If ' n-1 <= WP >= n/2 ' group them in C_i, where i=1, 2, 3.
 - Case III: If one term 'n' match group them in cluster C_i, where i=1, 2, 3, exclude.
5. Calculate cumulative weight of web pages with respect to the Query which are grouped into clusters using the formulas-

$$W(x) = N(x_i) / \sum_{i=0}^n N(x_n)$$

where

W(x) - Weight of node x

N(x_i) - Number of times x_i term of query occur in web page

N (x_n) - Total occurrence of all term in web page

$$CW_q(x) = \sum_{i=0}^n x_i$$

where

CW_q(x) - cumulative weight of query in x page

6. Calculate scRank of the web pages grouped in clusters using Formula-

$$scRank(N) = (1 - \infty) + CW_q(x) + CW(n) + [(rank(N1) / O(N1) + rank(N2) / O(N2) + \dots + rank(Nn) / O(Nn)^* \infty]$$

where

scRank(N) - rank of web pages N

α – constant term between 0-1, taken as 0.85 as standard value.

$CW_q(x)$ - cumulative weight of query in x page

$CW(n)$ - cumulative weight of node

7. Finally ranked web pages are retrieved.

3.3 Dataset Used

A dataset consisting of 8910 URLs are collected for testing the algorithm. The collected URLs are then tested on a set of different queries that generate a proper ranking result. A total of 100 queries are taken and are tested on the collected URLs. A small part of the dataset and queries employed for evaluating the ranking result is shown for illustration purpose in Figure 3.2 and Figure 3.3. The proposed work is experimentally performed on Spyder (Python 3.7).



Index	0
0	https://searcherp.techtarget.com/definition...
1	https://www.investopedia.com/terms/s/scm.asp
2	https://www.knowledgekafe.com/course/supply-chain-management-125022
3	https://en.wikipedia.org/wiki/Supply_chain_management
4	https://www.cio.com/article/2439493/what-is...
5	https://searcherp.techtarget.com/definition/supply-chain-management-SCM
6	https://en.wikipedia.org/wiki/Supply_chain_management#Importance
7	https://en.wikipedia.org/wiki/Supply_chain_...
8	https://en.wikipedia.org/wiki/Supply_chain_...
9	https://en.wikipedia.org/wiki/Supply_chain_management#Supply_chain
10	https://scm.ncsu.edu/scm-articles/article/what-is-supply-chain-management-scm
11	https://www.youtube.com/watch?v=1ZP05Rc1ZEo
12	https://www.ibm.com/topics/supply-chain-management
13	https://www.oracle.com/applications/supply-chain-management/
14	https://www.coursera.org/specializations/supply-chain-management
15	https://www.tradegecko.com/supply-chain-management
16	https://cscmp.org/CSCMP/Develop/Starting_Yo...
17	https://www.managementstudyguide.com/supply-chain-management-definition.htm
18	https://www.emerald.com/insight/publication/issn/1359-8546
19	https://hbr.org/2018/06/the-death-of-supply-chain-management
20	https://www.edx.org/micromasters/mitx-supply-chain-management

Figure.3.2. Dataset of web pages

Index	0
0	Google Driver Less Car
1	Micro Machining
2	Solar Powered Aircraft
3	High Speed Machining
4	Quad Copter
5	Non-Pneumatic Tyres
6	Electrochemical Machining
7	Dual Fuel Engine
8	Methanol Fueled Marine Diesel Engine
9	Intelligent Variable Valve Timing
10	Solid Waste Management
11	Hemi Engine
12	Scram Jet Engine for Hypersonic Flight
13	Sidewinder Missile
14	Laser Cutting System
15	Airless Tyres
16	Pneumatic Tyres
17	Chassis Frame
18	Pulse Detonation Engine
19	Solar Collector
20	Semi Active Suspension System

Figure.3.3. Queries tested on web pages

3.3.4 Experimental Analysis

The proposed work is executed by considering A, B, C, D, E, F, and G web pages. Figure 3.4 represents a directed graph of interlinked seven web pages.

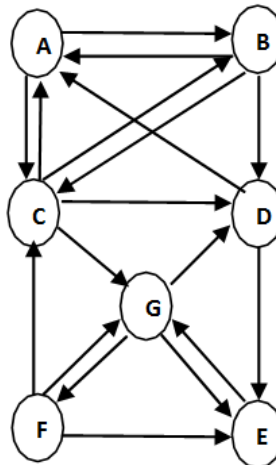


Figure 3.4 Hyperlink Structure of web pages

For a specified query the frequency of the terms among the different web pages is outlined in Table 3.1.

Table 3.1 Frequency of terms in Web Pages

Terms\ Nodes	A	B	C	D	E	F	G
Information	20	8	8	10	20	0	25
Retrieval	10	12	13	15	10	0	0
Evolutionary	12	0	6	8	5	0	0
Computation	8	0	2	6	2	6	0

The cumulative weight is calculated using formula 3.1 based on hyperlink structure for all the nodes and is given in Table 3.2 using formula-

$$CW(n) = I(w) / (I(w) + O(w)) \quad (3.1)$$

Table 3.2 Cumulative Weight for the nodes

Node	Inlink weight	Inlink weight	Cumulative weight
A	3	2	0.60
B	2	3	0.40
C	3	4	0.43
D	3	2	0.60
E	3	1	0.75
F	1	3	0.25
G	3	3	0.50

Based on the rules of the proposed algorithm specified in Step 4, the nodes are clustered. Case I is applied on nodes A, D, C, and E as all query words are found which are then

clustered together and Case II is applied on node B which will be placed in the second cluster. Case III is applied to F and G nodes that are excluded. After clustering, the cumulative weight of web pages concerning the Query is evaluated for Cluster 1 and Cluster 2 as is depicted in Table 3.3, Table 3.4 and Table 3.5 using formula 3.2.

$$W(x) = N(xi) / \sum_{i=0}^n N(x_i) \quad (3.2)$$

Table 3.3 Weight of each term of the query in (A)

Weight(x)	Value
W(x1)	0.40
W(x2)	0.20
W(x3)	0.24
W(x4)	0.16

Similarly, weight for pages B, C, D, and E are evaluated and finally, cumulative weight is commutated using the given formula in 3.3.

$$CWq(x) = \sum_{i=0}^n x_i \quad (3.3)$$

Table 3.4 Cumulative weight of nodes for query in Cluster 1

Cumulative Weight(x)	Value
CW(A)	1
CW(C)	0.97
CW(D)	1.08
CW(E)	0.99

Table 3.5 Cumulative weight of nodes for query in Cluster 2

Cumulative Weight(x)	Value
CW(B)	1

scRank formula is finally applied to the clusters and is given in the equation 3.4. The ranking of the nodes is presented in Table 3.6 and Table 3.7.

$$scRank(N) = (1 - \infty) + CWq(x) + CW(n) + [(rank(N1) / O(N1) + rank(N2) / O(N2) + \dots + rank(Nn) / O(Nn) * \infty] \quad (3.4)$$

The scRank values of nodes in Cluster 1 that are clustered together are depicted in Table 3.6.

Table 3.6 scRank of Cluster 1

Iteration\ Node	PR(A)	PR(C)	PR(D)	PR(E)
0	1	1	1	1
1	2.570833333	3.278104167	3.384930469	3.895262116
2	4.0685259	3.2674157	3.2409925	3.8340885
3	4.005081	3.8168261	3.3577422	3.8837071
4	4.1714493	3.8875326	3.3727673	3.8900928
5	4.1928601	3.8966322	3.374701	3.8909146
6	4.1956156	3.8978033	3.3749499	3.8910204
7	4.1959702	3.897954	3.3749819	3.891034
8	4.1960159	3.8979734	3.374986	3.8910357
9	4.1960217	3.8979759	3.3749865	3.8910359
10	4.1960225	3.89797623	3.37498661	3.89103598
11	4.19602259	3.89797627	3.37498662	3.89103598
12	4.19602261	3.89797627	3.37498662	3.89103598
13	4.19602261	3.89797627	3.37498663	3.89103598
14	4.19602261	3.89797627	3.37498663	3.89103598
15	4.19602261	3.89797627	3.37498663	3.89103598

The Table 3.7 depicts the scRank values of Cluster 2 that are clustered together.

Table 3.7 scRank of Cluster 2

Iteration\ Node	PR(B)
0	1
1	2.1875
2	2.1875
3	2.1875
4	2.1875
5	2.1875
6	2.1875
7	2.1875
8	2.1875
9	2.1875
10	2.1875
11	2.1875
12	2.1875
13	2.1875
14	2.1875
15	2.1875

The procedure can be halted as it is discovered that after Iteration 11 all the values of the nodes are found to be similar. Ranking of the last nodes is depicted in Table 3.8.

Table 3.8 Ranking of the Nodes

Ranking Order	Ranked Nodes
1	A
2	C
3	E
4	D
5	B

The proposed methodology provides result in less iteration and help in minimizing the computational complexity of the process. It helps in reducing the processing time and cost of the process in computing the rank of the nodes.

By following the original PageRank, it is found that till 27 iterations we are not getting

similar values, but following the proposed methodology the results are obtained after 11 iterations. The result of the scRank and PageRank on the number of iteration needed for obtaining the result is shown in Table 3.9.

Table 3.9 Iteration Required

	scRank	PageRank
Iteration required after which the desired result is achieved	11	27

The result of the standard method of PageRank calculation of all nodes is depicted in Table 3.10.

Table 3.10 PageRank of all Nodes

Iteration\ Node	PR(A)	PR(B)	PR(C)	PR(D)	PR(E)	PR(F)	PR(G)
0	1	1	1	1	1	1	1
1	1.070833333	0.817604167	1.120092014	0.9030074	1.100444812	0.433333333	1.446175421
2	1.003452212	0.814486743	0.930016212	0.988149391	1.102490972	0.559749703	1.44334152
3	0.99836318	0.771932797	0.951614393	0.979879115	1.133991137	0.558946764	1.474478775
4	0.987380975	0.771854973	0.946697406	0.987634427	1.145881868	0.567768986	1.486040666
5	0.989610073	0.77175748	0.95011678	0.99160929	1.153346683	0.571044855	1.494040539
6	0.991998383	0.773499129	0.952553442	0.994887179	1.157934579	0.573311486	1.499100253
7	0.994402744	0.775038772	0.954653739	0.99720331	1.160994733	0.574745072	1.502553879
8	0.996269645	0.776278519	0.956204617	0.998862661	1.163084667	0.575723599	1.504937134
9	0.997655692	0.77719715	0.957331215	1.000037597	1.16453652	0.576398855	1.506601934
10	0.998654721	0.77786114	0.958135255	1.000868279	1.165552575	0.576870548	1.507770086
11	0.99936675	0.77833461	0.958705664	1.001454617	1.166266392	0.577201524	1.508591819
12	0.999871306	0.778670258	0.959108977	1.001868246	1.166768785	0.577434349	1.509170524
13	1.000227902	0.778907516	0.95939372	1.002159943	1.16712269	0.577598315	1.509578308
14	1.000479604	0.779074997	0.959594604	1.002365623	1.1673721	0.577713854	1.50986573
15	1.000657159	0.779193146	0.959736276	1.00251064	1.167547904	0.57779529	1.510068343
16	1.000782372	0.779276467	0.959836172	1.002612883	1.167671838	0.577852697	1.51021118
17	1.000870661	0.779335217	0.959906607	1.002684966	1.167759209	0.577893168	1.510311879
18	1.00093291	0.77937664	0.959956265	1.002735787	1.167820806	0.577921699	1.510382873
19	1.000976797	0.779405845	0.959991276	1.002771616	1.167864232	0.577941814	1.510432924
20	1.001007739	0.779426435	0.96001596	1.002796877	1.167894849	0.577955995	1.510468211
21	1.001029554	0.779440952	0.960033362	1.002814686	1.167916433	0.577965993	1.510493089
22	1.001044934	0.779451186	0.960045631	1.002827241	1.167931651	0.577973042	1.510510628
23	1.001055777	0.779458402	0.960054281	1.002836093	1.16794238	0.577978011	1.510522994
24	1.001063422	0.779463489	0.960060379	1.002842334	1.167949943	0.577981515	1.510531712
25	1.001068811	0.779467075	0.960064679	1.002846734	1.167955276	0.577983985	1.510537858
26	1.001072611	0.779469604	0.96006771	1.002849836	1.167959036	0.577985726	1.510542191
27	1.00107529	0.779471386	0.960069847	1.002852023	1.167961686	0.577986954	1.510545246
28	1.001077178	0.779472643	0.960071353	1.002853565	1.167963555	0.57798782	1.5105474
29	1.00107851	0.779473529	0.960072416	1.002854652	1.167964872	0.57798843	1.510548918
30	1.001079448	0.779474154	0.960073164	1.002855418	1.167965801	0.57798886	1.510549989

In Table 3.11 the final ranking of the nodes are displayed.

Table 3.11 Ranked Nodes

Ranking Order	Ranked Nodes
1	G
2	E
3	D
4	A
5	C
6	B
7	F

It has also been found by observing the above computation that though PageRank value of node G is more, but it hold one query term only. This shows the scantiness in getting the true value by following a single approach of structural links which is not enough for analyzing the ranking of the nodes.

The proposed algorithm is now implemented on the given dataset having 8910 URLs and tested on 100 queries using Spyder (Python 3.7). The cumulative weight of the URLs is represented in Table 3.12.

Table 3.12 Cumulative weight of the URLs

Index	ind	Supply	Chain	anagement	i(w)	o(w)	cw(n)	cw(x)
0	1	94	95	61	7	10	0.411765	1.132
1	2	34	37	16	2	5	0.285714	3.25287
2	3	21	21	26	1	4	0.2	4.16176
3	4	317	308	241	26	29	0.472727	0.32679
4	5	64	65	16	5	8	0.384615	1.95172
5	6	94	95	61	7	10	0.411765	1.132
6	7	317	308	241	26	29	0.472727	0.32679
7	8	317	308	241	26	29	0.472727	0.32679
8	9	317	308	241	26	29	0.472727	0.32679
9	10	317	308	241	26	29	0.472727	0.32679
10	11	25	24	13	2	5	0.285714	4.56452
11	12	14	16	13	1	4	0.2	6.5814
12	13	57	60	29	4	7	0.363636	1.93836
13	14	29	28	14	2	5	0.285714	3.98592
14	15	45	45	22	3	6	0.333333	2.52679
15	16	54	52	30	4	7	0.363636	2.08088
16	17	45	45	8	3	6	0.333333	2.88776
17	18	28	26	72	2	5	0.285714	2.24603
18	19	51	49	16	4	7	0.363636	2.43966
19	20	29	28	8	2	5	0.285714	4.35385
20	22	17	17	9	1	4	0.2	6.5814

After applying the rules of the proposed algorithm, clustering is done and the cumulative weight of web pages concerning the Query is evaluated for Clusters and shown in Table 3.13.

Table 3.13 Cumulative weight of nodes in Cluster

Index	Type	Size	Value
0	float	1	6.80776
1	float	1	8.79181
2	float	1	9.60855
3	float	1	6.06823
4	float	1	7.59765
5	float	1	6.80776
6	float	1	6.06823
7	float	1	6.06823
8	float	1	6.06823
9	float	1	6.06823
10	float	1	10.10083
11	float	1	12.02214
12	float	1	7.56199
13	float	1	9.52339
14	float	1	8.11739
15	float	1	7.70432
16	float	1	8.47776
17	float	1	7.78698
18	float	1	8.06258
19	float	1	9.89058

The processing of the scRank algorithm in different iteration is shown in Table 3.14.

Table 3.14 Processing steps of Different Iteration in scRank

	0	1	2	3	4	5	6	7	8	9	10	11	12
0	1.54401	1.54301	1.54251	1.54466	1.54376	1.54401	1.54466	1.54466	1.54466	1.54466	1.54301	1.54251	1.54358
1	1.84442	1.84287	1.8421	1.84543	1.84403	1.84442	1.84543	1.84543	1.84543	1.84543	1.84287	1.8421	1.84376
2	2.01031	2.00847	2.00755	2.01152	2.00985	2.01031	2.01152	2.01152	2.01152	2.01152	2.00847	2.00755	2.00952
3	2.10193	2.09992	2.09892	2.10324	2.10143	2.10193	2.10324	2.10324	2.10324	2.10324	2.09992	2.09892	2.10107
4	2.15252	2.15042	2.14938	2.1539	2.152	2.15252	2.1539	2.1539	2.1539	2.1539	2.15042	2.14938	2.15162
5	2.18046	2.17831	2.17724	2.18187	2.17992	2.18046	2.18187	2.18187	2.18187	2.18187	2.17831	2.17724	2.17954
6	2.19589	2.19372	2.19263	2.19732	2.19535	2.19589	2.19732	2.19732	2.19732	2.19732	2.19372	2.19263	2.19496
7	2.20441	2.20222	2.20113	2.20585	2.20386	2.20441	2.20585	2.20585	2.20585	2.20585	2.20222	2.20113	2.20347
8	2.20912	2.20692	2.20582	2.21056	2.20857	2.20912	2.21056	2.21056	2.21056	2.21056	2.20692	2.20582	2.20818
9	2.21172	2.20951	2.20841	2.21316	2.21117	2.21172	2.21316	2.21316	2.21316	2.21316	2.20951	2.20841	2.21077
10	2.21315	2.21095	2.20984	2.2146	2.2126	2.21315	2.2146	2.2146	2.2146	2.2146	2.21095	2.20984	2.21221
11	2.21395	2.21174	2.21063	2.2154	2.21339	2.21395	2.2154	2.2154	2.2154	2.2154	2.21174	2.21063	2.213
12	2.21438	2.21217	2.21107	2.21583	2.21383	2.21438	2.21583	2.21583	2.21583	2.21583	2.21217	2.21107	2.21344
13	2.21462	2.21241	2.21131	2.21607	2.21407	2.21462	2.21607	2.21607	2.21607	2.21607	2.21241	2.21131	2.21368
14	2.21476	2.21255	2.21144	2.21621	2.2142	2.21476	2.21621	2.21621	2.21621	2.21621	2.21255	2.21144	2.21381
15	2.21483	2.21262	2.21152	2.21628	2.21428	2.21483	2.21628	2.21628	2.21628	2.21628	2.21262	2.21152	2.21388
16	2.21487	2.21266	2.21156	2.21632	2.21432	2.21487	2.21632	2.21632	2.21632	2.21632	2.21266	2.21156	2.21392
17	2.21489	2.21268	2.21158	2.21635	2.21434	2.21489	2.21635	2.21635	2.21635	2.21635	2.21268	2.21158	2.21395
18	2.21491	2.2127	2.21159	2.21636	2.21435	2.21491	2.21636	2.21636	2.21636	2.21636	2.2127	2.21159	2.21396

After performing the steps of the algorithm the result of scRank and PageRank is illustrated in Table 3.15 and Table 3.16.

Table 3.15 Ranking result in scRank

Index	0	1
0	27	5.84686
1	41	5.94063
2	235	6.04904
3	203	6.0659
4	4	6.06823
5	7	6.06823
6	8	6.06823
7	9	6.06823
8	10	6.06823
9	201	6.10429
10	72	6.15956
11	234	6.17104
12	217	6.18337
13	205	6.22277
14	215	6.2318
15	208	6.26634
16	218	6.26716
17	137	6.27523
18	231	6.30192
19	232	6.32625
20	140	6.3393

Table 3.16 Ranking result in PageRank

Index	0	1
0	20	2.20977
1	22	2.20977
2	24	2.20977
3	27	2.20977
4	28	2.20977
5	30	2.20977
6	31	2.20977
7	36	2.20977
8	41	2.20977
9	43	2.20977
10	45	2.20977
11	46	2.20977
12	47	2.20977
13	48	2.20977
14	54	2.20977
15	56	2.20977
16	57	2.20977
17	60	2.20977
18	66	2.20977
19	69	2.20977
20	72	2.20977

The comparative result of the scRank and PageRank is depicted in Table 3.17.

Table 3.17 Ranking result in scRank and PageRank

Index	scrank	val_sc	pgrank	val_pg
0	27	5.84686	20	2.20977
1	41	5.94063	22	2.20977
2	235	6.04904	24	2.20977
3	203	6.0659	27	2.20977
4	4	6.06823	28	2.20977
5	7	6.06823	30	2.20977
6	8	6.06823	31	2.20977
7	9	6.06823	36	2.20977
8	10	6.06823	41	2.20977
9	201	6.10429	43	2.20977
10	72	6.15956	45	2.20977
11	234	6.17104	46	2.20977
12	217	6.18337	47	2.20977
13	205	6.22277	48	2.20977
14	215	6.2318	54	2.20977
15	208	6.26634	56	2.20977
16	218	6.26716	57	2.20977
17	137	6.27523	60	2.20977
18	231	6.30192	66	2.20977
19	232	6.32625	69	2.20977
20	140	6.3393	72	2.20977

3.4 Result Analysis

The final representation shows an improvement in scRank to the basic PageRank model. It has been found that the execution time and iteration cycle of scRank is comparatively less concerning the PageRank algorithm. The execution time of the running process is graphically represented in Figure 3.5(a) and 3.5(b).

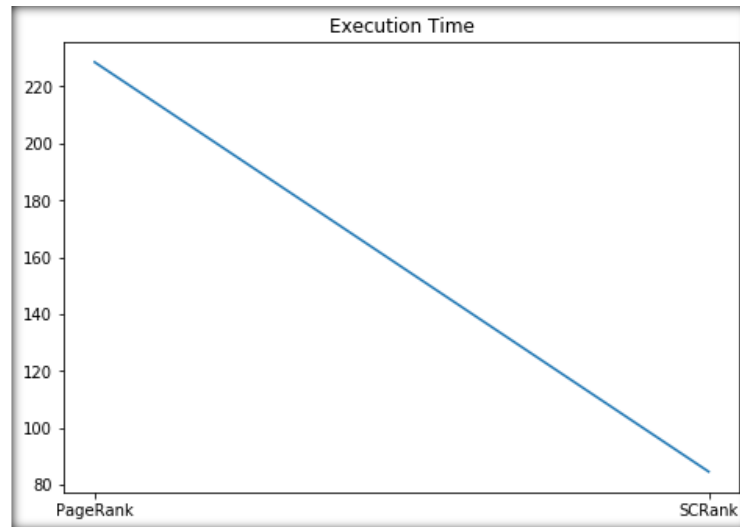


Figure 3.5(a) Execution time of scRank and PageRank



Figure 3.5(b) Bar Graph of Execution time for scRank and PageRank

The number of iterations to perform the working of scRank and PageRank is graphically presented in Figure 3.6 that shows that scRank takes 19 iterations to complete the processing while PageRank require 25 iterations.

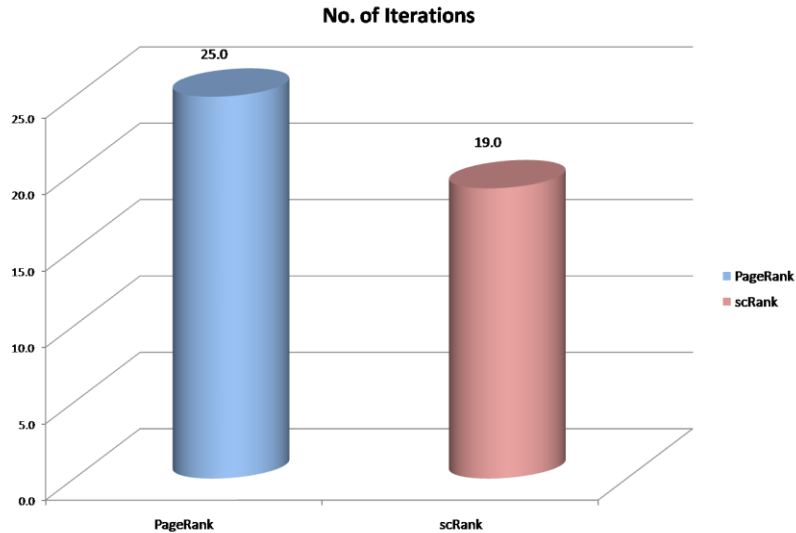


Figure 3.6 Number of iterations for scRank and PageRank

Figure 3.7 (a) and Figure3.7 (b) depict the results of different iterations of web pages in the scRank and PageRank algorithm.

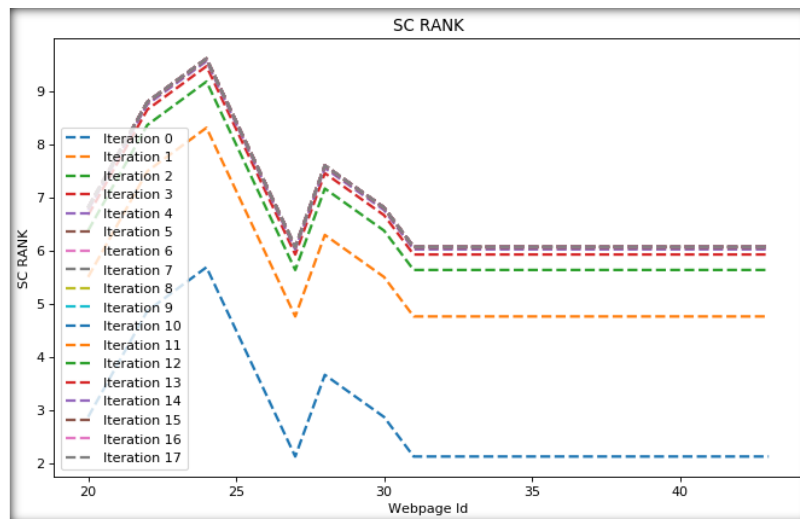


Figure 3.7(a) Result of iterations in scRank

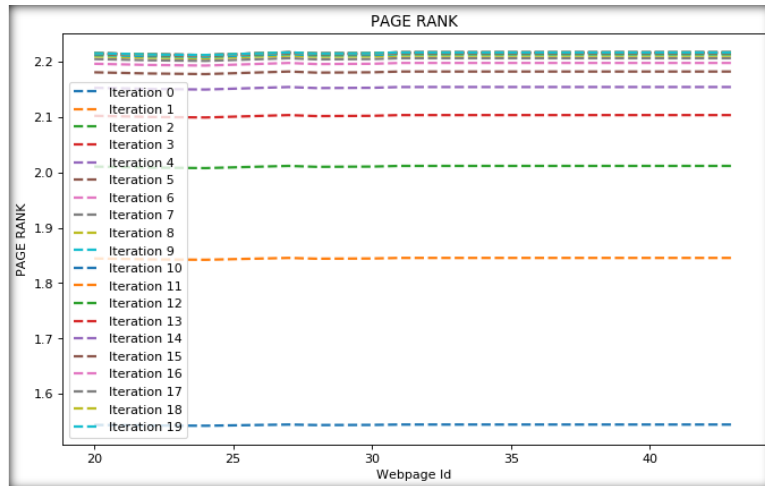


Figure 3.7(b) Result of iterations in PageRank

Finally to access the ranking quality of scRank with PageRank different evaluating measures like Precision, Recall, Average Precision, and Mean Average Precision is performed and the results are presented in Figure 3.8, Figure 3.9, Figure 3.10 and Figure 3.11. The Figure 3.8 shows that the precision value for scRank is approximately 48 while PageRank has a value of 40, which shows high performance of scRank algorithm.

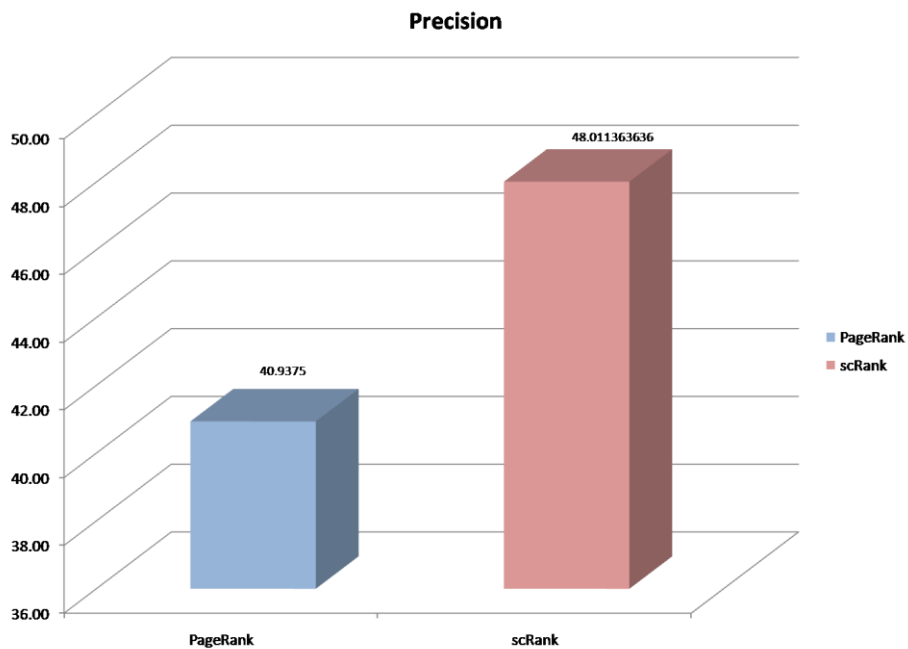


Figure 3.8 Precision values for scRank and PageRank

The Figure 3.9 shows that the recall value for scRank is approximately 52 while PageRank has a value of 46.

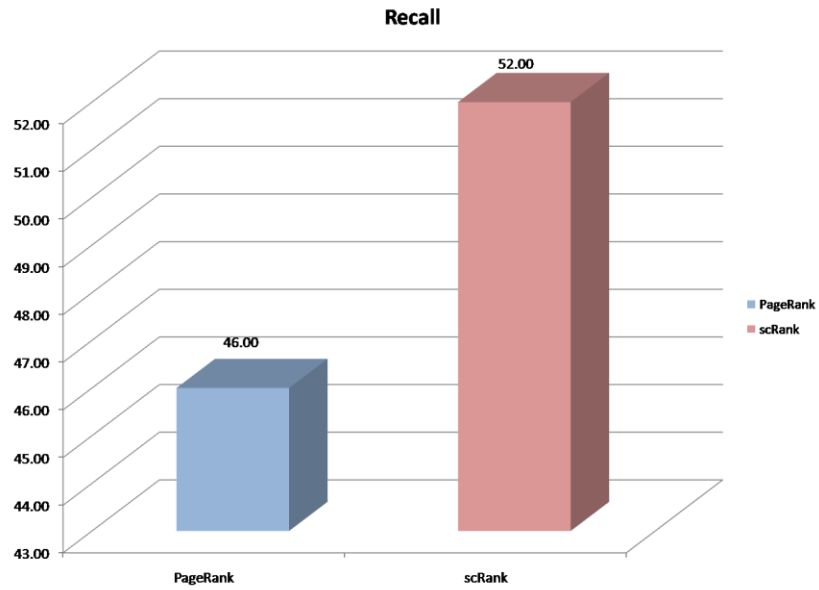


Figure 3.9 Recall value for scRank and PageRank

The Figure 3.10 shows that the average precision value for scRank is approximately 67 while PageRank has a value of 45.

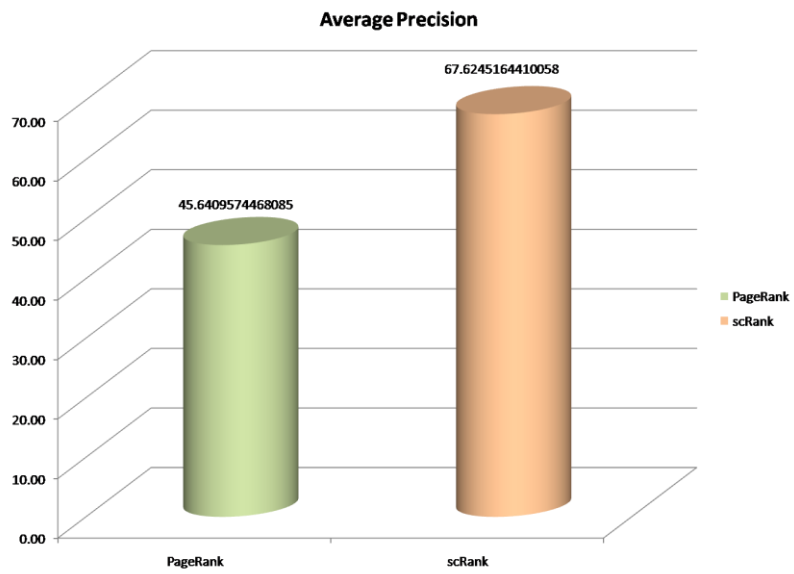


Figure 3.10 Average Precision for scRank and PageRank

The Figure 3.11 shows that the mean average precision value for scRank is approximately 57 while PageRank has a value of 43.

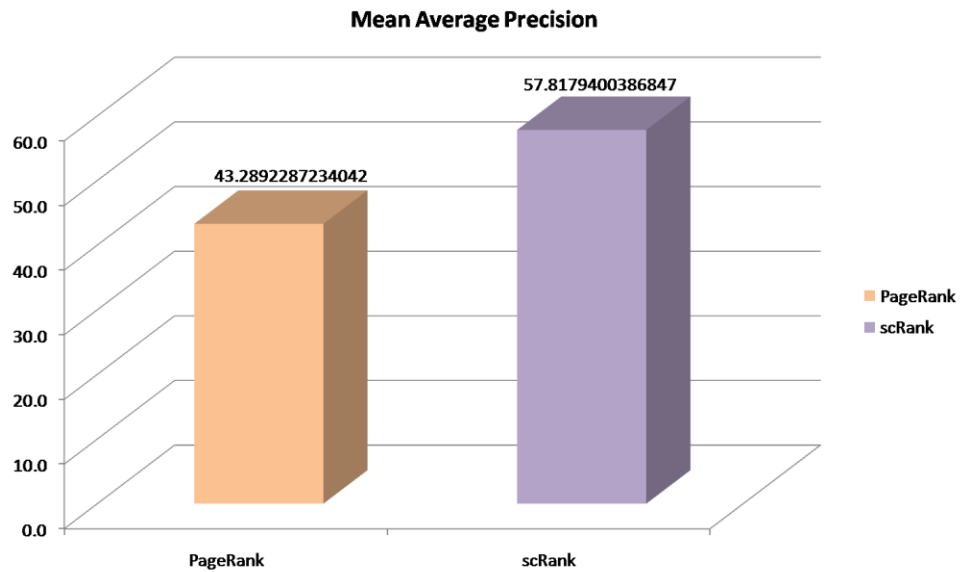


Figure 3.11 Mean Average Precision for scRank and PageRank

To assess the ranking quality of the algorithm, Cumulative Gain, Discounted Cumulative Gain, Idealized Discounted Cumulative Gain, and Normalized Discounted Cumulative Gain is used which shows the performance improvement and is depicted in Figure 3.12, Figure 3.13, Figure 3.14 and Figure 3.15.

The Figure 3.12 shows that the cumulative gain value for scRank is approximately 331 while PageRank has a value of 262.

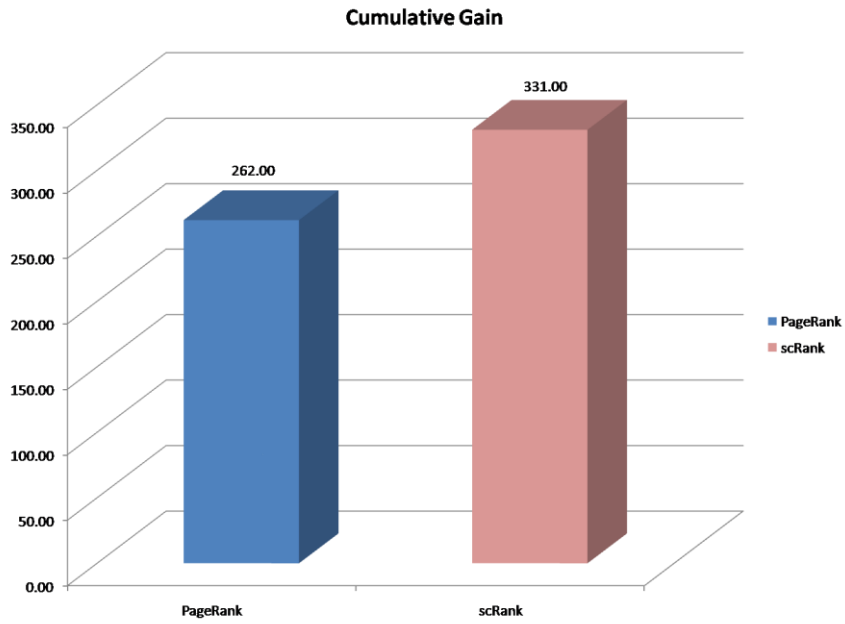


Figure 3.12 Cumulative Gain for scRank and PageRank

The Figure 3.13 shows that the discounted cumulative gain value for scRank is approximately 59 while PageRank has a value of 46.

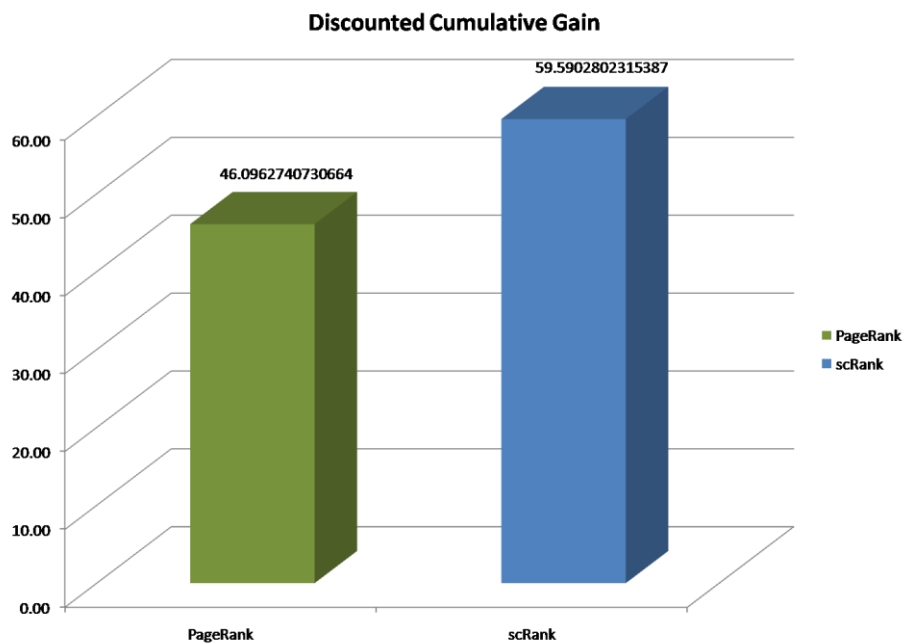


Figure 3.13 Discounted Cumulative Gain for scRank and PageRank

The Figure 3.14 shows that the idealized discounted cumulative gain value for scRank is approximately 65 while PageRank has a value of 55.

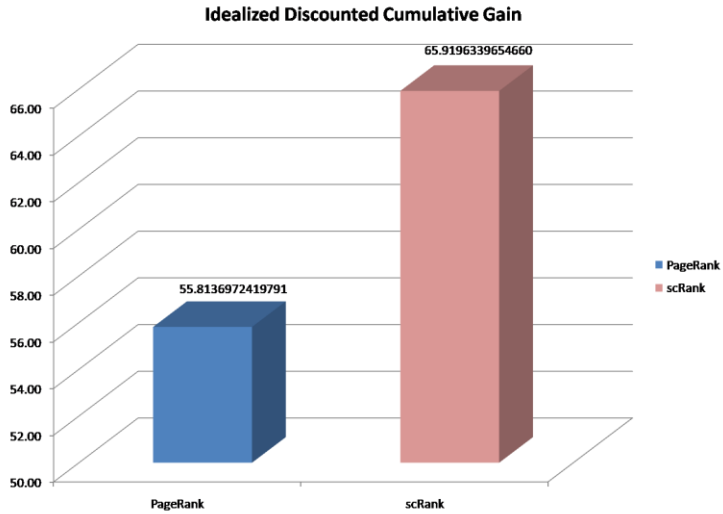


Figure 3.14 Idealized Discounted Cumulative Gain for scRank and PageRank

The Figure 3.15 shows that the normalized discounted cumulative gain value for scRank is approximately 0.90 while PageRank has a value of 0.82 that show an improvement in the ranking quality of the proposed algorithm.

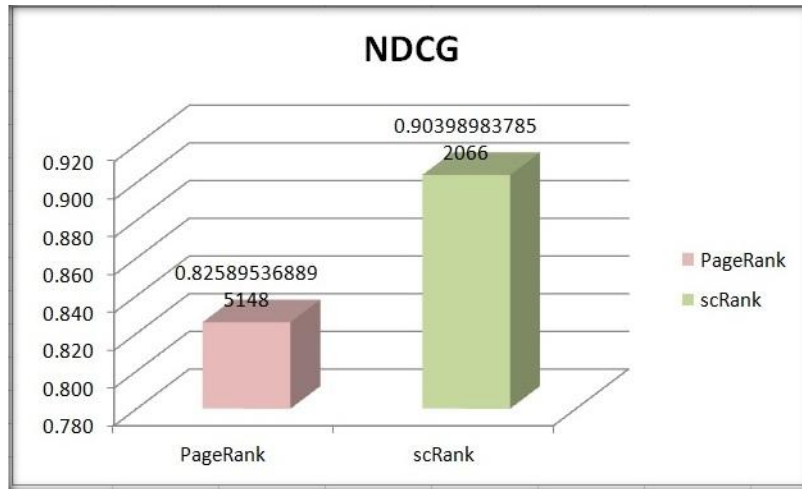


Figure 3.15 Normalized Discounted Cumulative Gain for scRank and PageRank

3.5 Summary

The proposed approach provides high precision value for ranking web pages that encapsulate both the structural links and content for the web pages, while PageRank only considers the link approach. By using both the approach high-quality, user-relevant information can be retrieved in a lesser time duration. The proposed work successfully provides more accurate web pages by using a clustering approach to group nodes at the initial phase of processing on the given data set. It is seen that however the hyperlink count is high for a node, yet at the same time due to the absence of content it is not considered for ranking and is eliminated at the preliminary phase.

The projected work helps in reducing the number of iterations and overall minimizes the computational complexity of the ranking process. It is well-known that information retrieval process is a difficult task and much of the time is spent in retrieving relevant web pages, so by following this procedure only relevant nodes are considered and we exclude irrelevant nodes that lack content thus improving the efficiency of the process and also reduce the computational complexity of the process while it is not found in PageRank approach and it evaluates all the pages irrespective of lack of content thus increasing the processing time. scRank also provides the results in less number of iterations compared with PageRank. The evaluation measure of both the approach depicts a rise in the precision value and an improved normalized value is obtained while using the proposed approach.

CHAPTER 4

DESIGN AND IMPLEMENTATION OF GRANK ALGORITHM USING GENETIC ALGORITHM

4.1 Introduction

In recent years the expansion of online information is taking place at a fast rate. World Wide Web repository comprises of diverse information that help in fulfilling the requirements of the user. In order to retrieve the derived information the structure of the internet must be examined properly. In information retrieval system, different ranking algorithms are recommended that are considered as the backbone. Nature-inspired evolutionary measures can be utilized that help to optimize the retrieved information. For handling the complex environment of the web effectively, Genetic Algorithm is used here for efficient search. This chapter presents a gRank algorithm that ranks the results of the query using genetic algorithm. The web pages are clustered together by applying certain conditions which is followed by selection. It is followed by crossover and finally, the mutation is applied. The chapter presents the working process of gRank and a relationship is set between gRank and PageRank algorithm. By following a limited set of iterations the proposed method optimize the retrieved information in minimal time duration.

4.2 Related work

The work by various researchers have pointed out that different approaches are used for ranking query result by incorporating a Genetic Algorithm. Rathod et al., (2017) [99], proposed a personalized system that is used for tailoring the search results which is

based on user interest and preferences. Genetic Algorithm is used for finding best feature selection based on the fitness relevancy and provides effective and efficient search result. Yates et al., (2019) [100], proposed CombGenRank, an effective ranking algorithm that provides efficient web pages. It introduces a novel selection criterion that is effective for supporting web document ranking.

Chen, (2018) [101], proposed a page clipping search engine that works on page-discussion topics where it clips the relevant page's topic wise based on user interest. It uses an adjustedN-gram language model that uses a hash method for producing significant topics and also employs a genetic algorithm for clipping results. The method helps in finding multiple clustering relationships which is a cost-effective solution. Yan et al., (2011) [102], introduced a genetic PageRank algorithm that solves the web search problem by taking advantage of the genetic algorithm. Mueller, (2017) [103], proposed a method that selects items repeatedly using recommendations and produces a new population using a genetic algorithm. It uses a weighted hybridization method to create a single one by combining multiple recommendations. The method provides significant improvement for long lists of recommended items.

Subramanian and Sivaprakasam, (2014) [104], method work for removing duplicate web documents by employing the Genetic Algorithm that helps to improve the relevancy of the documents that are retrieved. Thada and Jaglan, (2014) [105], proposed the work for finding relevant documents using Rogers-Tanimoto, MountFord, and Baroni-Urbani/Buser similarity coefficients measures. Genetic Algorithm is being used for depicting resemblance between query and document and it mount high when the Probability of mutation is lowered while the Probability of crossover is increased. Singh and Solanki, (2019) [106], presents a survey about various metrics that are used in recommender system performance analysis and also throws light on various nature-inspired algorithms for recommender systems. S.V. Semenikhin, (2016) [107], present a ranking function article that learns to rank process in information retrieval systems. It uses a hybrid method that works on a modified genetic algorithm and the Nelder-Mead method that helps in improving ranking graded-metrics, like NDCG, and overall reduces the time for learning to rank. Radwan et al., (2008) [108], investigated the use of a Genetic Algorithm by introducing a new fitness function that approximates the

information retrieval in a fast manner. Yeh et al., (2007) [109], proposed a machine learning approach for generating effective ranking functions for Information Retrieval. The RankGP uses genetic programming for learning a ranking function by employing content, structure, and query independent features.

4.3 Genetic Algorithm

The nature-inspired algorithms are used for handling real-world optimization problems that are quite large and complex. Optimization techniques make use of available resources and help in finding the best solution without disturbing the desired conditions. The genetic algorithm is regarded as a stochastic global optimization technique that follows Darwin's theory of evolution where principles like selection and crossover are adopted to find solutions for a search space. GA is a computational effective and adaptive method and is usually employed where search space is of high-dimensionality and mathematically computed. Complex problems can be efficiently solved using Genetic Algorithm and can be used for optimization purpose [111]. Clustering is an unsupervised problem that handles a large volume of data that act as data processing technique and GA can be employed in this area. [110], [112].

The search space in the Genetic Algorithm is composed of chromosomes that comprise a fitness value and these chromosomes along with their fitness value constitute a population [108]. It follows the natural process of evolution where the population is usually encoded as chromosomes. The chromosomes are judged based on the fitness value that shows its degree of goodness. To generate better solutions, the selection operator is used followed by crossover and mutation. Genetic Algorithm works with coding of the parameter set by considering multiple points in a blind search manner using stochastic operators. The new population is generated by the mating process by choosing the best individuals within the current population. This helps in carrying good characteristics to the next generations and an optimal solution is obtained [83].

4.4 Proposed Methodology

The given approach gRank Algorithm uses a Genetic Algorithm for web page ranking and consider both the the structural characteristics and the content of web pages. Following the given query Q it extracts W web pages using the filtering criteria. Fitness and probability of the pages are evaluated, and by monitoring the higher fitness value pages are selected and genetic operators- single-point crossover and flip mutator is applied for getting better results. Roulette Wheel is applied for selection which is directly proportional to the fitness value.

Proposed methodology gRank clusters the relevant web pages and eliminate those pages that lack proper content and then apply GA for ranking purpose while PageRank approach consider all the web pages for computation that later increases the complexity of the process. gRank reduces the computational complexity of the process by ranking the nodes based on the fitness value.

4.4.1 Architecture

The outline of the given gRank approach is represented in Figure 4.1

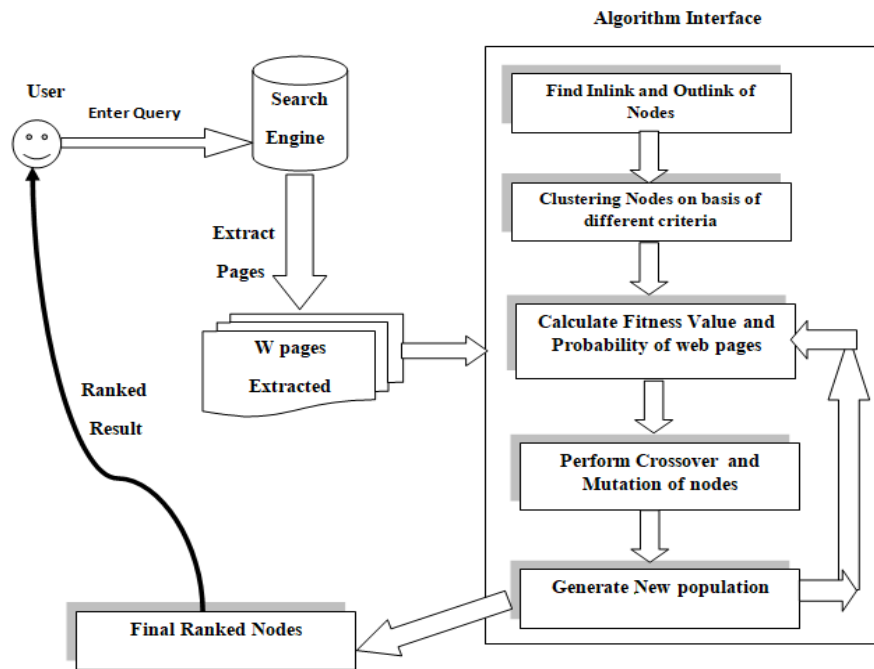


Figure 4.1 Proposed Architectural Model for gRank

4.4.2 Algorithm

The proposed algorithm provides ranked web pages by accepting user query and follows certain steps of the algorithm and is illustrated as-

Algorithm: gRank

Input: User Query

Output: Relevant Ranked Web Pages (RW)

Steps:

1. User entered query Q comprises of 'n' terms that extract 'W' pages by removing stopwords.
2. Count number of terms 'n' of query 'Q'.
3. Calculate Inlink (I) and Outlink (O) weight of the nodes.
4. Cluster Web pages using following criteria-
 - Case I: If all term 'n' of 'Q' appear in W_i where $i=1, 2, 3, \dots, X$ group them in CL_i , where $i=1, 2, 3$.
 - Case II: If ' $n-1 \leq W \leq n/2$ ' group them in CL_i , where $i=1, 2, 3, \dots, X$.
 - Case III: If one term 'n' match group them in cluster CL_i , where $1, 2, 3, \dots, X$, exclude.
5. Calculate Fitness value of the web pages $F(X)$ using the formulas-

$$F(X) = (W(X) / Q(n)) + (I(W) / O(W))$$

where

$W(X)$ - Term weight of page X corresponding to Q

$Q(n)$ - Term weight n of query Q

$I(W)$ - Inlink weight of web page X

$O(W)$ - Outlink weight of web page X

6. Calculate probability of the web pages $P(X)$ using Formula-

$$P(X) = F(X) / \sum_{i=1}^X F(X)$$

where

F(X) - Fitness of the web page

7. Calculate Expected Count of the web page EC(X) and Actual Count of the web page AC(X) which is approximate of EC(X) using Formula-

$$EC(X) = F(X) / \overline{F(X)}$$

where

F(X) - Fitness of the web page

$\overline{F(X)}$ – Average of fitness of the web pages

8. Select Mating pool and mate. Apply one- point crossover and Flip mutation to produce new population.
9. Fitness value is again evaluated for the New generation and the steps 5- 8 is repeated till termination condition is reached, which is repetition of same strings of chromosomes.
10. Finally web pages are ranked on the basis of Fitness values.

4.4.3 Dataset Used

A dataset consisting of 8910 URLs are collected for testing the algorithm. The collected URLs are then tested on a set of different queries that generate the proper ranking result. A total of 100 queries are taken and are tested on the collected URLs. A small part of the dataset and queries employed for evaluating the ranking result is shown for illustration purpose in Fig. 7.1 and Fig 7.2. The proposed work is experimentally performed on Spyder (Python 3.7).

Index	0
0	https://searcherp.techtarget.com/definition...
1	https://www.investopedia.com/terms/s/scm.asp
2	https://www.knowledgekafe.com/course/supply-chain-management-125022
3	https://en.wikipedia.org/wiki/Supply_chain_management
4	https://www.cio.com/article/2439493/what-is...
5	https://searcherp.techtarget.com/definition/supply-chain-management-SCM
6	https://en.wikipedia.org/wiki/Supply_chain_management#Importance
7	https://en.wikipedia.org/wiki/Supply_chain_...
8	https://en.wikipedia.org/wiki/Supply_chain_...
9	https://en.wikipedia.org/wiki/Supply_chain_management#Supply_chain
10	https://scm.ncsu.edu/scm-articles/article/what-is-supply-chain-management-scm
11	https://www.youtube.com/watch?v=lZPOSrclZEo
12	https://www.ibm.com/topics/supply-chain-management
13	https://www.oracle.com/applications/supply-chain-management/
14	https://www.coursera.org/specializations/supply-chain-management
15	https://www.tradegecko.com/supply-chain-management
16	https://cscmp.org/CSCMP/Develop/Starting_Yo...
17	https://www.managementstudyguide.com/supply-chain-management-definition.htm
18	https://www.emerald.com/insight/publication/issn/1359-8546
19	https://hbr.org/2018/06/the-death-of-supply-chain-management
20	https://www.edx.org/micromasters/mitx-suoolv-chain-management

Figure 4.2 Dataset of web pages

Index	0
0	Google Driver Less Car
1	Micro Machining
2	Solar Powered Aircraft
3	High Speed Machining
4	Quad Copter
5	Non-Pneumatic Tyres
6	Electrochemical Machining
7	Dual Fuel Engine
8	Methanol Fueled Marine Diesel Engine
9	Intelligent Variable Valve Timing
10	Solid Waste Management
11	Hemi Engine
12	Scram Jet Engine for Hypersonic Flight
13	Sidewinder Missile
14	Laser Cutting System
15	Airless Tyres
16	Pneumatic Tyres
17	Chassis Frame
18	Pulse Detonation Engine
19	Solar Collector
20	Semi Active Suspension System

Figure 4.3 Queries tested on web pages

4.4.4 Experimental Analysis

For implementing the gRank approach a directed graph having nodes A, B, C, D, E, F, and G is taken as shown in Figure 2 that are interlinked among each other.

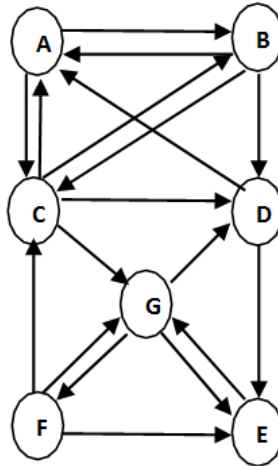


Figure 4.4 Hyperlink Structure of web pages

Table 4.1 displays the occurrence of the terms found on the web pages based on the query.

Table 4.1 Occurrence of terms on Web Pages

Terms\ Nodes	A	B	C	D	E	F	G
Information	20	8	8	10	20	0	25
Retrieval	10	12	13	15	10	0	0
Evolutionary	12	0	6	8	5	0	0
Computation	8	0	2	6	2	6	0

The Inlink weight (IW) and Outlink (OW) weight of the nodes is evaluated as shown in Table 4.2.

Table 4.2 Weight of the nodes

Node	I(W)	O(W)
A	3	2
B	2	3
C	3	4
D	3	2
E	3	1
F	1	3
G	3	3

Based on the rules of the algorithm, the nodes are now clustered as illustrated in Step 4. By close observation, it is found that the node A, D, C, and E contain all query words and therefore Case I is applied and are clustered in one group. On node B, Case II is applied which is settled in the second cluster. Following Case III, the two nodes F and G are left out. After clustering the nodes, the Fitness value of the web pages $F(X)$ is calculated using the formula shown in 4.1.

$$F(X) = (W(X) / Q(n)) + (I(W) / O(W)) \quad (4.1)$$

After evaluating web page Fitness value, the probability $P(X)$ for web pages is calculated using the formula shown in 4.2.

$$P(X) = F(X) / \sum_{i=1}^X F(X) \quad (4.2)$$

The probability percentage is evaluated and finally, the Expected Count of the web page $EC(X)$ and Actual Count of the web page $AC(X)$ is calculated using formula shown in 4.3, the $AC(X)$ is approximate of $EC(X)$.

$$(4.3)$$

$$EC(X) = F(X) / \overline{F(X)}$$

The values of Fitness, probability, probability percentage, Expected Count, and Actual Count are shown in Table 4.3.

Table 4. 3 Value of Fitness, Probability, Probability Percentage, Expected Count and Actual Count for the nodes

Nodes	I(W)	O(W)	Q(n)	W(X)	F(X)	P(X)	P(X)%	EC(X)	AC(X)
A	3	2	4	50	14	0.273615635	27.3615635	1.368078176	1
B	2	3	4	20	5.666666667	0.110749186	11.0749186	0.553745928	1
C	3	4	4	29	8	0.156351792	15.6351792	0.781758958	1
D	3	2	4	39	11.25	0.219869707	21.9869707	1.099348534	1
E	3	1	4	37	12.25	0.239413681	23.9413681	1.197068404	1
Sum of F(X)					51.16666667				
Avg					10.23333333				

After finding the solution space the genetic operators are applied like a crossover (CO), mutation (MU), and again steps 5- 8 are repeated to find the next generation of population as shown in Table 4.4, Table 4.5 and Table 4.6. The MP represents Mating Pool, M represents Mate and NP is the New Population.

Table 4.4 1st Iteration after applying Genetic Operators

Nodes	MP	M	CO	MU	NP	Fi	Pi	P%	EC	AC
A	00110010	AB	00110100	00111100	60	16.5	0.292467	29.246677	1.462334	1
B	00010100	BA	00010010	00011010	26	7.166666667	0.127031	12.703102	0.635155	1
C	00011101	CD	00010111	00011111	31	8.5	0.150665	15.06647	0.753323	1
D	00100111	DC	00101101	00100101	37	10.75	0.190547	19.054653	0.952733	1
E	00100101	EA	00100010	00101010	42	13.5	0.239291	23.929099	1.196455	1
Sum of F(X)						56.41666667				
Avg						11.28333333				

Table 4.5 2nd Iteration after applying Genetic Operators

Nodes	MP	M	CO	MU	NP	Fi	Pi	P%	EC	AC
A	00111100	AB	00111010	00110010	50	14	0.2749591	27.49591	1.374795	1
B	00011010	BA	00011100	00010100	20	5.666666667	0.111293	11.1293	0.556465	1
C	00011111	CD	00010101	00011101	29	8	0.1571195	15.71195	0.785597	1
D	00100101	DC	00101111	00100111	39	11.25	0.2209493	22.09493	1.104746	1
E	00101010	EA	00101100	00100100	37	12.25	0.2356792	23.56792	1.178396	1
Sum of F(X)						50.91666667				
Avg						10.18333333				

Table 4.6 3rd Iteration after applying Genetic Operators

Nodes	MP	M	CO	MU	NP	Fi	Pi	P%	EC	AC
A	00110010	AB	00110100	00111100	60	16.5	0.292467	29.246677	1.462334	1
B	00010100	BA	00010010	00011010	26	7.166666667	0.127031	12.703102	0.635155	1
C	00011101	CD	00010111	00011111	31	8.5	0.150665	15.06647	0.753323	1
D	00100111	DC	00101101	00100101	37	10.75	0.190547	19.054653	0.952733	1
E	00100100	EA	00100010	00101010	42	13.5	0.239291	23.929099	1.196455	1
Sum of F(X)						56.41666667				
Avg						11.28333333				

After observing the above result we can conclude that the values are repeating so we can terminate the process. The fitness values of all the nodes are presented in Table 4.7 and illustrated in Figure 4.5.

Table 4.7 Fitness Values in Different Iterations

Nodes	F(X)-I	F(X)-II	F(X)-III	F(X)-IV
A	14	16.5	14	16.5
B	5.666666667	7.1666667	5.666667	7.166667
C	8	8.5	8	8.5
D	11.25	10.75	11.25	10.75
E	12.25	13.5	12.25	13.5

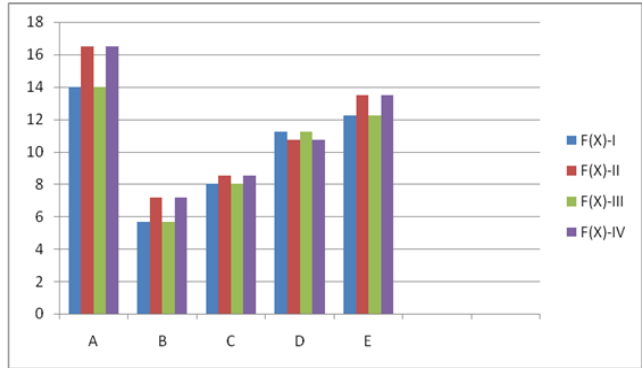


Figure 4.5 Representations of Fitness Values

The fitness value ranges in the proposed example from 5 to 17 and we are getting similar values after certain number of steps. The final ranking of the nodes is done based on Fitness value which is directly proportional to selection and is given in Table 4.8 and illustrated graphically in Figure 4.6.

Table 4.8 Ranked Nodes along with Fitness Values

Ranked Nodes	Fitness Value
A	16.5
E	13.5
D	10.75
C	8.5
B	7.166667

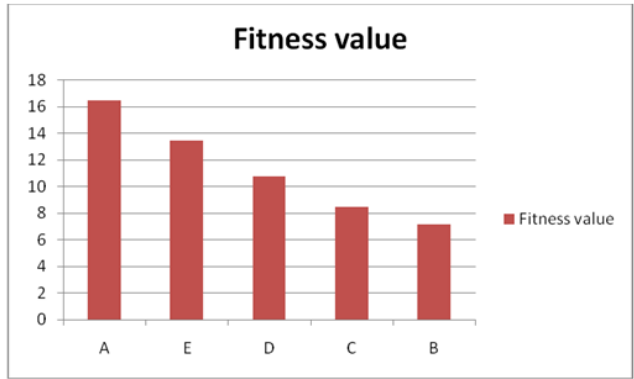


Figure 4.6 Ranking of Nodes in gRank Model

The given approach helps in achieving the result by performing less calculation to rank the nodes and help in minimizing the complexity of the process by achieving results in minimum time duration.

The original PageRank performs computation for all the nodes thus increasing the complexity and time of the process and the results are not yet achieved till 27 iterations.

Table 4.9 depicts the result of PageRank and Table 4.10 finally ranked nodes.

Table 4.9 PageRank of all Nodes

Iteration\ Node	PR(A)	PR(B)	PR(C)	PR(D)	PR(E)	PR(F)	PR(G)
0	1	1	1	1	1	1	1
1	1.070833333	0.817604167	1.120092014	0.9030074	1.100444812	0.433333333	1.446175421
2	1.003452212	0.814486743	0.930016212	0.988149391	1.102490972	0.559749703	1.44334152
3	0.99836318	0.771932797	0.951614393	0.979879115	1.133991137	0.558946764	1.474478775
4	0.987380975	0.771854973	0.946697406	0.987634427	1.145881868	0.567768986	1.486040666
5	0.989610073	0.77175748	0.95011678	0.99160929	1.153346683	0.571044855	1.494040539
6	0.991998383	0.773499129	0.952553442	0.994887179	1.157934579	0.573311486	1.499100253
7	0.994402744	0.775038772	0.954653739	0.99720331	1.160994733	0.574745072	1.502553879
8	0.996269645	0.776278519	0.956204617	0.998862661	1.163084667	0.575723599	1.504937134
9	0.997655692	0.77719715	0.957331215	1.000037597	1.16453652	0.576398855	1.506601934
10	0.998654721	0.77786114	0.958135255	1.000868279	1.165552575	0.576870548	1.507770086
11	0.99936675	0.77833461	0.958705664	1.001454617	1.166266392	0.577201524	1.508591819
12	0.999871306	0.778670258	0.959108977	1.001868246	1.166768785	0.577434349	1.509170524
13	1.000227902	0.778907516	0.95939372	1.002159943	1.16712269	0.577598315	1.509578308
14	1.000479604	0.779074997	0.959594604	1.002365623	1.1673721	0.577713854	1.50986573
15	1.000657159	0.779193146	0.959736276	1.00251064	1.167547904	0.57779529	1.510068343
16	1.000782372	0.779276467	0.959836172	1.002612883	1.167671838	0.577852697	1.51021118
17	1.000870661	0.779335217	0.959906607	1.002684966	1.167759209	0.577893168	1.510311879
18	1.00093291	0.77937664	0.959956265	1.002735787	1.167820806	0.577921699	1.510382873
19	1.000976797	0.779405845	0.959991276	1.002771616	1.167864232	0.577941814	1.510432924
20	1.001007739	0.779426435	0.96001596	1.002796877	1.167894849	0.577955995	1.510468211
21	1.001029554	0.779440952	0.960033362	1.002814686	1.167916433	0.577965993	1.510493089
22	1.001044934	0.779451186	0.960045631	1.002827241	1.167931651	0.577973042	1.510510628
23	1.001055777	0.779458402	0.960054281	1.002836093	1.16794238	0.577978011	1.510522994
24	1.001063422	0.779463489	0.960060379	1.002842334	1.167949943	0.577981515	1.510531712
25	1.001068811	0.779467075	0.960064679	1.002846734	1.167955276	0.577983985	1.510537858
26	1.001072611	0.779469604	0.96006771	1.002849836	1.167959036	0.577985726	1.510542191
27	1.00107529	0.779471386	0.960069847	1.002852023	1.167961686	0.577986954	1.510545246
28	1.001077178	0.779472643	0.960071353	1.002853565	1.167963555	0.57798782	1.5105474
29	1.00107851	0.779473529	0.960072416	1.002854652	1.167964872	0.57798843	1.510548918
30	1.001079448	0.779474154	0.960073164	1.002855418	1.167965801	0.57798886	1.510549989

Table 4.10 Ranked Nodes

Ranked Nodes
G
E
D
A
C
B
F

The observation shows that ranking of node G is high in the PageRank algorithm though it lacks content which shows the inadequacy of the approach. The gRank not only helps to remove irrelevant nodes that lack content but also consider the interconnectivity of web pages and content while calculating the ranking of the nodes. Table 4.11 depicts that the result in gRank is achieved in 4 steps while PageRank does not achieve the desired result up till 27 iterations, which shows that computational complexity is minimized in using the gRank approach.

Table 4.11 Iteration Required

	gRank	PageRank
Iteration required after which the desired result is achieved	4	27

The proposed algorithm is now implemented on the given dataset having 8910 URLs and tested on 100 queries using Spyder (Python 3.7). The Inlink weight (IW), Outlink (OW) weight, query, and document weight of the nodes based on the query is evaluated and depicted in Table 4.12.

Table 4.12 Weight of the URLs

Index	ind	Supply	Chain	lanagemer	i(w)	o(w)	w(x)	q(n)
0	1	94	95	61	7	10	250	3
1	2	34	37	16	2	5	87	3
2	3	21	21	26	1	4	68	3
3	4	317	308	241	26	29	866	3
4	5	64	65	16	5	8	145	3
5	6	94	95	61	7	10	250	3
6	7	317	308	241	26	29	866	3
7	8	317	308	241	26	29	866	3
8	9	317	308	241	26	29	866	3
9	10	317	308	241	26	29	866	3
10	11	25	24	13	2	5	62	3
11	12	14	16	13	1	4	43	3
12	13	57	60	29	4	7	146	3
13	14	29	28	14	2	5	71	3
14	15	45	45	22	3	6	112	3
15	16	54	52	30	4	7	136	3
16	17	45	45	8	3	6	98	3
17	18	28	26	72	2	5	126	3
18	19	51	49	16	4	7	116	3
19	20	29	28	8	2	5	65	3
20	22	17	17	9	1	4	43	3

The fitness value of the nodes in different iterations in the gRank algorithm is evaluated and is depicted in Table 4.13.

Table 4.13 Fitness values during Iterations

f0	f1	f2	f3	f4	f5	f6	f7	f8	f9	f10
84.0333	41.7	20.7	9.36667	3.7	2.7	1.36667	2.7	1.36667	2.7	2.7
29.4	13.7333	7.73333	4.73333	1.06667	2.4	1.06667	2.4	1.06667	1.06667	1.4
22.9167	48.9167	24.5833	12.9167	6.25	4.91667	0.916667	2.25	2.25	2.58333	0.916667
289.563	37.2299	17.5632	10.8966	4.22989	2.89655	1.56322	1.56322	1.89655	2.89655	3.22989
48.9583	26.625	25.2917	13.2917	6.625	5.29167	2.625	2.95833	1.29167	1.625	0.625
84.0333	166.7	84.0333	42.7	20.0333	9.36667	4.36667	2.7	3.03333	2.03333	0.7
289.563	145.563	73.5632	37.5632	18.8966	11.2299	4.22989	1.89655	0.896552	2.22989	1.56322
289.563	145.563	73.5632	36.2299	17.8966	10.8966	5.89655	2.22989	0.896552	2.89655	2.89655
289.563	145.563	37.5632	19.2299	9.56322	4.56322	0.896552	2.22989	1.56322	1.56322	2.22989
289.563	36.2299	17.8966	10.8966	5.89655	2.22989	0.896552	2.89655	2.89655	0.896552	1.22989
21.0667	10.7333	3.73333	1.4	0.4	1.73333	1.06667	1.06667	1.73333	2.06667	1.4
14.5833	7.58333	5.25	1.58333	0.25	2.25	2.25	0.25	0.583333	2.58333	2.25
49.2381	25.5714	11.2381	7.2381	3.90476	1.2381	1.90476	2.2381	1.57143	1.2381	2.2381
24.0667	12.4	5.73333	5.06667	2.4	0.4	0.733333	2.73333	2.4	0.733333	0.4
37.8333	20.5	9.16667	3.83333	1.83333	2.16667	1.5	1.16667	2.16667	1.83333	1.83333
45.9048	23.9048	13.2381	5.90476	3.57143	2.90476	2.57143	0.904762	0.571429	0.571429	0.571429
33.1667	17.1667	9.83333	4.83333	1.5	1.16667	2.16667	1.83333	1.83333	1.83333	0.833333
42.4	19.0667	8.73333	5.4	2.4	0.733333	0.4	0.4	0.4	2.06667	2.06667
39.2381	20.9048	9.57143	3.90476	2.2381	1.90476	1.90476	1.90476	0.904762	0.904762	2.2381
22.0667	13.4	7.73333	3.4	0.4	0.4	0.4	2.06667	2.06667	0.733333	2.4
14.5833	6.25	4.58333	1.58333	1.58333	1.58333	0.583333	0.583333	1.91667	0.916667	0.25

After performing the steps of the algorithm the comparative result of the gRank and PageRank is depicted in Table 4.14.

Table 4.14 Ranking result in gRank and PageRank

Index	pgrank	val_pg	grank	val_gg
0	20	2.20977	22	0.25
1	22	2.20977	133	0.25
2	24	2.20977	134	0.25
3	27	2.20977	87	0.333333
4	28	2.20977	125	0.333333
5	30	2.20977	183	0.333333
6	31	2.20977	193	0.333333
7	36	2.20977	198	0.333333
8	41	2.20977	225	0.333333
9	43	2.20977	14	0.4
10	45	2.20977	82	0.4
11	46	2.20977	84	0.4
12	47	2.20977	168	0.4
13	48	2.20977	16	0.571429
14	54	2.20977	142	0.583333
15	56	2.20977	187	0.583333
16	57	2.20977	5	0.625
17	60	2.20977	24	0.625
18	66	2.20977	60	0.625
19	69	2.20977	46	0.666667
20	72	2.20977	85	0.666667

4.5 Result Analysis

The final representation shows an improvement in gRank to the basic PageRank model. It has been found that the execution time and iteration cycle of gRank is comparatively less concerning the PageRank algorithm. The execution time of the running process is graphically represented in Figure 4.7(a) and 4.7(b).

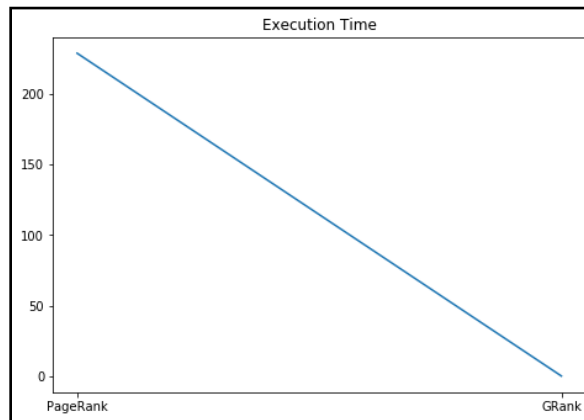


Figure 4.7(a) Execution time of gRank and PageRank

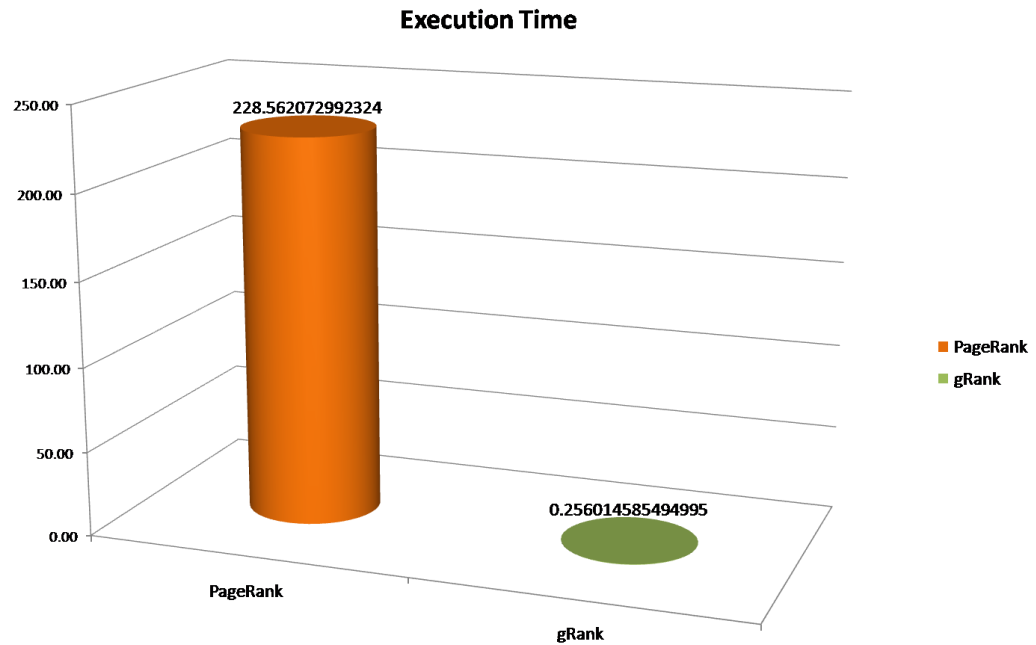


Figure 4.7(b) Bar Graph of Execution time for gRank and PageRank

The number of iterations to perform the working of gRank and PageRank is graphically illustrated in Figure 4.8 that shows that gRank takes only 10 iterations to complete the processing while PageRank require 25 iterations.

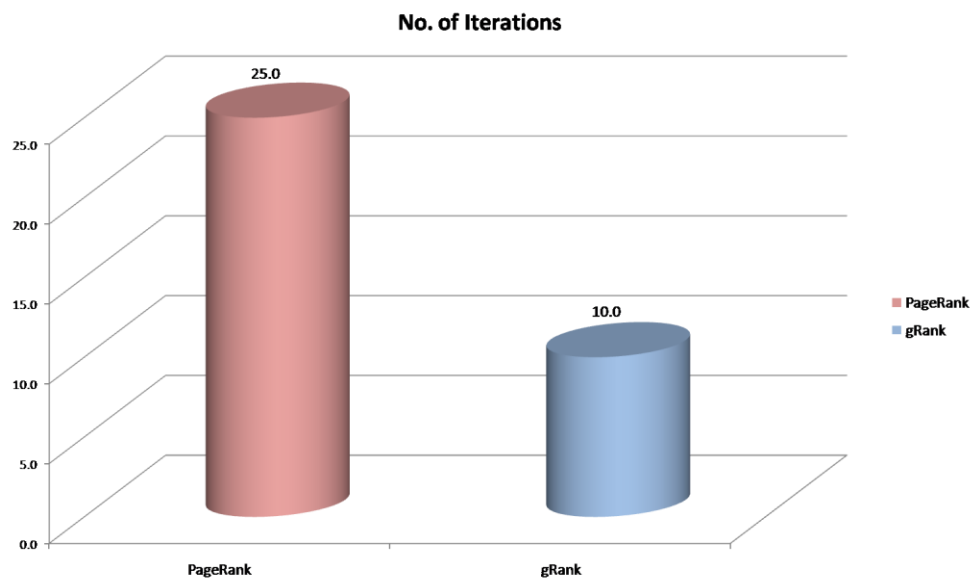


Figure 4.8 Number of iterations for gRank and PageRank

Figure 4.9(a) and Figure 4.9(b) represent the results of iterations of web pages in the gRank and PageRank algorithm.

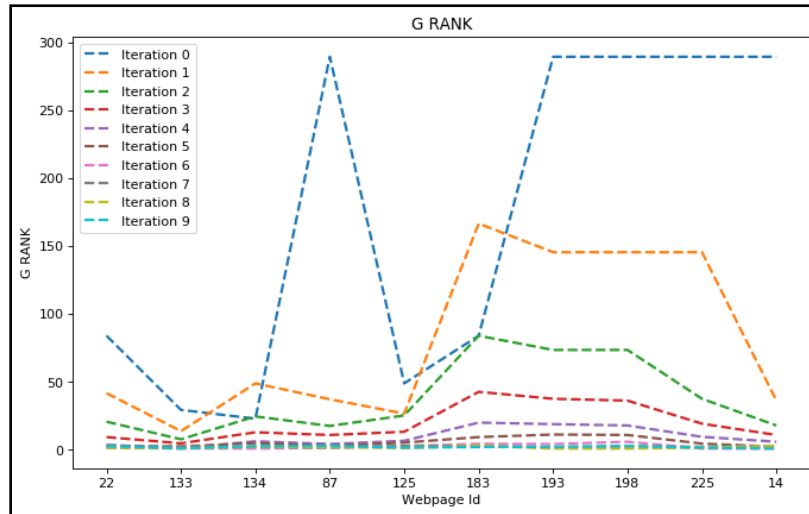


Figure 4.9(a) Result of iterations in gRank

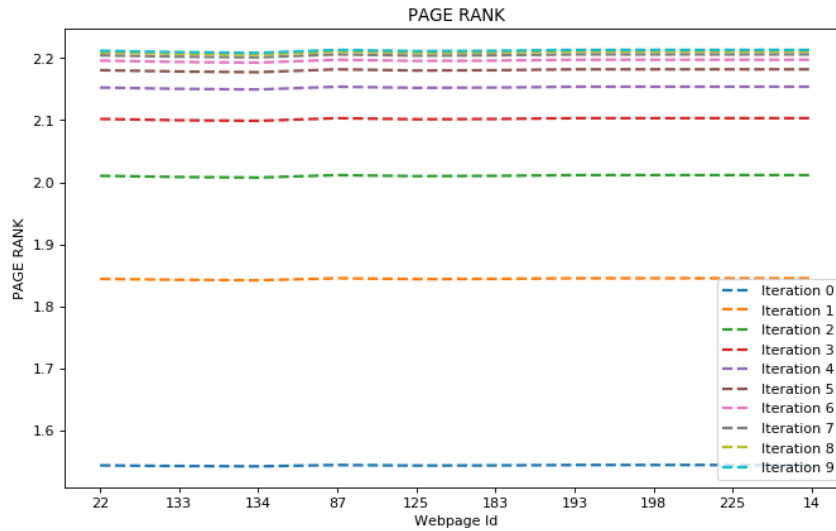


Figure 4.9(b) Result of iterations in PageRank

Finally to access the ranking quality of gRank with PageRank different evaluating measures like Precision, Recall, Average Precision, and Mean Average Precision is performed and the results are shown in Figure 4.10, Figure 4.11, Figure 4.12 and Figure

4.13. The Figure 4.10 shows that the precision value for gRank is approximately 49 while PageRank has a value of 40, which shows high performance of gRank algorithm.

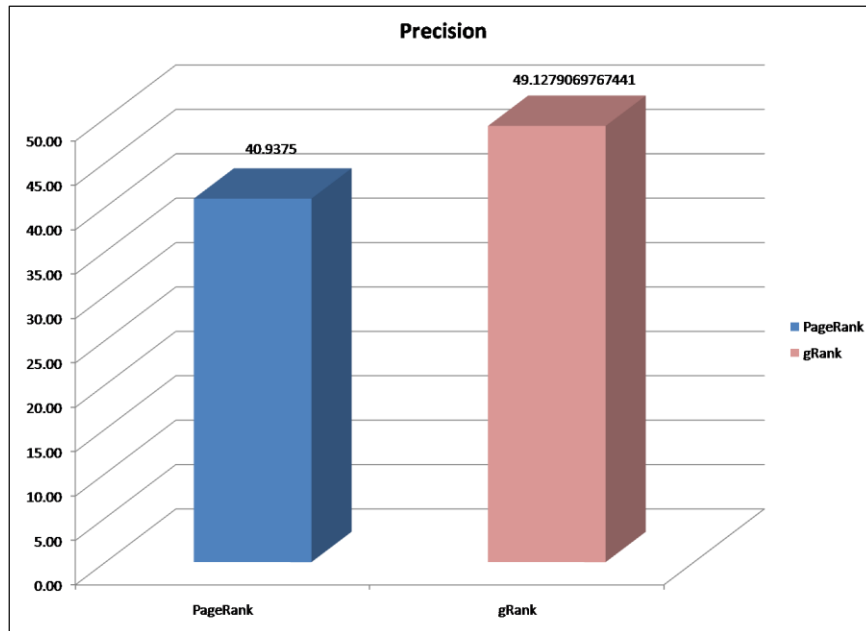


Figure 4.10 Precision values for gRank and PageRank

The Figure 4.11 shows that the recall value for gRank is approximately 60 while PageRank has a value of 46.

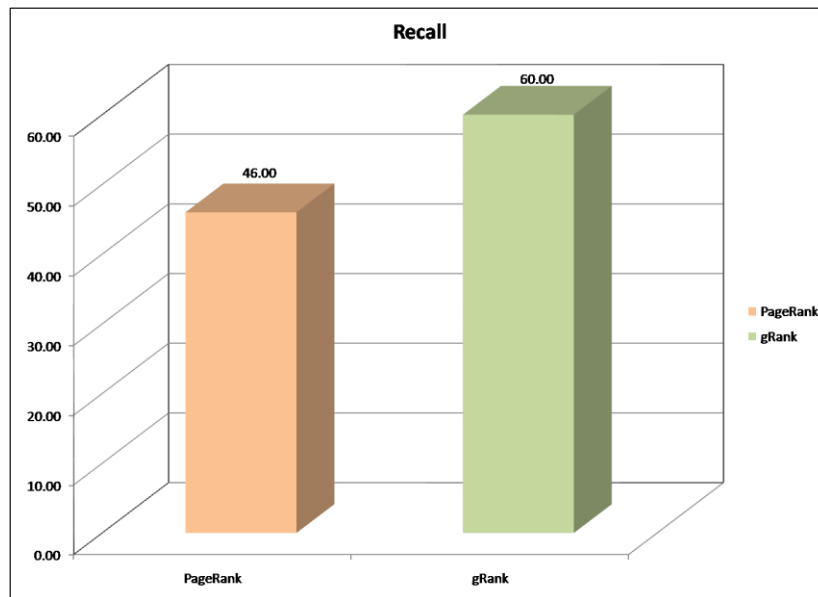


Figure 4.11 Recall value for gRank and PageRank

The Figure 4.12 shows that the average precision value for gRank is approximately 71 while PageRank has a value of 45.

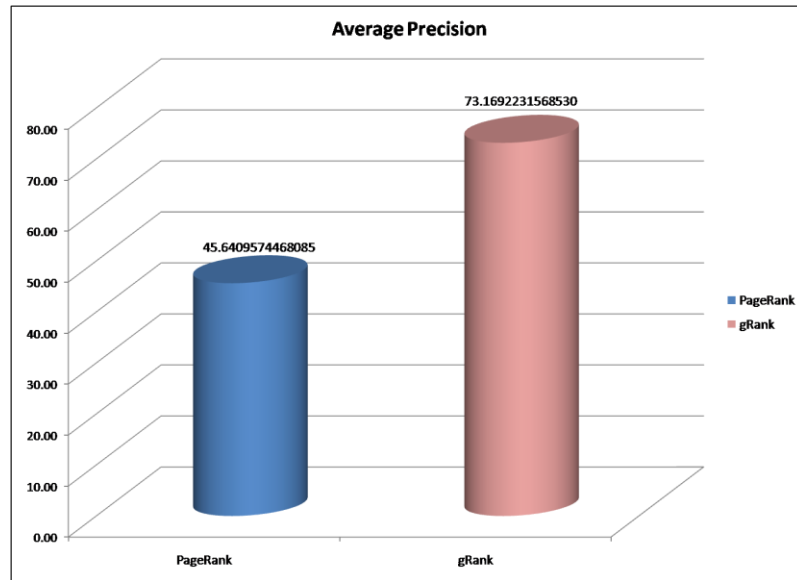


Figure 4.12 Average Precision for gRank and PageRank

The Figure 4.13 shows that the mean average precision value for gRank is approximately 61 while PageRank has a value of 43.

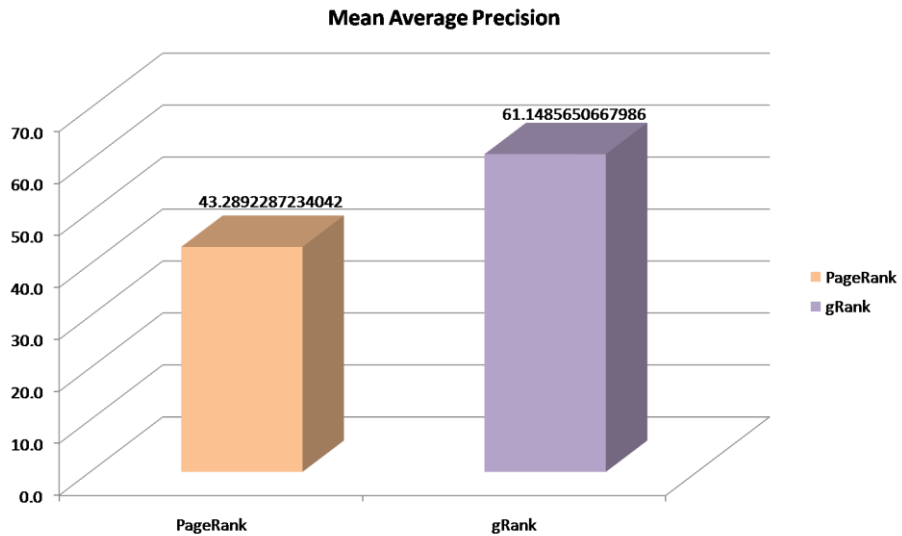


Figure 4.13 Mean Average Precision for gRank and PageRank

For measuring the ranking quality of the algorithm, Cumulative Gain, Discounted Cumulative Gain, Idealized Discounted Cumulative Gain, and Normalized Discounted Cumulative Gain is calculated. It has been found out that the final value of NDCG of gRank is more in comparison with the PageRank algorithm and the result is shown in Figure 4.14, Figure 4.15, Figure 4.16, and Figure 4.17.

The Figure 4.14 shows that the cumulative gain value for gRank is approximately 350 while PageRank has a value of 262.

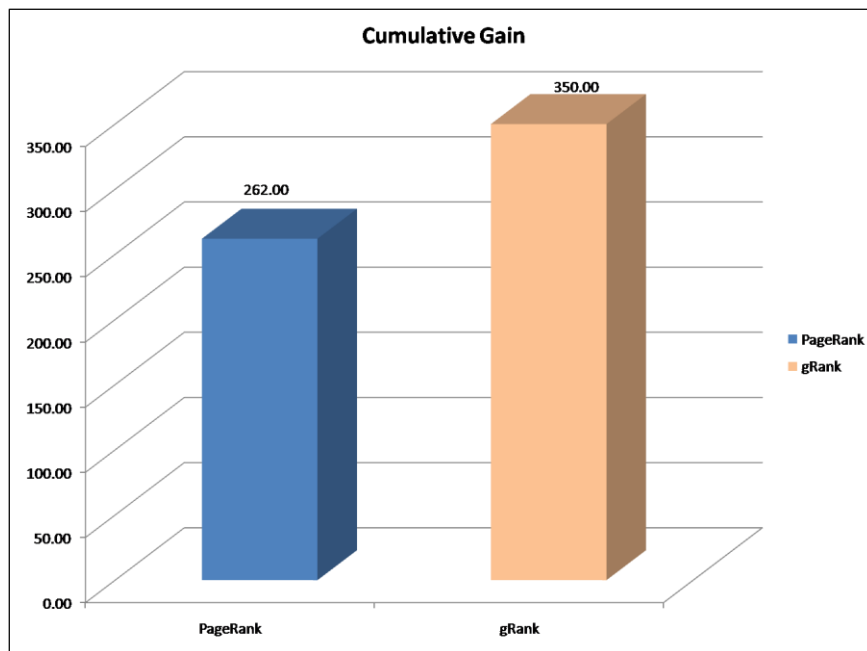


Figure 4.14 Cumulative Gain for gRank and PageRank

The Figure 4.15 shows that the discounted cumulative gain value for gRank is approximately 59 while PageRank has a value of 46.

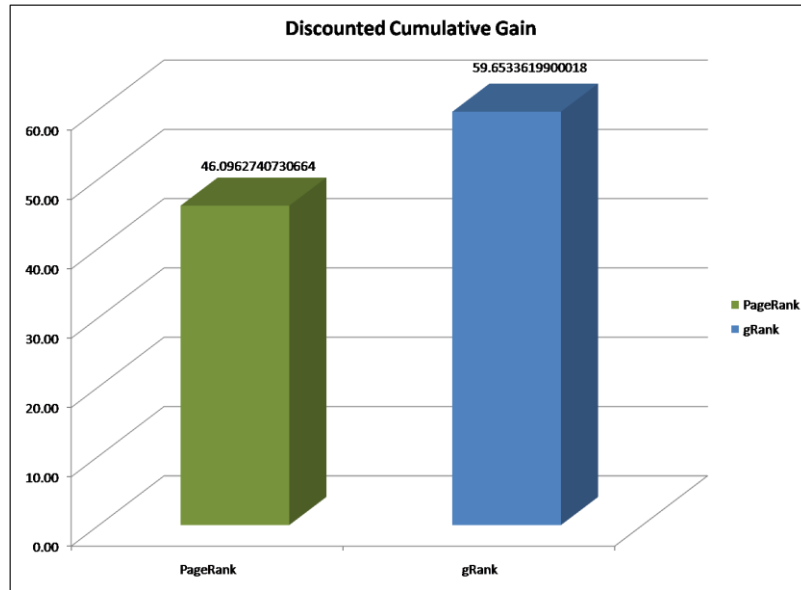


Figure 4.15 Discounted Cumulative Gain for gRank and PageRank

The Figure 4.16 shows that the idealized discounted cumulative gain value for gRank is approximately 68 while PageRank has a value of 55.

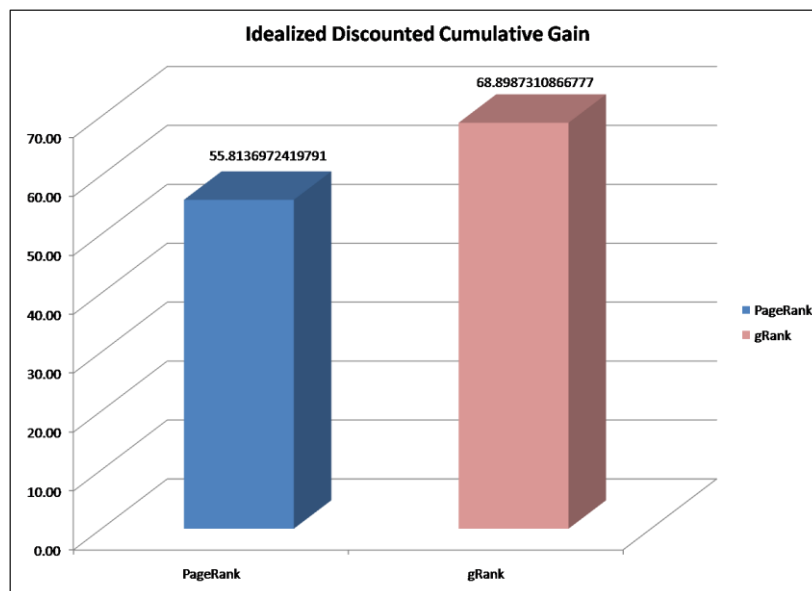


Figure 4.16 Idealized Discounted Cumulative Gain for gRank and PageRank

The Figure 4.17 shows that the normalized discounted cumulative gain value for gRank is approximately 0.86 while PageRank has a value of 0.82 that show an improvement in the ranking quality of the proposed algorithm.

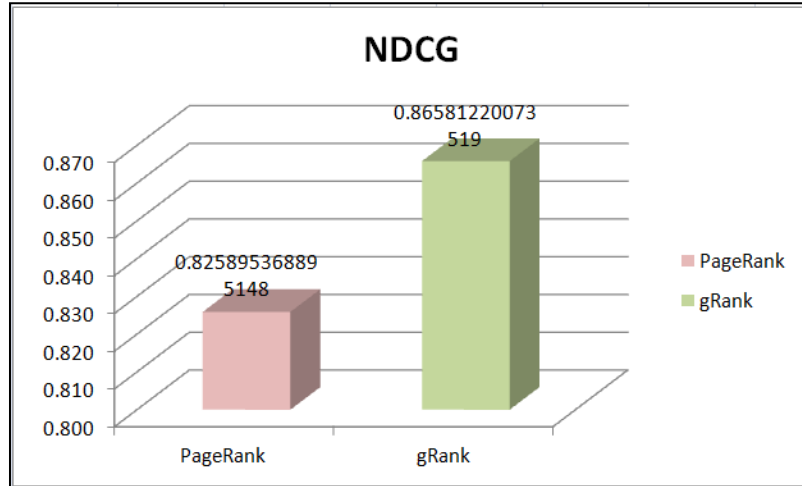


Figure 4.17 Representation of Normalized Discounted Cumulative Gain in gRank and PageRank

4.6 Summary

Genetic Algorithm based ranking system gRank is proposed that helps in accessing relevant pages based on user query. The proposed methodology provides a better and accurate result in less time duration. Content and structural links of the web pages are considered for providing relevant results. It clusters the nodes for the given data set based on content and leave irrelevant nodes that lack proper content and provide efficient result.

The proposed approach applies the genetic operators on the set of population and the nodes are ranked based higher fitness values. The results depict that if the node lacks proper content it is not considered for the ranking process even though its hyperlink count is more. The results are obtained in less number of iterations thus increasing the relevancy of the approach and also reduce the computational complexity of the process.

DESIGN AND IMPLEMENTATION OF BEERANK ALGORITHM USING ARTIFICIAL BEE COLONY ALGORITHM

5.1 Introduction

There is an incredible change in the World Wide Web and the users face difficulty in accessing the needed information as per their need. Search engines are working in different perspectives to improve the quality of results obtained. Different algorithms are devised at each step of the information retrieval process, and it is observed that ranking is one of the core ingredients of any search engine that has a paramount role in arranging the information. A good ranking system should incorporate various measures to improve the result quality and should not stick to a single measure. In this regard, different measures are adopted for web page ranking by using content, structure, or log data. In the coming years, it has been observed that various nature-inspired ranking algorithms are used which integrate different measures and help to optimize the quality of the result obtained. The BeeRank algorithm is proposed in this chapter that provides quality results, which is inspired by the Artificial Bee Colony algorithm for ranking web page and uses both the structural and content approach for calculating the rank value and provides better results. It also helps the users in uncovering the relevant web pages by minimizing the computational complexity of the process and achieves the result in minimum time duration. The working is illustrated and is compared with the traditional PageRank algorithm that incorporates only structural links, and the result shows an improvement in ranking and provides user-specific results.

5.2 Related work

Different work is being carried out in the domain of information retrieval by incorporating nature-inspired algorithms in real-life scenario. Deo et al., (2019) [113], presents a retrieval system for the web pages that uses PSO which identify and refine features set by storing them in a database, which later can be used for webpage retrieval. Tf-idf is used for the feature selection process. Karaboga et al., (2011) [114], proposed ABC for clustering data which can be applied for multivariate data clustering and classification benchmark problems. It is compared with PSO and other techniques used for optimization tasks. Ju et al., (2013) [115], proposes NABC (New Artificial Bee Colony) which helps in altering the search pattern for employed and onlooker bees. Neighborhood solutions are searched and new best solutions are generated. Anuradha and Lavanya, (2014) [116], suggested a new optimization perspective for solving the ranking problem. The given method assigns a rank to the web pages by considering various factors like users interest, total web site linkage, and growth analysis rate. Rani et al., (2017) [117], work on the referential concepts and present an approach of web page ranking using ant colony and ABC approach. User interest URLs are captured based on search requirements and the weights of the web pages are updated periodically. The Ant colony and Bee colony trace an ideal path for searching the desired result and produce a better result in comparison to the legacy algorithm.

Sakulin et al., (2018) [118], presents an interface optimization method for the web page interface with focusing on genetic algorithms (GAs), artificial bee colony algorithms (ABC), and charged system search algorithms (CSSs). Optimization is implemented using DOM models. For simple sample, GA provide good result while ABC gives result in minimum time duration, CSS help to elaborate the principle for interfacing elements. Gupta et al., (2017) [119], introduced a ranking algorithm based on feedback, preference, and web page characteristics. It helps in providing the best web pages according to user interest that facilitate, to enhance the user satisfaction and overall search experience. Karaboga and Akay, (2009) [87], work provides a relative analysis of Artificial Bee Colony (ABC) with differential evolution, swarm optimization and genetic algorithm, which is utilized for optimizing numerical test functions. The outcome of the

results depict that ABC uses less parameter and provide a better solution for large scale of unconstrained test functions. Kavitha et al., (2014) [120], proposes an approach that focuses on web document classification and considers both snippets and page count by using Artificial Bee Colony (ABC) algorithm that uses a clustering algorithm that focuses on the sequential pattern. Anuar et al., (2016) [121], introduced a change technique and model that help to enhance the performance of scout bee behavior in terms of exploration for the standard ABC called ABC-ROC (artificial bee colony rate of change). The proposed work keeps a track on the slope that is changing on the performance graph. Ju. C et al., (2013) [122], presents a novel approach that considers the K-means clustering algorithm for collaborative filtering recommendation. The Artificial Bee Colony (ABC) algorithm is used for the clustering process that helps in overcoming the difficulty of local optima thereafter cosine similarity is implemented for computing similarity between the users. Abdullah and Hadi, (2014) [123], proposed an algorithm MABC using Artificial Bee Colony which produces a better quality solution in terms of response time and the result evaluated can be used to judge the relevance of the approach for large scale data set more suited to the web environment. Gao et al., (2018) [124], presents an approach that helps in boosting the performance of onlooker bees and employed bees by using learning strategies from individuals that have better performance. It also balances the global and local searches by using new control operators and also helps in accelerating the worst employed bee convergence rate using an intelligence learning mechanism. Xiao et al., (2019) [125], presents an approach ABC-ESDL that works on dimension learning and elite strategy and therefore helps in accelerating the search of ABC by balancing exploration and exploitation. The exiting works try to ameliorate the ABC process by employing different techniques that enhance the working of scout bees or onlooker bees. The proposed work helps in improving the ideology, by just considering the layman users as bees that search for the honey, here we are considering desired information that is being searched by the users. The fitness value at each step is being improved to discover the best web pages that fulfill the quest of the users.

Gadekallu et al., (2020) [126], investigated the Diabetic Retinopathy, a vision loss problem, and suggested the use of the PCA and Firefly algorithm to resolve the problem. PCA helps extract significant features and the Firefly algorithm help in choosing the

optimal parameters which helps avoid local minima. Bhattacharya et al., (2020) [127] proposed a model for intrusion detection using a hybrid approach of PCA and firefly, which overall helps in dimensionality reduction. PCA help in removing redundant attributes from high dimension data and the firefly algorithm help in selecting relevant features thus reducing training time. Firefly algorithm can be employed to attract web pages that are rich in content thus improving the ranking of the web pages. Reddy et al.,(2019) [128], proposed fuzzy logic with an adaptive genetic algorithm that helps in diagnosing heart disease at early stages. Every rule of the classifiers is considered as chromosomes and is selected based on fitness value. We can also employ a genetic algorithm for ranking the web pages that are retrieved during query time. Iwendi et al., (2020) [129], works on increasing the network lifetime by reducing the energy consumption of sensors that are found in the IoT network by using a hybrid approach of Whale Optimization and Simulated Annealing. WOA is based on attacking pattern towards the prey, which is quite helpful in the proposed work. Gadekallu at al., (2020) [130], investigated Linear Discriminant Analysis (LDA) and Principal Component Analysis (PCA), a dimensionality reduction method on different Machine Learning algorithms by removing irrelevant attributes and patterns that helps in reducing the load on ML algorithms and can be employed for making relevant decisions. It is found that the results obtained by PCA are better than LDA.

5.3 Artificial Bee Colony

For solving high-dimensional problem nature-inspired optimization techniques are more preferred in comparison to classical optimization methods. Various optimizations algorithms are present that helps to understand the complexity of the real-world. The social behavior of various species like birds, fish, and insects leads to the rise of swarm intelligence algorithms that solve different optimization problems. ABC work on the foraging behavior of honey bee and is considered as an optimization algorithm.

Artificial bee colony (ABC) is a population-based stochastic swarm intelligence algorithm inspired by the behavior of honey bees. In many optimization problems it shows good search abilities and help to resolve distinct kinds of problems and provide effective and efficient results. Three types of bees are considered in Artificial bee colony

(ABC) algorithm- employed bees that help in searching food sources, onlooker bees help in selecting good food sources while scout bees abandon food sources and search new ones. The higher quality (fitness) food source will definitely have a greater likelihood in getting picked by the onlooker bees. [131].

1. Load training samples
2. Generate the initial population $z_i = 1 \dots SN$
3. Evaluate the fitness (f_i) of the population
4. Set cycle to 1
5. Repeat
6. **FOR** each employed bee {
 - Produce new solution y_i by using (6)
 - Calculate the value f_i
 - Apply greedy selection process }
7. Calculate the probability values p_i for the solutions (z_i) by (5)
8. **FOR** each onlooker bee {
 - Select a solution z_i depending on p_i
 - Produce new solution y_i
 - Calculate the value f_i
 - Apply greedy selection process }
9. **If** there is an abandoned solution for the scout
 - then** replace it with a new solution which will be randomly produced by (7)
10. Memorize the best solution so far
11. cycle = cycle + 1
12. **until** cycle= MCN

Figure 5.1 Pseudo-code of ABC Algorithm [87]

In ABC, the local search relies on neighborhood search using a greedy selection procedure, and a random search process is performed in a global search.

5.4 Proposed Methodology

The BeeRank algorithm is inspired by Artificial Bee Colony (ABC) algorithm and employs some of its characteristics where the food sources of bees are treated as web pages and the generated user query is the bee. As in normal Artificial Bee Colony approach the bees search for the flowers having the maximum nectar amount and extract nectar from them, inform other bees, and return to the hive. Similarly here we are considering the user query as bee which goes from pages to pages and collects its desired content from the web page which has the max quantity of the user query, the pages are prioritized in order of higher fitness value and are arranged in BeeRank order. The given approach undertakes both the structural linking of the web pages and the content for web page ranking and thus enhances the efficiency of the information retrieval process while other algorithms use solely link approach or content approach. By using both the approach high-quality pages can be retrieved in minimum time duration.

The user query extracts random web pages and abandoned the web pages having a single query term which are then evaluated to calculate the fitness value and thereby probability is calculated. The concentration of the food source and its neighbor is evaluated that is based on the best value obtained and the food source is updated and finally updated fitness and BeeRank are calculated. The proposed method does not singly handle the linking of the web pages as governed by the PageRank algorithm but the content of the pages is also considered thus improving the ranking of the nodes and the process stops when a similar value is obtained.

5.4.1 Architecture

The outline of the BeeRank is presented in Figure 5.2

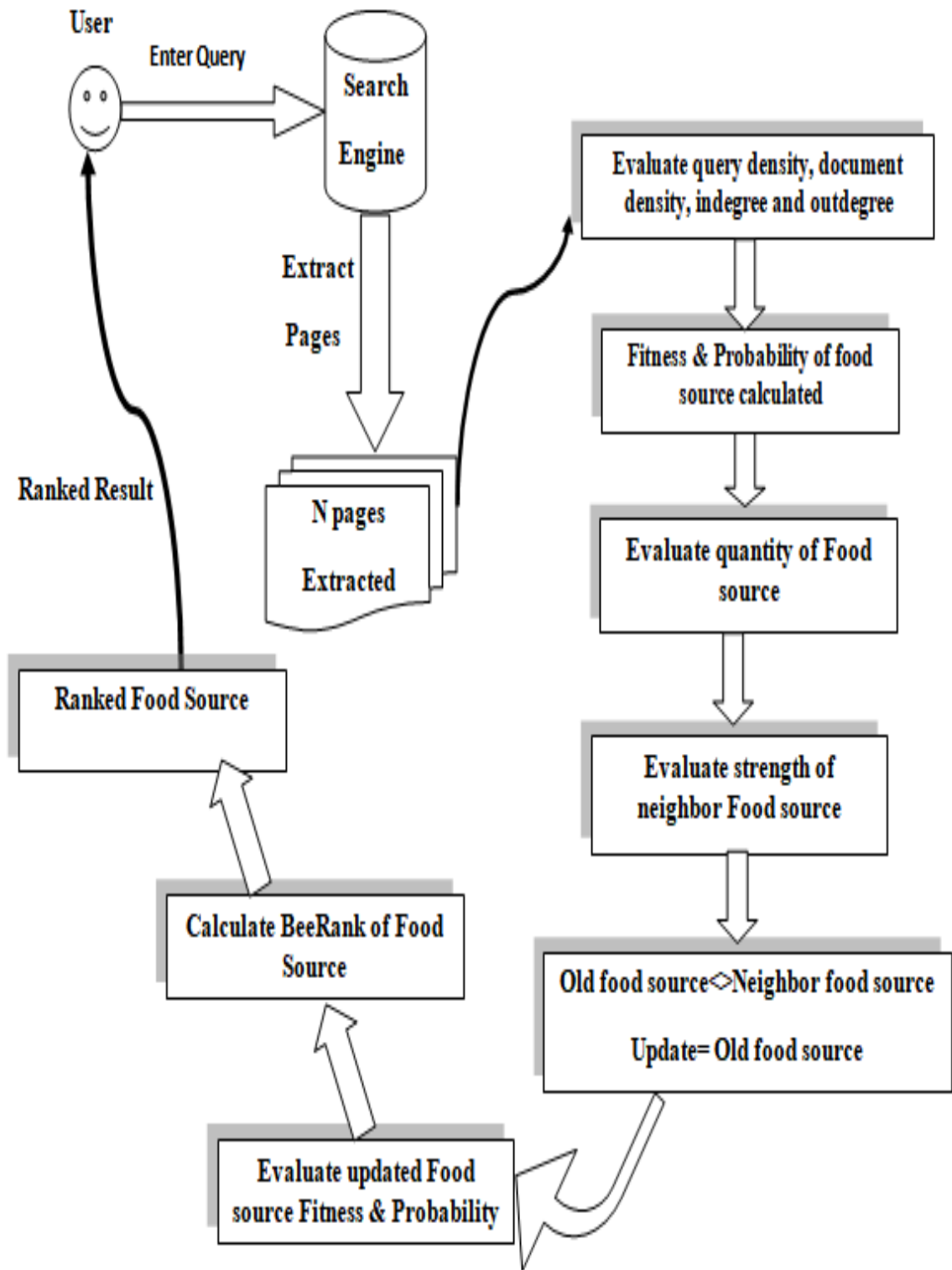


Figure 5.2 Proposed Architectural Model for BeeRank

5.4.2 Algorithm

The proposed algorithm provides ranked web pages that rely on user query and follow certain steps of the algorithm that is outlined as-

Algorithm BeeRank:

Input: Query as Bee

Output: Ranked web pages as Food Source (FS)

Steps:

1. Let there are randomly distributed Food Source (Web Pages) comprises of 'S' solution where each solution $S_i = \{i=1, 2, 3 \dots N\}$. $K \in \{1, 2, 3 \dots N\}$ represented in 'i' dimension, where $i! = k$. Query is represented by 'j' dimension vector where $j = \{1, 2 \dots Q\}$.
2. The Query 'q' having 'n' terms, randomly extracts 'W' pages by removing stopwords.
3. Query density 'qd', document density 'dd' and density of nodes having indegree and outdegree is evaluated.
4. Food source containing a single term of query is abandoned by the bee as it has low food amount, and new sources are searched.
5. Fitness of Food Source 'f(FS)' is calculated using-

$$f(FS) = \text{sqrt}[(dd(W) / qd)] / (I(W) / O(W))$$

where

dd- document density of web page

qd- query density

I (W) - Indegree of web page

O (W) - Outdegree of web page

6. Calculate probability $p(FS)$ of food source using-

$$p(FS) = f(FS) / \sum_{i=1}^N f(FS)$$

where

$f(FS)$ - fitness of food source

7. Calculate quantity of food source using-

$$x_{ij} = x_{j \min} + C[-1, 1](x_{j \max} - x_{j \min})$$

where

x_{ij} - i^{th} food source in j^{th} direction

$x_{j \min}$ -lower bound of food source in j^{th} direction

$x_{j \max}$ -upper bound of food source in j^{th} direction

C - random number between (-1, 1)

8. Calculate strength of neighbor food source using-

$$N_{ij} = x_{ij} + C_{ij}(x_{ij} - x_{kj})$$

where

N_{ij} - neighbor food source i^{th} in j^{th} direction

x_{ij} - i^{th} food source in j^{th} direction

x_{kj} - second vector ($i \neq k$)

C - random number between (-1, 1)

9. Old food source (x_{ij}) and new neighbor food source (N_{ij}) is compared and value of x_{ij} is updated by selecting the best value between them.

10. Calculate fitness of updated food source using-

$$f(UFS) = 1 / (1 + f(x_{ij}))$$

where

f (UFS) - fitness value after updation of food sources

f (x_{ij}) - fitness value for quantity of food sources

11. Calculate probability and probability percentage of new food source using-

$$p(UFS) = f(UFS) / \sum_{i=1}^N f(UFS)$$

where

p (UFS) – probability of updated food source

f (UFS) - fitness value after updation of food sources

12. BeeRank of food sources is calculated using-

$$BeeRank(W) = (1 - \alpha) + f(UFS) * \alpha$$

where

f (UFS) - fitness value after updation of food sources

α -constant (0-1) standard taken as 0.85

13. Finally ranked web pages.

5.4.3 Dataset Used

A dataset consisting of 8910 URLs are collected for testing the algorithm. The collected URLs are then tested on a set of different queries that generate a proper ranking result. A

total of 100 queries are taken and are tested on the collected URLs. A small part of the dataset and queries employed for evaluating the ranking result is shown for illustration purpose in Figure 5.3 and Figure 5.4. The proposed work is experimentally performed on Spyder (Python 3.7).

Index	0
0	https://searcherp.techtarget.com/definition...
1	https://www.investopedia.com/terms/s/scm.asp
2	https://www.knowledgekafe.com/course/supply-chain-management-125022
3	https://en.wikipedia.org/wiki/Supply_chain_management
4	https://www.cio.com/article/2439493/what-is-...
5	https://searcherp.techtarget.com/definition/supply-chain-management-SCM
6	https://en.wikipedia.org/wiki/Supply_chain_management#Importance
7	https://en.wikipedia.org/wiki/Supply_chain_...
8	https://en.wikipedia.org/wiki/Supply_chain_...
9	https://en.wikipedia.org/wiki/Supply_chain_management#Supply_chain
10	https://scm.ncsu.edu/scm-articles/article/what-is-supply-chain-management-scm
11	https://www.youtube.com/watch?v=1ZP05RclZEo
12	https://www.ibm.com/topics/supply-chain-management
13	https://www.oracle.com/applications/supply-chain-management/
14	https://www.coursera.org/specializations/supply-chain-management
15	https://www.tradegecko.com/supply-chain-management
16	https://cscmp.org/CSCMP/Develop/Starting_Yo...
17	https://www.managementstudyguide.com/supply-chain-management-definition.htm
18	https://www.emerald.com/insight/publication/issn/1359-8546
19	https://hbr.org/2018/06/the-death-of-supply-chain-management
20	https://www.edx.org/micromasters/mitx-supply-chain-management

Figure 5.3 Dataset of web pages

Index	0
0	Google Driver Less Car
1	Micro Machining
2	Solar Powered Aircraft
3	High Speed Machining
4	Quad Copter
5	Non-Pneumatic Tyres
6	Electrochemical Machining
7	Dual Fuel Engine
8	Methanol Fueled Marine Diesel Engine
9	Intelligent Variable Valve Timing
10	Solid Waste Management
11	Hemi Engine
12	Scram Jet Engine for Hypersonic Flight
13	Sidewinder Missile
14	Laser Cutting System
15	Airless Tyres
16	Pneumatic Tyres
17	Chassis Frame
18	Pulse Detonation Engine
19	Solar Collector
20	Semi Active Suspension System

Figure 5.4 Queries tested on web pages

5.4.4 Experimental Analysis

For implementing the BeeRank approach a directed graph comprises nodes A, B, C, D, E, F, and G is taken as shown in Figure 5.5.

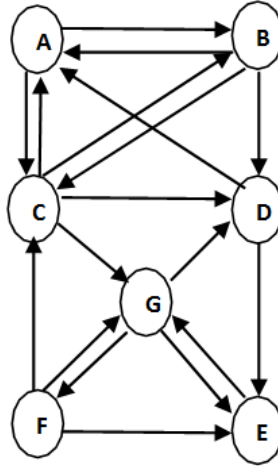


Figure 5.5 Structure of web page

The occurrence of respective terms on the web page for a query is shown in Table 5.1.

Table 5.1 Occurrence of terms in Web Pages

Terms\ Nodes	A	B	C	D	E	F	G
Information	20	8	8	10	20	0	25
Retrieval	10	12	13	15	10	0	0
Evolutionary	12	0	6	8	5	0	0
Computation	8	0	2	6	2	6	0

The food sources that contain a single query term are abandoned by the bee initially, thus filtering the food sources. The fitness value, probability and probability percentage of the food sources of the remaining nodes are evaluated using the formula shown in 5.1 and

5.2.

$$f(FS) = \text{sqrt}[(dd(W) / qd)] / (I(W) / O(W)) \quad (5.1)$$

$$p(FS) = f(FS) / \sum_{i=1}^N f(FS) \quad (5.2)$$

The I (W), O (W), fitness, and the probability of the initial phase are shown in Table 5.2.

Table 5.2 I (W), O (W), Fitness and probability of the nodes

W	qd	dd	I(W)	O(W)	f(FS)	p(FS)	p%
A	4	50	3	2	5.035533906	0.228423466	22.84234657
B	4	20	2	3	2.902734644	0.131674758	13.16747578
C	4	29	3	4	3.442582404	0.156163501	15.61635009
D	4	39	3	2	4.622498999	0.209687247	20.9687247
E	4	37	3	1	6.041381265	0.274051029	27.40510286
					22.04473122		

After evaluating the Fitness value, the concentration of food (x_{ij}) in the food source is evaluated using the formula shown in 5.3.

$$x_{ij} = x_{j \text{ min}} + C[-1, 1](x_{j \text{ max}} - x_{j \text{ min}}) \quad (5.3)$$

After evaluation of the food source, the neighbor food sources are estimated (N_{ij}) using the formula shown in 5.4.

$$N_{ij} = x_{ij} + C_{ij}(x_{ij} - x_{kj}) \quad (5.4)$$

The value of old food source and new neighbor food source is compared and an update is done in (xij) for assessing the best value, now updated fitness of food source and probability is evaluated using formula shown in 5.5 and 5.6.

$$f(UFS) = 1 / (1 + f(x_{ij})) \quad (5.5)$$

$$p(UFS) = f(UFS) / \sum_{i=1}^N f(UFS) \quad (5.6)$$

Finally BeeRank is calculated for ranking the food source using the formula shown in 5.7.

$$BeeRank(W) = (1 - \alpha) + f(UFS) * \alpha \quad (5.7)$$

The result of the process can be visualized in the 1st iteration in Table 5.3.

Table 5.3 Value of xij, Nij, f (UFS) and p (UFS) of the nodes in 1st iteration

W	f(FS)	xij	Nij	New xij	f(UFS)	p(UFS)	p%	BeeRank
E	6.041381265	4.472057955	4.723519794	4.72351979425545	0.174717663	0.161504269	16.15042687	0.298510013
A	5.035533906	3.969134275	3.717672435	3.96913427575	0.201242298	0.186022922	18.60229222	0.321055953
D	4.622498999	3.762616822	3.407896255	3.7626168215215	0.209968603	0.194089282	19.40892819	0.328473312
C	3.442582404	3.172658524	2.522958809	3.17265852424	0.239655365	0.221530919	22.15309193	0.35370706
B	2.902734644	2.902734644	2.118072989	2.90273464444	0.25623059	0.236852608	23.6852608	0.367796001
					1.081814518			

The process is repeated and the next iterations are processed as shown in Table 5.4.

Table 5.4 Value of x_{ij} , N_{ij} , f (UFS) and p (UFS) of the nodes in 2nd iteration

W	f(FS)	x_{ij}	N_{ij}	New x_{ij}	f(UFS)	p(UFS)	p%	BeeRank
E	0.174717663	0.215474127	0.208842968	0.2154741265265	0.822724218	0.20507796	20.50779603	0.849315585
A	0.201242298	0.228736444	0.235367603	0.2353676027544	0.809475655	0.201775532	20.17755315	0.838054307
D	0.209968603	0.233099597	0.241912332	0.2419123315965	0.805209816	0.200712199	20.07121987	0.834428344
C	0.239655365	0.247942978	0.264177403	0.264177403775	0.791028219	0.197177195	19.71771952	0.822373986
B	0.25623059	0.25623059	0.276608822	0.276608821759	0.783325309	0.195257114	19.52571143	0.815826513
					4.011763217			

These steps are repeated and it is found that after a few iterations the values are repeated and the process can be terminated. The fitness values of the entire food sources are presented in Table 5.5 and illustrated in Figure 5.6.

Table 5.5 Fitness Values in Different Iterations

WebPage	f-I	f-II	f-III	f-IV	f-V	f-VI	f-VII	f-VIII	f-IX	f-X
E	0.174717663	0.822724218	0.553606582	0.642187183	0.609264734	0.621305885	0.616806862	0.618496445	0.617858696	0.618100476
A	0.201242298	0.809475655	0.556668744	0.641241441	0.609528144	0.621229632	0.61682862	0.618490203	0.617860483	0.618099964
D	0.209968603	0.805209816	0.557330479	0.641037431	0.609566045	0.621218663	0.616830707	0.618489604	0.617860598	0.618099931
C	0.239655365	0.791028219	0.559541742	0.640356651	0.60969255	0.62118205	0.616837673	0.618487605	0.617860977	0.618099822
B	0.25623059	0.783325309	0.560750187	0.639985219	0.609761593	0.621162069	0.616841474	0.618486515	0.617861187	0.618099762

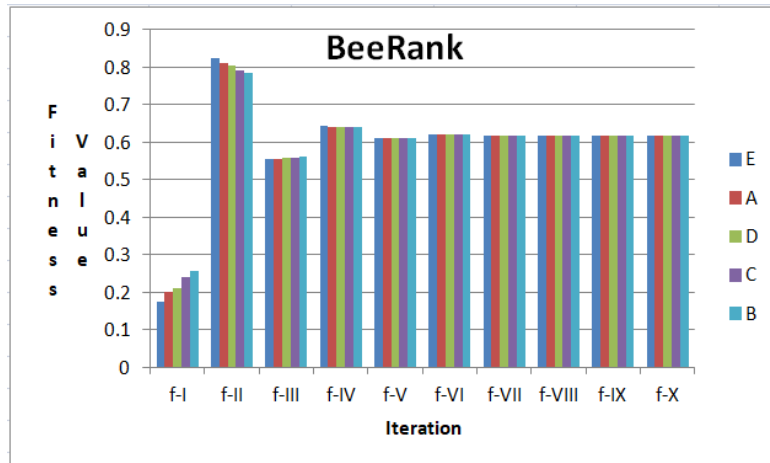


Figure 5.6 Representation of Fitness value in different Iterations

Similarly, the BeeRank values in each iteration is depicted in Table 5.6 and shown in Figure 5.7.

Table 5.6 BeeRank Values in Different Iterations

WebPage	BR-I	BR-II	BR-III	BR-IV	BR-V	BR-VI	BR-VII	BR-VIII	BR-IX	BR-X
E	0.298510013	0.849315585	0.620565595	0.695859105	0.667875024	0.678110002	0.674285833	0.675721978	0.675179892	0.675385405
A	0.321055953	0.838054307	0.623168432	0.695055225	0.668098922	0.678045188	0.674304327	0.675716672	0.675181411	0.675384969
D	0.328473312	0.834428344	0.623730907	0.694881817	0.668131138	0.678035863	0.674306101	0.675716163	0.675181508	0.675384941
C	0.35370706	0.822373986	0.625610481	0.694303153	0.668238667	0.678004742	0.674312022	0.675714464	0.675181831	0.675384849
B	0.367796001	0.815826513	0.626637659	0.693987436	0.668297354	0.677987759	0.674315253	0.675713537	0.675182009	0.675384798

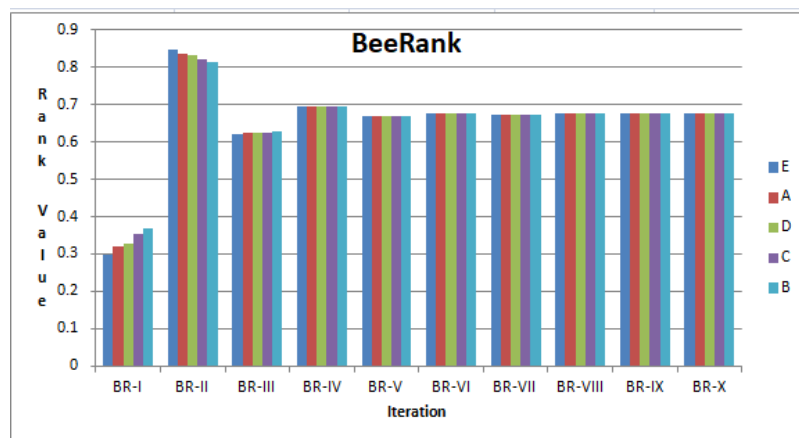


Figure 5.7 Representation of BeeRank value in different Iterations

The final ranking of the nodes is done based on BeeRank value which is directly

proportional to selection and is shown in Table 5.7 and illustrated graphically in Figure 5.8.

Table 5.7 Ranked Nodes along with BeeRank Values

WebPage	Rank
E	0.675385405
A	0.675384969
D	0.675384941
C	0.675384849
B	0.675384798

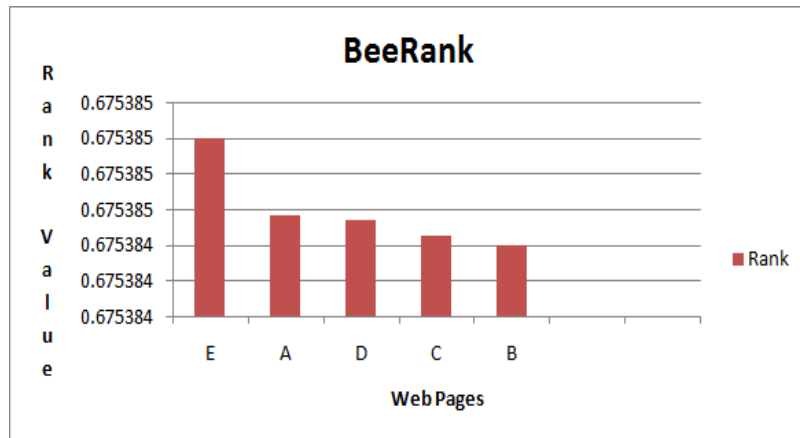


Figure 5.8 Ranking of Nodes in BeeRank Model

The proposed approach helps in reaching the desired goal in less number of steps which overall reduces the computational calculation and complexity of the given method. Following less number of iterations to obtain the results also help to decrease the time duration of the procedure.

The PageRank algorithm increases the complexity of the ranking process and also takes more time to reach the desired goal and it is found that the result is not obtained till 27 iterations while in BeeRank we are getting the result after 8 iterations. Table 5.8 depicts the result of PageRank and Table 5.9 finally ranked nodes and is illustrated in Figure 5.9.

Table 5.8 PageRank of Nodes

Iteration\ Node	PR(A)	PR(B)	PR(C)	PR(D)	PR(E)	PR(F)	PR(G)
0	1	1	1	1	1	1	1
1	1.070833333	0.817604167	1.120092014	0.9030074	1.100444812	0.433333333	1.446175421
2	1.003452212	0.814486743	0.930016212	0.988149391	1.102490972	0.559749703	1.44334152
3	0.99836318	0.771932797	0.951614393	0.979879115	1.133991137	0.558946764	1.474478775
4	0.987380975	0.771854973	0.946697406	0.987634427	1.145881868	0.567768986	1.486040666
5	0.989610073	0.77175748	0.95011678	0.99160929	1.153346683	0.571044855	1.494040539
6	0.991998383	0.773499129	0.952553442	0.994887179	1.157934579	0.573311486	1.499100253
7	0.994402744	0.775038772	0.954653739	0.99720331	1.160994733	0.574745072	1.502553879
8	0.996269645	0.776278519	0.956204617	0.998862661	1.163084667	0.575723599	1.504937134
9	0.997655692	0.77719715	0.957331215	1.000037597	1.16453652	0.576398855	1.506601934
10	0.998654721	0.77786114	0.958135255	1.000868279	1.165552575	0.576870548	1.507770086
11	0.99936675	0.77833461	0.958705664	1.001454617	1.166266392	0.577201524	1.508591819
12	0.999871306	0.778670258	0.959108977	1.001868246	1.166768785	0.577434349	1.509170524
13	1.000227902	0.778907516	0.95939372	1.002159943	1.16712269	0.577598315	1.509578308
14	1.000479604	0.779074997	0.959594604	1.002365623	1.1673721	0.577713854	1.50986573
15	1.000657159	0.779193146	0.959736276	1.00251064	1.167547904	0.57779529	1.510068343
16	1.000782372	0.779276467	0.959836172	1.002612883	1.167671838	0.577852697	1.51021118
17	1.000870661	0.779335217	0.959906607	1.002684966	1.167759209	0.577893168	1.510311879
18	1.00093291	0.77937664	0.959956265	1.002735787	1.167820806	0.577921699	1.510382873
19	1.000976797	0.779405845	0.959991276	1.002771616	1.167864232	0.577941814	1.510432924
20	1.001007739	0.779426435	0.96001596	1.002796877	1.167894849	0.577955995	1.510468211
21	1.001029554	0.779440952	0.960033362	1.002814686	1.167916433	0.577965993	1.510493089
22	1.001044934	0.779451186	0.960045631	1.002827241	1.167931651	0.577973042	1.510510628
23	1.001055777	0.779458402	0.960054281	1.002836093	1.16794238	0.577978011	1.510522994
24	1.001063422	0.779463489	0.960060379	1.002842334	1.167949943	0.577981515	1.510531712
25	1.001068811	0.779467075	0.960064679	1.002846734	1.167955276	0.577983985	1.510537858
26	1.001072611	0.779469604	0.96006771	1.002849836	1.167959036	0.577985726	1.510542191
27	1.00107529	0.779471386	0.960069847	1.002852023	1.167961686	0.577986954	1.510545246
28	1.001077178	0.779472643	0.960071353	1.002853565	1.167963555	0.57798782	1.5105474
29	1.00107851	0.779473529	0.960072416	1.002854652	1.167964872	0.57798843	1.510548918
30	1.001079448	0.779474154	0.960073164	1.002855418	1.167965801	0.57798886	1.510549989

Table 5.9 Ranked Nodes

Ranked Nodes
G
E
D
A
C
B
F

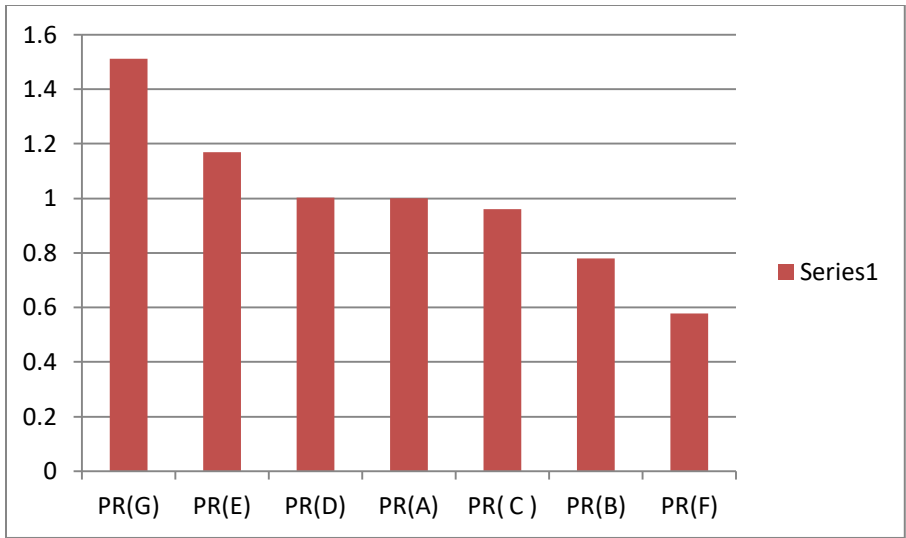


Figure 5.9 Ranking of Nodes in PageRank algorithm

It is observed that for some node the hyperlink count is high, but owing to paucity of relevant content it is not considered high for ranking as depicted in BeeRank algorithm and can be eliminated initially if irrelevant content is found, as it considers both structure and content approach, but in PageRank algorithm, it is at top position showing scantiness of the approach.

The result of the BeeRank and PageRank on the number of iteration needed to achieve the result is shown in Table 5.10.

Table 5.10 Iteration Required

	PageRank	BeeRank
Iteration required after which the desired result is achieved	27	8

It has been found by observing the result that the PageRank algorithm is unable to achieve the result till 27 iterations while if we use the BeeRank approach then results are obtained after 8 iterations and we are getting similar values which help to terminate the process.

The proposed algorithm is now implemented on the given dataset having 8910 URLs and tested on 100 queries using Spyder (Python 3.7). The document density and query density of the URLs is represented in Table 5.11.

Table 5.11 Density of Nodes and Query

Index	ind	Supply	Chain	lanagemei	i(w)	o(w)	w(x)	q(n)
0	1	94	95	61	7	10	250	3
1	2	34	37	16	2	5	87	3
2	3	21	21	26	1	4	68	3
3	4	317	308	241	26	29	866	3
4	5	64	65	16	5	8	145	3
5	6	94	95	61	7	10	250	3
6	7	317	308	241	26	29	866	3
7	8	317	308	241	26	29	866	3
8	9	317	308	241	26	29	866	3
9	10	317	308	241	26	29	866	3
10	11	25	24	13	2	5	62	3
11	12	14	16	13	1	4	43	3
12	13	57	60	29	4	7	146	3
13	14	29	28	14	2	5	71	3
14	15	45	45	22	3	6	112	3
15	16	54	52	30	4	7	136	3
16	17	45	45	8	3	6	98	3
17	18	28	26	72	2	5	126	3
18	19	51	49	16	4	7	116	3
19	20	29	28	8	2	5	65	3
20	22	17	17	9	1	4	43	3

The result of the BeeRank process can be visualized in different iteration and is depicted in Table 5.12.

Table 5.12 Processing steps of Different Iteration in BeeRank

Index	ffs0	xij0	nij0	new_nij0	fufs0	bee0	ffs1	xij1	nij1	new_nij1	fufs1	bee1	ffs2	xij2	nij2	new_nij2	fufs2	bee2	ffs3
0	34.0892	17.5446	2.71957	18	0.0526316	0.194737	0.0526316	0.058623	0	0	1	1	1	0.502959	0.125	1	0.5	0.575	0.5
1	23.211	12.1055	2.1812	12	0.0769231	0.215385	0.0769231	0.058623	1	1	0.5	0.575	0.5	0.252959	1.04167	1	0.5	0.575	0.5
2	18.4862	9.74308	0.818803	10	0.0909091	0.227273	0.0909091	0.058623	2	2	0.333333	0.433333	0.333333	0.169625	1.95833	2	0.333333	0.433333	0.333333
3	18.0021	9.50103	1.69778	10	0.0909091	0.227273	0.0909091	0.058623	3	3	0.25	0.3625	0.25	0.127959	2.9375	3	0.25	0.3625	0.25
4	17.9436	9.47182	2.68317	9	0.1	0.235	0.1	0.058623	4	4	0.2	0.32	0.2	0.102959	3.925	4	0.2	0.32	0.2
5	17.9436	9.47182	3.68317	9	0.1	0.235	0.1	0.058623	5	5	0.166667	0.291667	0.166667	0.0862919	4.91667	5	0.166667	0.291667	0.166667
6	17.9436	9.47182	4.68317	9	0.1	0.235	0.1	0.058623	6	6	0.142857	0.271429	0.142857	0.0743872	5.91071	6	0.142857	0.271429	0.142857
7	17.9436	9.47182	5.68317	9	0.1	0.235	0.1	0.058623	7	7	0.125	0.25625	0.125	0.0654586	6.90625	7	0.125	0.25625	0.125
8	17.9436	9.47182	6.68317	9	0.1	0.235	0.1	0.058623	8	8	0.111111	0.244444	0.111111	0.0585141	7.90278	8	0.111111	0.244444	0.111111
9	17.0477	9.02387	7.4592	9	0.1	0.235	0.1	0.058623	9	9	0.1	0.235	0.1	0.0529586	8.9	9	0.1	0.235	0.1
10	15.9726	8.48628	8.1904	8	0.111111	0.244444	0.111111	0.058623	10	10	0.0909091	0.227273	0.0909091	0.0484131	9.89773	10	0.0909091	0.227273	0.0909091
11	15.6614	8.33071	9.11262	9	0.1	0.235	0.1	0.0530675	10.9972	11	0.0833333	0.220833	0.0833333	0.0446252	10.8958	11	0.0833333	0.220833	0.0833333
12	15.4556	8.22779	10.0612	10	0.0909091	0.227273	0.0909091	0.048522	11.9949	12	0.0769231	0.215385	0.0769231	0.0414201	11.8942	12	0.0769231	0.215385	0.0769231
13	14.964	7.98201	10.9383	11	0.0833333	0.220833	0.0833333	0.0447342	12.9931	13	0.0714286	0.210714	0.0714286	0.0386729	12.8929	13	0.0714286	0.210714	0.0714286
14	14.8997	7.94983	11.9222	12	0.0769231	0.215385	0.0769231	0.041529	13.9915	14	0.0666667	0.206667	0.0666667	0.0362919	13.8917	14	0.0666667	0.206667	0.0666667
15	14.5831	7.79155	12.843	13	0.0714286	0.210714	0.0714286	0.0387818	14.9901	15	0.0625	0.203125	0.0625	0.0342086	14.8906	15	0.0625	0.203125	0.0625
16	14.289	7.6445	13.7695	14	0.0666667	0.206667	0.0666667	0.0364008	15.9889	16	0.0588235	0.2	0.0588235	0.0323703	15.8897	16	0.0588235	0.2	0.0588235
17	14.2639	7.63194	14.7632	15	0.0625	0.203125	0.0625	0.0343175	16.9878	17	0.0555556	0.197222	0.0555556	0.0307364	16.8889	17	0.0555556	0.197222	0.0555556
18	14.2537	7.62683	15.7607	16	0.0588235	0.2	0.0588235	0.0324792	17.9869	18	0.0526316	0.194737	0.0526316	0.0292744	17.8882	18	0.0526316	0.194737	0.0526316
19	13.8444	7.42218	16.6584	17	0.0555556	0.197222	0.0555556	0.0308453	18.9861	19	0.05	0.1925	0.05	0.0279586	18.8875	19	0.05	0.1925	0.05
20	13.8361	7.41807	17.6563	18	0.0526316	0.194737	0.0526316	0.0293833	19.9854	20	0.047619	0.190476	0.047619	0.0267681	19.8859	20	0.047619	0.190476	0.047619

The different fitness values of the entire food sources are presented in Table 5.13 and are illustrated in Figure 5.10

Table 5.13 Fitness values during Iterations

Index	ffs1	ffs2	ffs3	ffs4
0	0.0526316	1	0.5	1
1	0.0769231	0.5	0.5	0.5
2	0.0909091	0.333333	0.333333	0.333333
3	0.0909091	0.25	0.25	0.25
4	0.1	0.2	0.2	0.2
5	0.1	0.166667	0.166667	0.166667
6	0.1	0.142857	0.142857	0.142857
7	0.1	0.125	0.125	0.125
8	0.1	0.111111	0.111111	0.111111
9	0.1	0.1	0.1	0.1
10	0.111111	0.0909091	0.0909091	0.0909091
11	0.1	0.0833333	0.0833333	0.0833333
12	0.0909091	0.0769231	0.0769231	0.0769231
13	0.0833333	0.0714286	0.0714286	0.0714286
14	0.0769231	0.0666667	0.0666667	0.0666667
15	0.0714286	0.0625	0.0625	0.0625
16	0.0666667	0.0588235	0.0588235	0.0588235
17	0.0625	0.0555556	0.0555556	0.0555556
18	0.0588235	0.0526316	0.0526316	0.0526316
19	0.0555556	0.05	0.05	0.05
20	0.0526316	0.047619	0.047619	0.047619

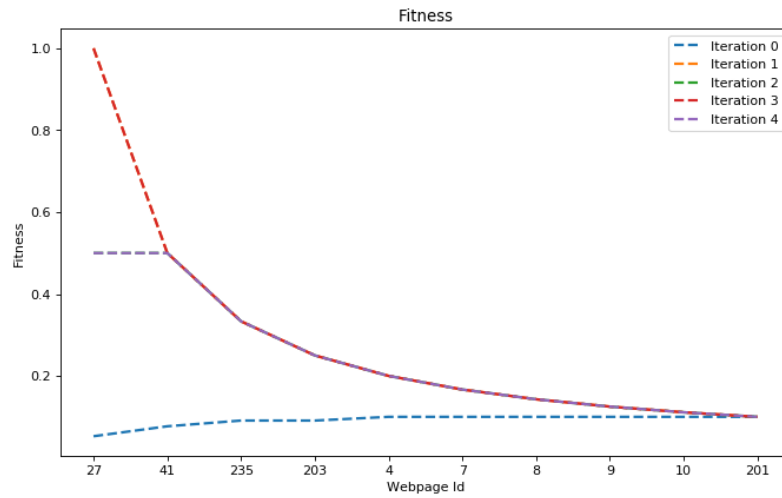


Figure 5.10 Fitness Graph during Iterations

Similarly the BeeRank values in each iteration is depicted in Table 5.14

Table 5.14 BeeRank value during different iterations

Index	bee0	bee1	bee2	bee3
0	0.194737	1	0.575	1
1	0.215385	0.575	0.575	0.575
2	0.227273	0.433333	0.433333	0.433333
3	0.227273	0.3625	0.3625	0.3625
4	0.235	0.32	0.32	0.32
5	0.235	0.291667	0.291667	0.291667
6	0.235	0.271429	0.271429	0.271429
7	0.235	0.25625	0.25625	0.25625
8	0.235	0.244444	0.244444	0.244444
9	0.235	0.235	0.235	0.235
10	0.244444	0.227273	0.227273	0.227273
11	0.235	0.220833	0.220833	0.220833
12	0.227273	0.215385	0.215385	0.215385
13	0.220833	0.210714	0.210714	0.210714
14	0.215385	0.206667	0.206667	0.206667
15	0.210714	0.203125	0.203125	0.203125
16	0.206667	0.2	0.2	0.2
17	0.203125	0.197222	0.197222	0.197222
18	0.2	0.194737	0.194737	0.194737
19	0.197222	0.1925	0.1925	0.1925
20	0.194737	0.190476	0.190476	0.190476

After performing the steps of the algorithm the final ranking of the URLs in the BeeRank algorithm is depicted in Table 5.15.

Table 5.15 Final BeeRank values

Index	rank	val_bee
0	27	34.0892
1	41	23.211
2	235	18.4862
3	203	18.0021
4	4	17.9436
5	7	17.9436
6	8	17.9436
7	9	17.9436
8	10	17.9436
9	201	17.0477
10	72	15.9726
11	234	15.6614
12	217	15.4556
13	205	14.964
14	215	14.8997
15	89	14.5831
16	208	14.289
17	137	14.2639
18	218	14.2537
19	232	13.8444
20	231	13.8361

5.5 Result Analysis

The performance of BeeRank and PageRank can be illustrated based on different evaluation measures that help in accessing the ranking quality of the algorithm. The results indicate that the computational complexity can be minimized which helps in saving time and effort. The execution time of the running process is graphically represented in Figure 5.11(a) and 5.11(b)

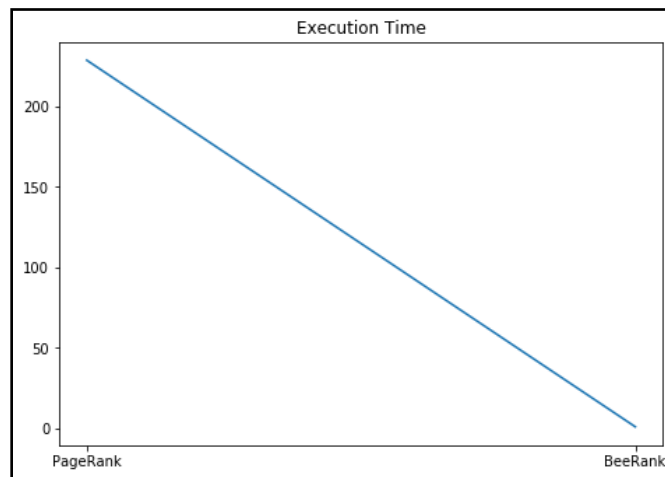


Figure 5.11(a) Execution time of BeeRank and PageRank

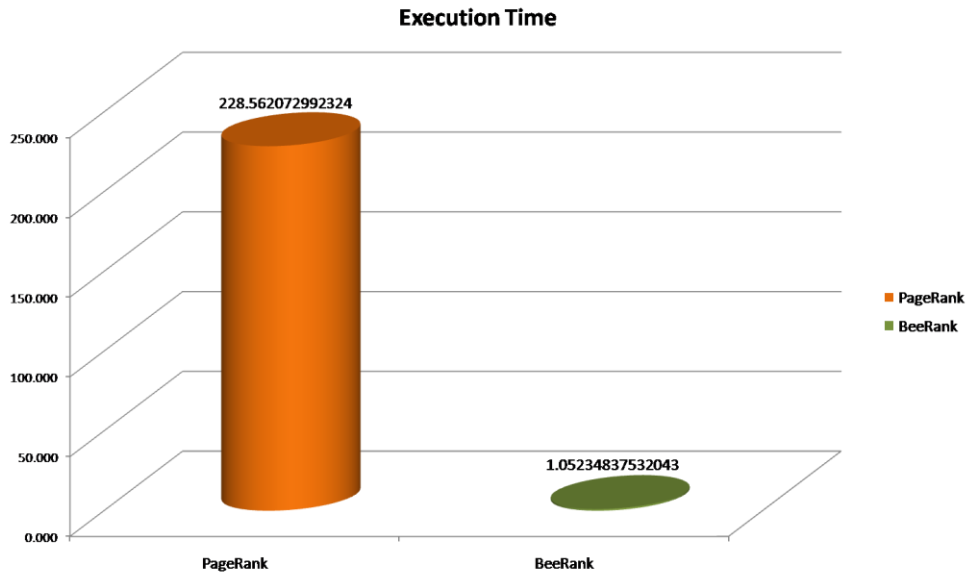


Figure 5.11(b) Bar Graph of Execution time for BeeRank and PageRank

The number of iterations to perform the working of BeeRank and PageRank is graphically illustrated in Figure 5.12 that shows that BeeRank takes 4 iterations to complete the processing while PageRank require 25 iterations.

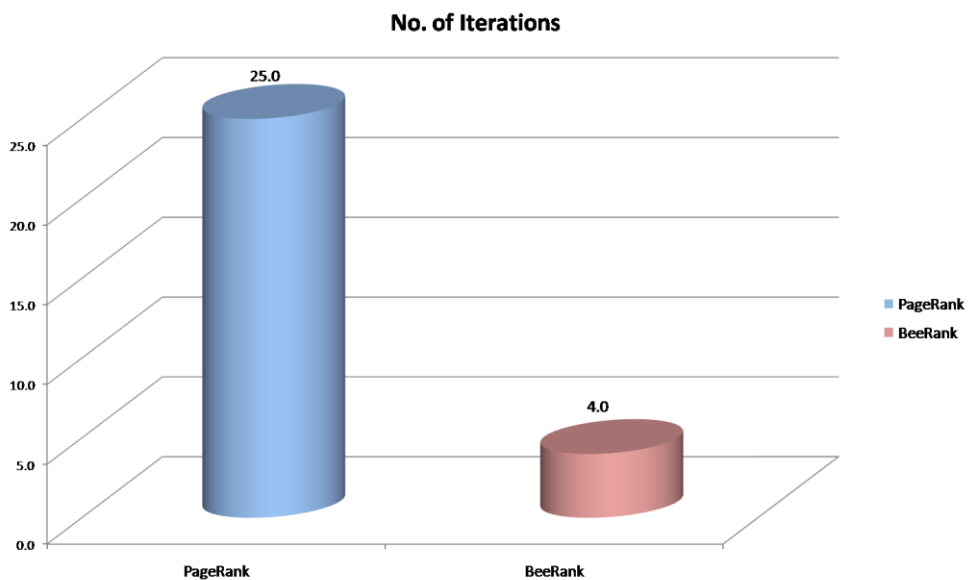


Figure 5.12 Number of iterations for BeeRank and PageRank

Figure 5.13(a) and Figure 5.13(b) depict the results of different iterations of web pages in the BeeRank and PageRank algorithm.

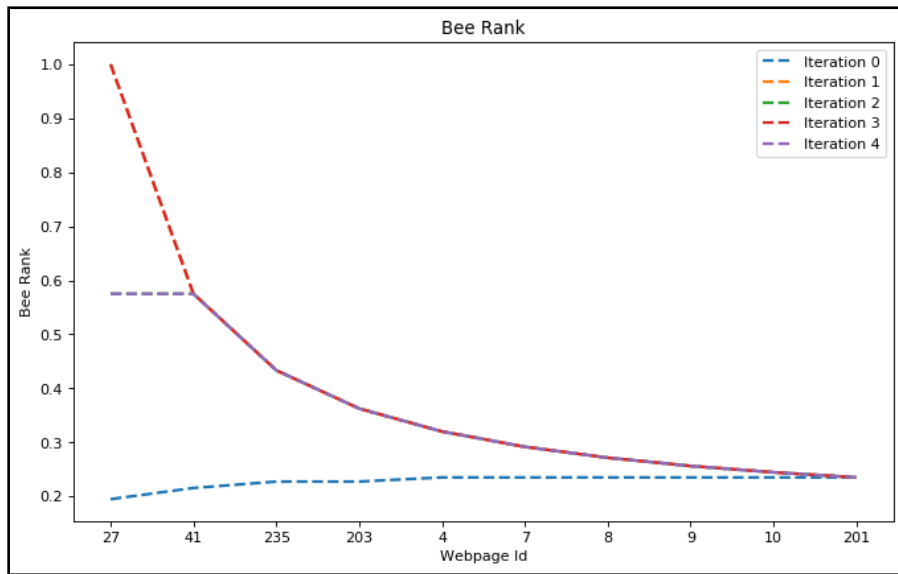


Figure 5.13(a) Result of iterations in BeeRank

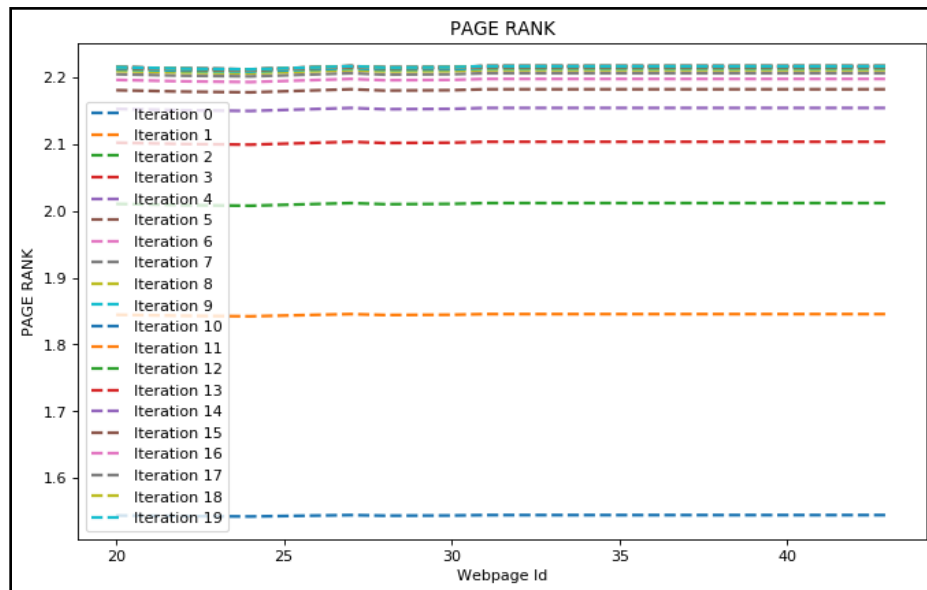


Figure 5.13(b) Result of iterations in PageRank

To evaluate measure and to access the ranking quality of BeeRank with PageRank different evaluating measures like Precision, Recall, Average Precision, and Mean Average Precision is performed and the results are shown in Figure 5.14, Figure 5.15, Figure 5.16 and Figure 5.17. The Figure 5.14 shows that the precision value for BeeRank is approximately 53 while PageRank has a value of 40, which shows high performance of BeeRank algorithm.

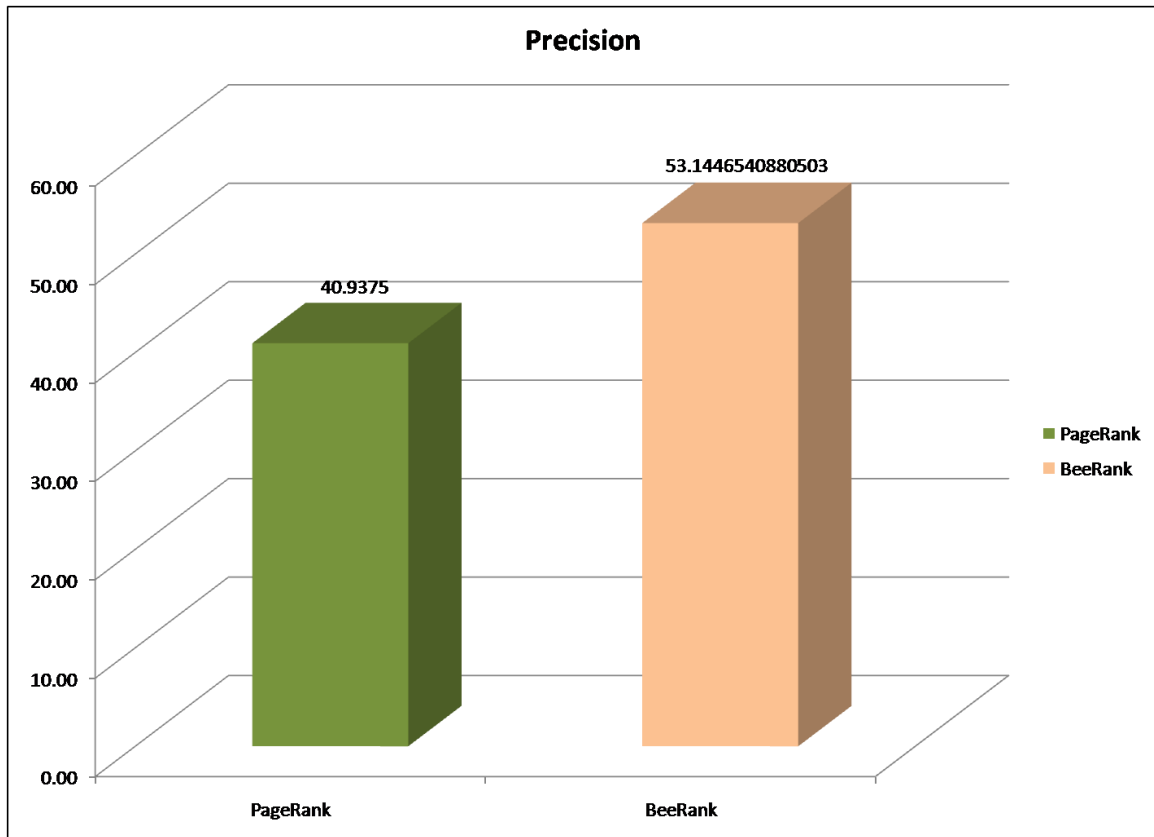


Figure 5.14 Precision values for BeeRank and PageRank

The Figure 5.15 shows that the recall value for BeeRank is approximately 86 while PageRank has a value of 46.

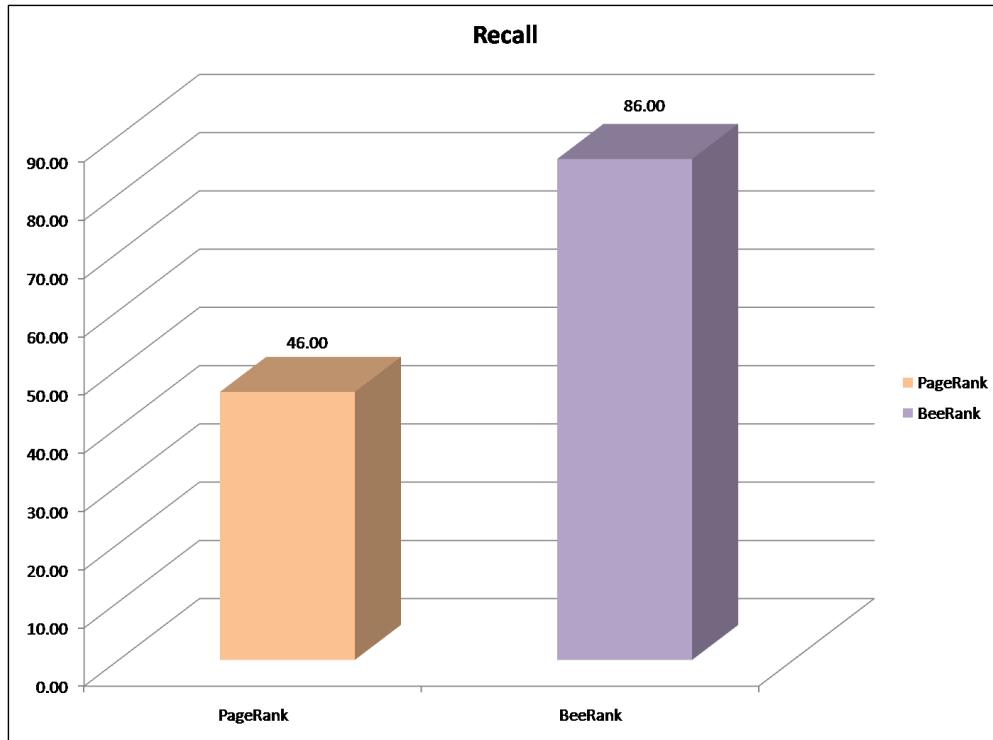


Figure 5.15 Recall values for BeeRank and PageRank

The Figure 5.16 shows that the average precision value for BeeRank is approximately 95 while PageRank has a value of 45.

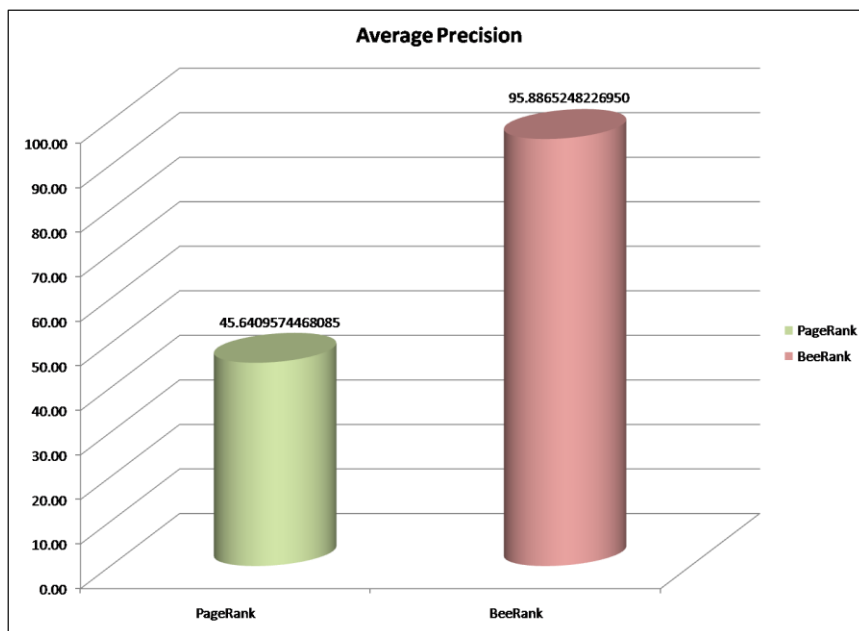


Figure 5.16 Average Precision for BeeRank and PageRank

The Figure 5.17 shows that the mean average precision value for BeeRank is approximately 74 while PageRank has a value of 43.

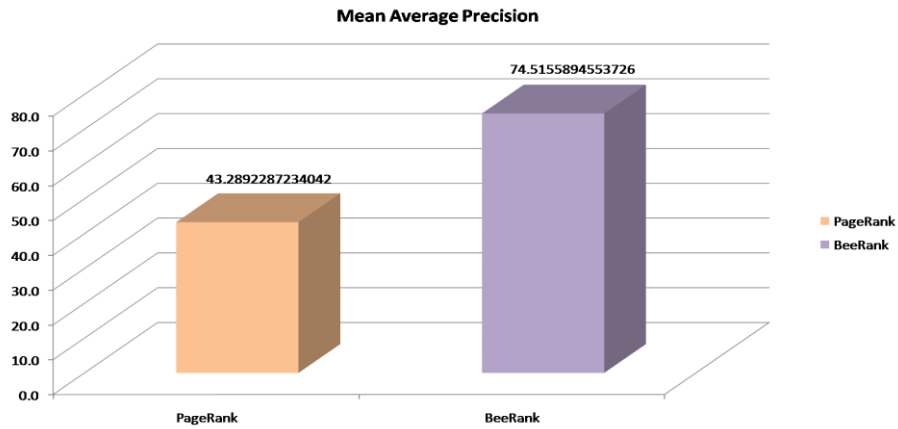


Figure 5.17 Mean Average Precision for BeeRank and PageRank

To access the ranking quality of the algorithm, Cumulative Gain, Discounted Cumulative Gain, Idealized Discounted Cumulative Gain, and Normalized Discounted Cumulative Gain is used which shows the performance improvement and is depicted in Figure 5.18, Figure 5.19, Figure 5.20 and Figure 5.21.

The Figure 5.18 shows that the cumulative gain value for BeeRank is approximately 424 while PageRank has a value of 262.

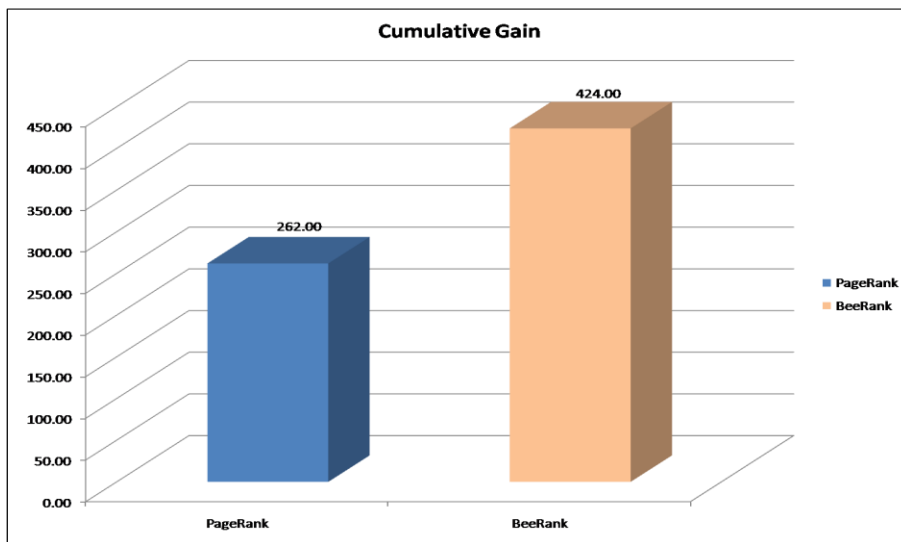


Figure 5.18 Cumulative Gain for BeeRank and PageRank

The Figure 5.19 shows that the discounted cumulative gain value for BeeRank is approximately 72 while PageRank has a value of 46.

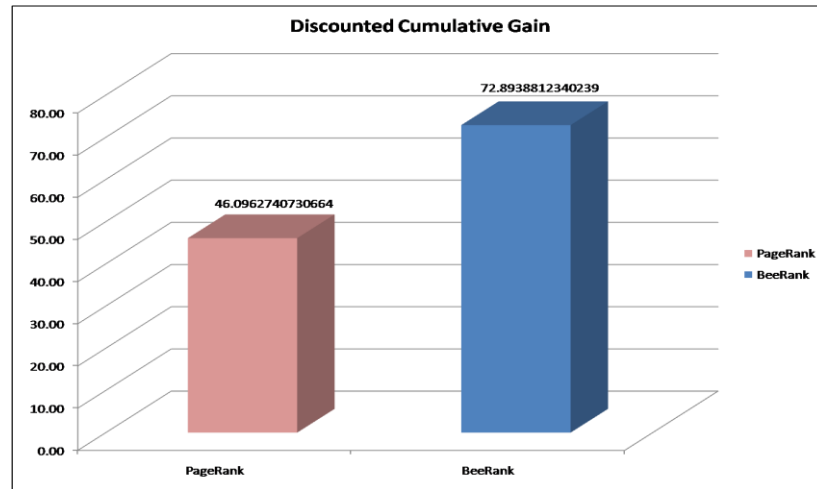


Figure 5.19 Discounted Cumulative Gain for BeeRank and PageRank

The Figure 5.20 shows that the idealized discounted cumulative gain value for BeeRank is approximately 79 while PageRank has a value of 55.

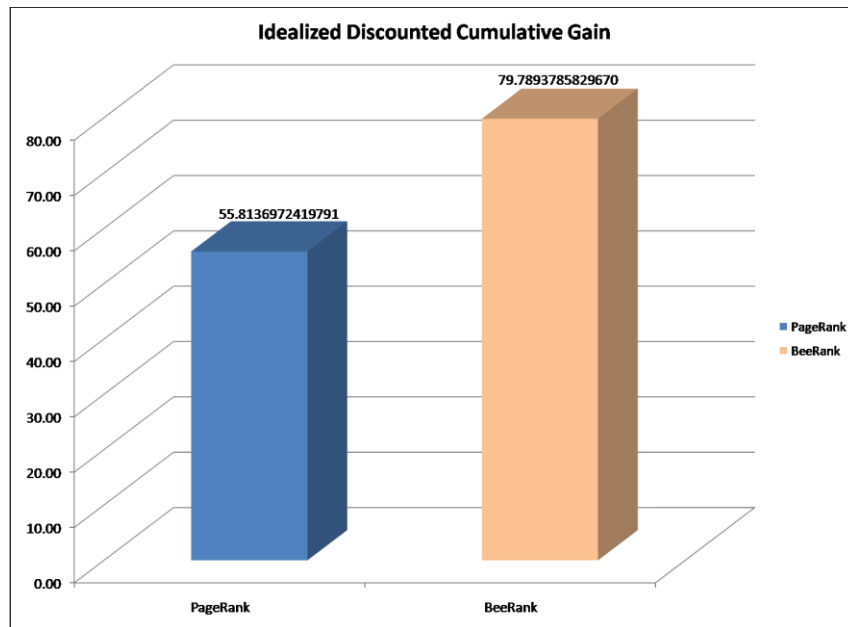


Figure 5.20 Idealized Discounted Cumulative Gain for BeeRank and PageRank

The Figure 5.21 shows that the normalized discounted cumulative gain value for BeeRank is approximately 0.91 while PageRank has a value of 0.82 that show an improvement in the ranking quality of the proposed algorithm.

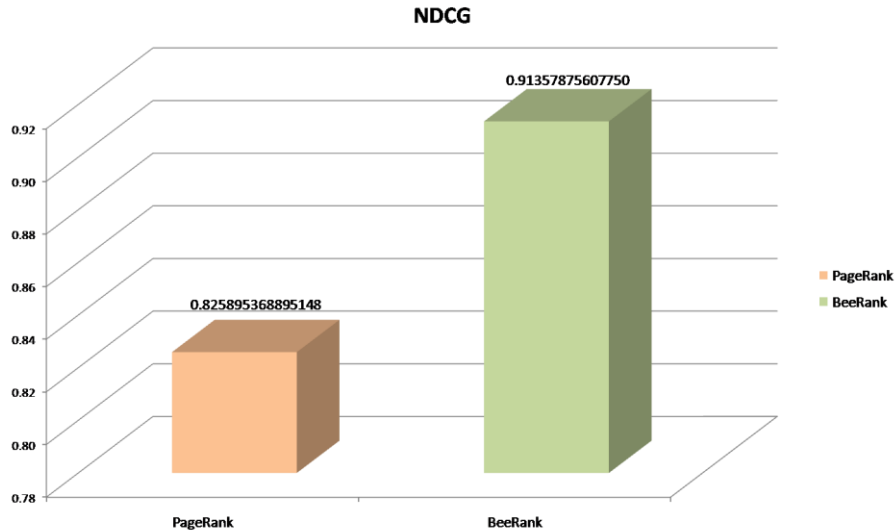


Figure 5.21 Representation of Normalized Discounted Cumulative Gain in BeeRank and PageRank

5.6 Summary

The proposed swarm intelligence based BeeRank approach is inspired by the Artificial Bee Colony algorithm for ranking the web pages. The given method tries to capture the best food source based on content and provide efficient and precise results by utilizing the structural approach of the web pages and also includes the content of the web pages.

The proposed approach estimates the fitness of the food source and based on the best food source the BeeRank is evaluated by overall minimizing the computational complexity which is considered for both process and time. The result depicts an improvement in the ranking process by using minimum time to achieve the ranking goal. By using both the approach high-quality, user-relevant information can be retrieved in a lesser time duration. The method of information retrieval is a complex task and much of the time is spent in retrieving relevant web pages, so by following this procedure relevant node can be obtained easily and irrelevant nodes are abandoned in the initial phase of processing thus improving the efficiency rate.

CHAPTER 6

PERFORMANCE EVALUATION AND RESULT ANALYSIS OF SCRANK, GRANK AND BEERANK

6.1 Introduction

The present era is submerged in data and it is quite difficult to manage the emergence of unstructured data for mining pertinent information as the information on the web is increasing continuously. Maximum time of the user is consumed in framing appropriate query and refining the resultant web pages. To represent information in proper format, the user has to face different problems. For ranking the desired result search engine has a crucial part in information filtering. It is still a dream to search for the accurate information and in this perspective the proposed work is performed which help to optimize the ranking algorithm for web page ranking. The work encapsulates a technique that employs both the content mining and structural mining which is analyzed using various performance evaluation measures. The filtering of the web pages at the initial stage helps to minimize the computational complexity of the algorithms and is illustrated in the proposed work. The results show a comparative analysis of scRank, gRank, and BeeRank with the PageRank Algorithm, and the obtained results show an improvement in results and therefore reduce the time in executing the web page ranking.

6.2 Performance Evaluation Measures

In Informational Retrieval System, user results based on query should be properly ranked so that relevant information can be obtained. Ranking of relevant web pages plays a vital role in the retrieval procedure. As such, various evaluation measures are acquired to access the performance evaluation. In the ranking process, the evaluation measure helps in assessing how the user query is fulfilled by the search result.

This work accesses the performance evaluation of the basic PageRank algorithm with different proposed algorithm scRank, gRank, and BeeRank. A comparative analysis is presented using various evaluation metrics which are quite useful in information retrieval and ranking process. The working of the algorithm is done on the dataset and the ranked pages are viewed based on user's queries and finally evaluated to judge its performance.

For retrieving information from the corpus of data and solving user-related queries, ranking plays a central role governing diverse applications which range from search engine to recommendation system, from machine learning to expert findings, it also deserves a role in tag suggestion for Tweets and Image label prediction. The key measures that are considered in a search engine are user happiness and there can be many factors that play a specific role like speed of response, relevance, rate of return of user response, etc [132]. It is one of the challenging roles to design a good ranking function that helps in the performance evaluation. It has been observed that there is no single ranking function that is optimal in all applications.

It is the need of the hour to evaluate the performance of the algorithm and it observed that a single measure is not enough for accessing the quality measure so various evaluation methods are used for judging the relevance of the algorithm. Precision, Recall, Average Precision (AP), Mean Average Precision (MAP), Cumulative Gain (CG), Discounted Cumulative Gain (DCG), Idealized Discounted Cumulative Gain (IDCG) and Normalized Discounted Cumulative Gain (NDCG) [133] are all used in accessing the ranking of the web pages. These measures help in evaluating the performance of a ranking function. [134], [135].

6.3 Performance Analysis of Proposed Algorithms

The performance of PageRank, scRank, gRank, and BeeRank can be illustrated based on different evaluation measures that help in accessing the ranking quality of the algorithm. All the proposed algorithms are tested on a dataset consisting of 8910 URLs having a total of 100 queries which are considered. It has been found by observing the result that the PageRank algorithm is unable to achieve the result till 25 iterations while other algorithms reach the desired result in much less iteration. The scRank needs 19 iterations, gRank needs 10 iterations, and BeeRank needs 4 iterations. After a limited number of iterations, we are getting the same value and thus the process can be terminated. The obtained results also reflect that by using other approaches computational complexity can be minimized that helps in saving the overall time and effort. Figure 6.1 illustrates the iteration required by using different approaches. It is observed that that scRank takes 19 iterations, gRank takes 10 iterations, and BeeRank takes 4 iterations while PageRank require 25 iterations to complete the processing.

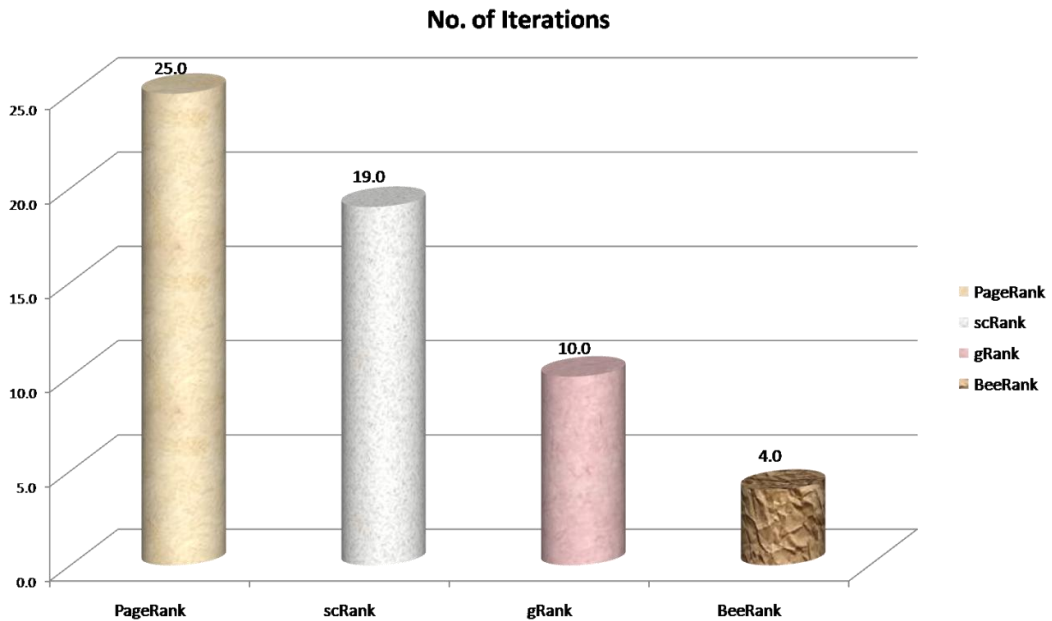


Figure 6.1 Number of iterations in PageRank, scRank, gRank and BeeRank

To access the ranking quality of different algorithm evaluation measures is analyzed and it has been observed that the BeeRank algorithm outperforms all the other algorithms like PageRank, scRank, and gRank. The number of iterations to reach the target goal is optimum for the BeeRank algorithm. The proposed work helps in saving the overall time and effort. The execution time of the running process is graphically represented in Figure 6.2(a) and Figure 6.2(b).

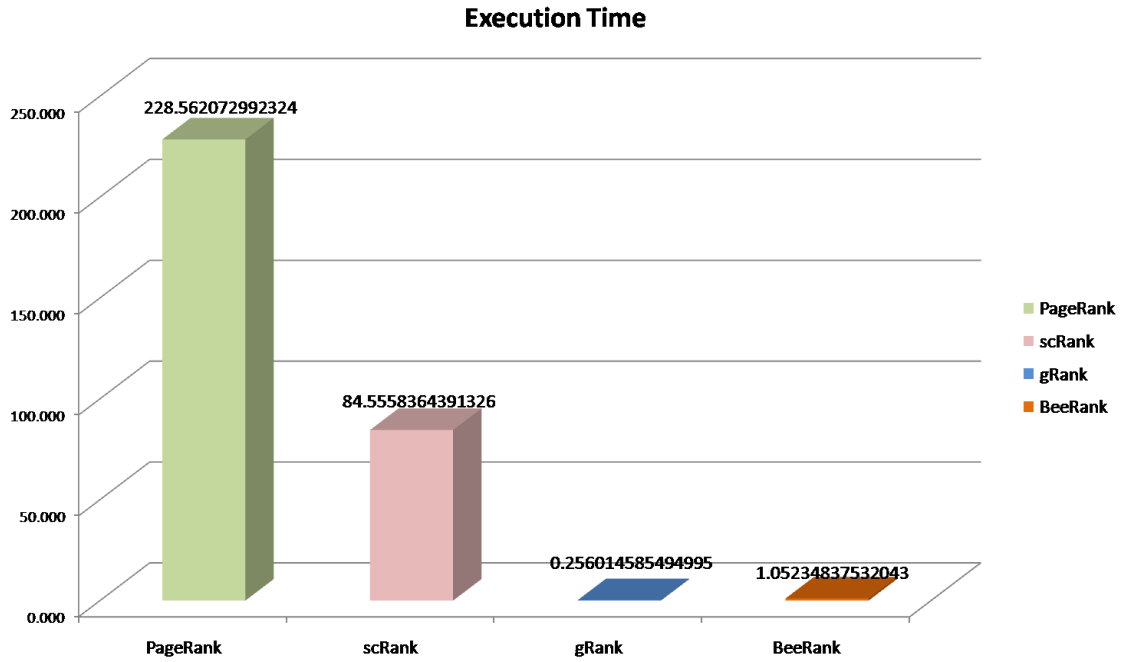


Figure 6.2 (a) Bar Graph of Execution Time for PageRank, scRank, gRank and BeeRank

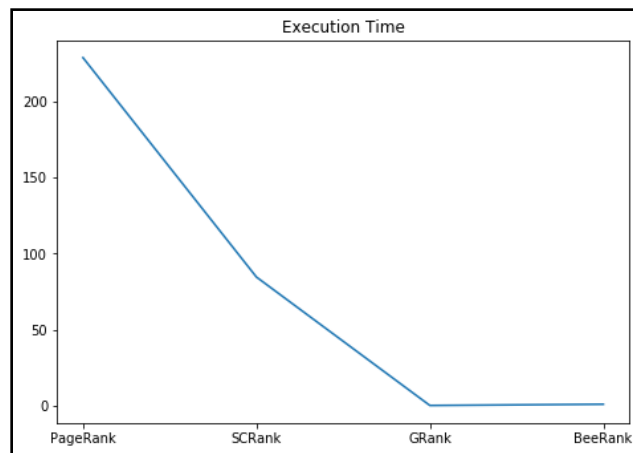


Figure 6.2 (b) Execution Time for PageRank, scRank, gRank and BeeRank

For evaluation measure Precision, Recall, Average Precision (AP), and Mean Average Precision (MAP) are evaluated and shown in Figure 6.3, Figure 6.4, Figure 6.5, and Figure 6.6.

The result depicts that the precision rate, recall, Average Precision (AP), and Mean Average Precision is increased in all the ranking algorithms specified, and the computational complexity is reduced overall as specified due to the filtering of web pages at the initial phase.

The precision value shown in Figure 6.3 for PageRank is approximately 40, scRank has 48, gRank has 49 and BeeRank has 53, which shows an overall performance enhancement.

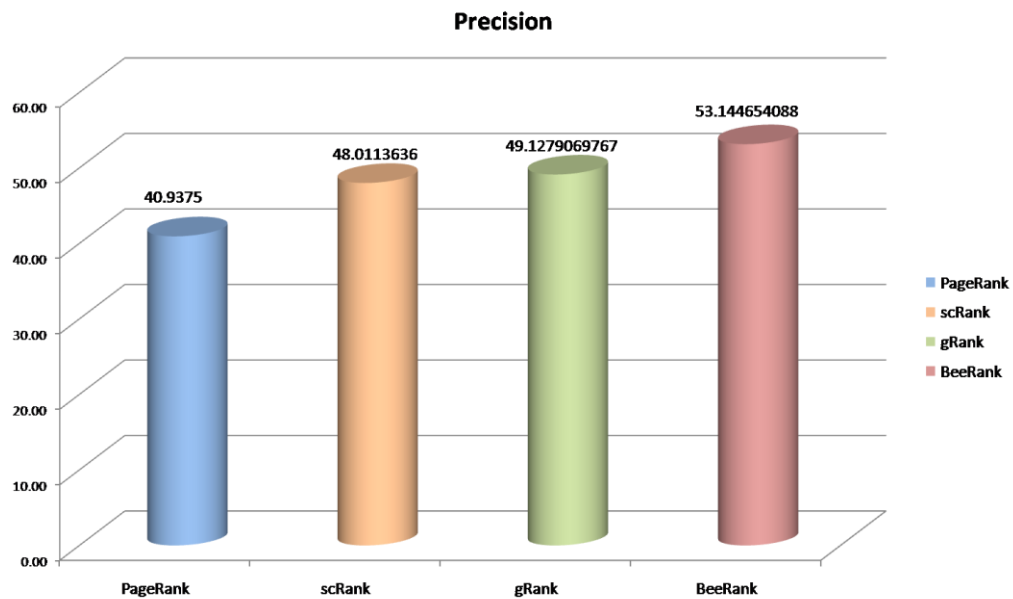


Figure 6.3 Precision values for PageRank, scRank, gRank and BeeRank

The recall value shown in Figure 6.4 for PageRank is approximately 46, scRank has 52, gRank has 60 and BeeRank has 86.

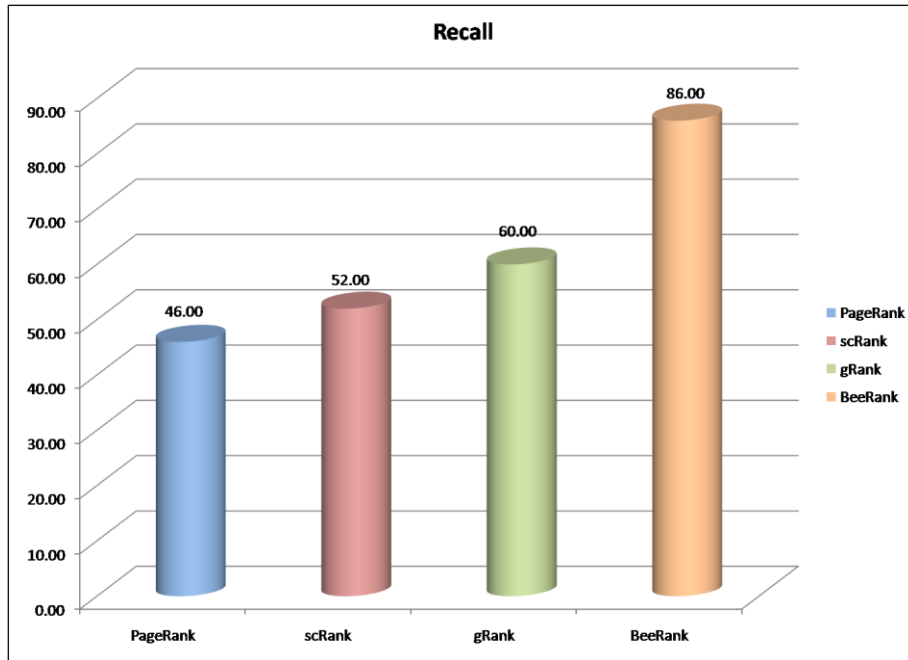


Figure 6.4 Recall value for PageRank, scRank, gRank and BeeRank

The average precision value shown in Figure 6.5 for PageRank is approximately 45, scRank has 67, gRank has 73 and BeeRank has 95 depicting continuous improvement.

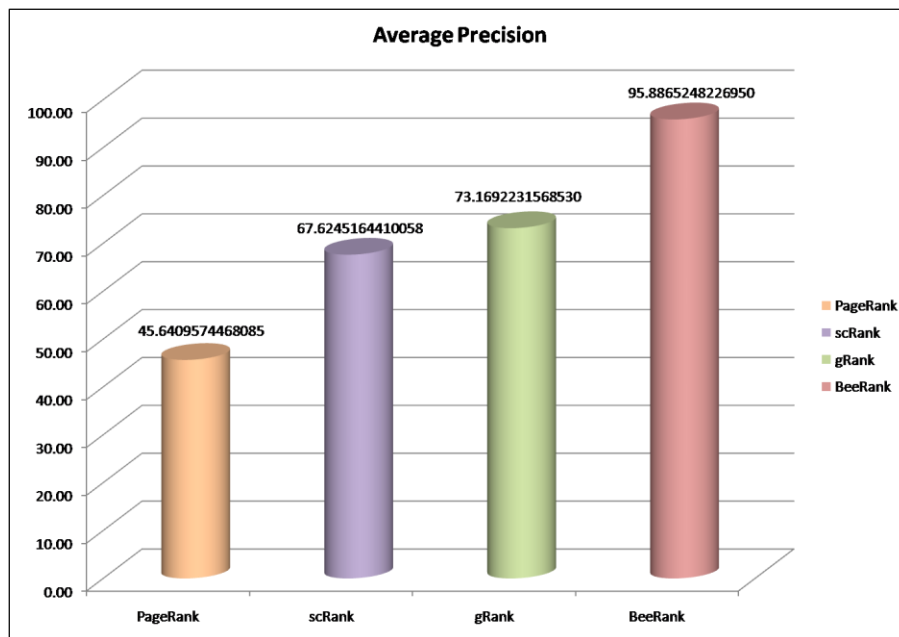


Figure 6.5 Average Precision for PageRank, scRank, gRank and BeeRank

The mean average precision value shown in Figure 6.6 for PageRank is approximately 43, scRank has 57, gRank has 61 and BeeRank has 74 showing enhancement in the algorithm performance.

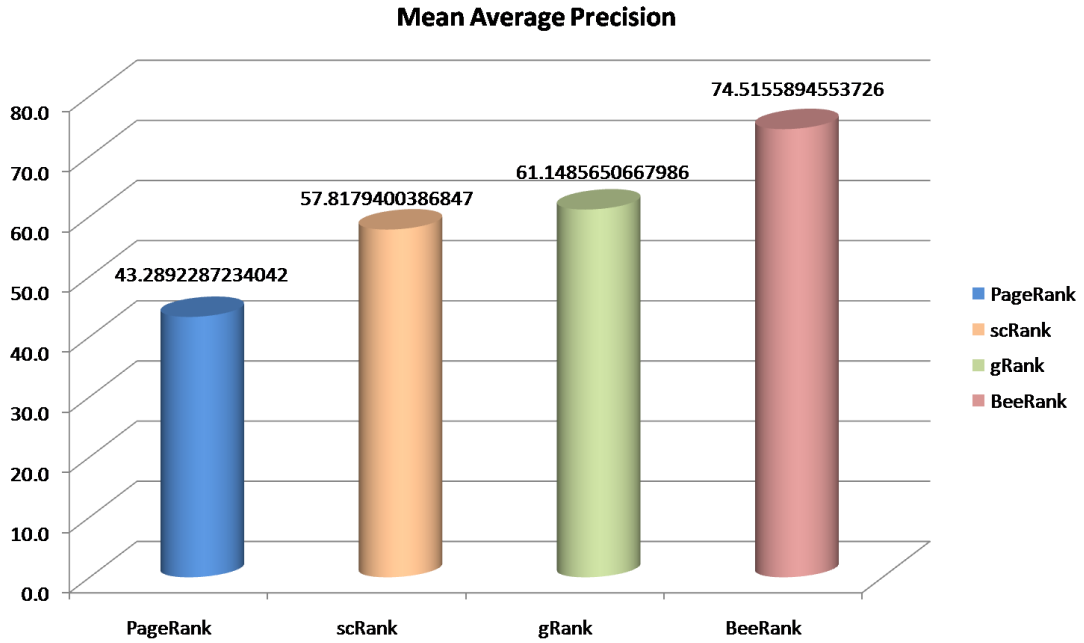


Figure 6.6 Mean Average Precision for PageRank, scRank, gRank and BeeRank

The ranking qualities of the algorithms are evaluated using Cumulative Gain (CG), Discounted Cumulative Gain (DCG), Idealized Discounted Cumulative Gain (IDCG), and Normalized Discounted Cumulative Gain (NDCG). Concerning the above analysis, it has been observed that the overall value of Cumulative Gain, Discounted Cumulative Gain, Idealized Cumulative Gain, and Normalized Discounted Cumulative Gain of BeeRank is high which depicts the efficiency of the given algorithm. The high value of NDCG shows the degree of relevancy of retrieved web pages and it helps in normalizing DCG by IDCG which is the best possible result based on best ranking measure. The results are depicted in Figure 6.7, Figure 6.8, Figure 6.9, and Figure 6.10.

The cumulative gain value shown in Figure 6.7 for PageRank is approximately 262, scRank has 331, gRank has 350 and BeeRank has 424.

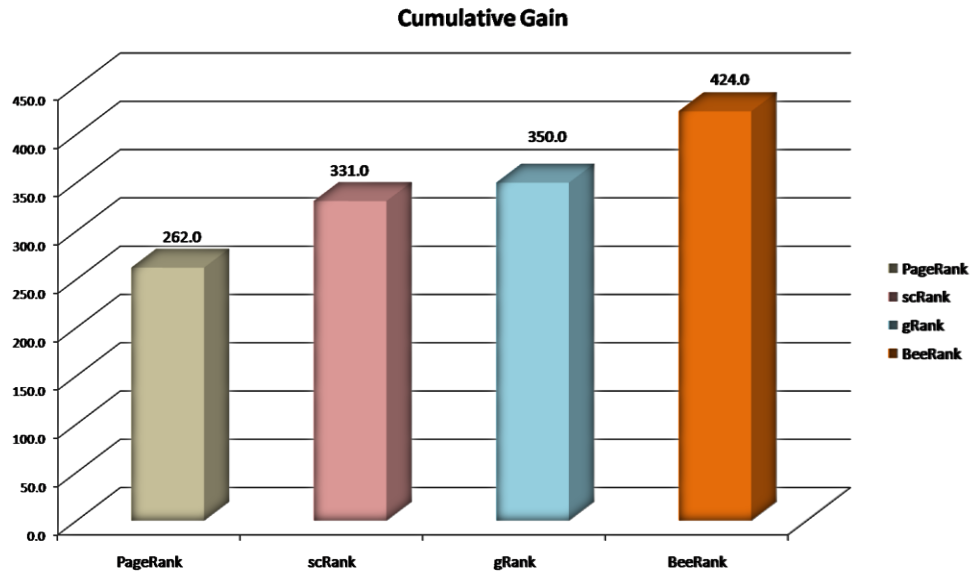


Figure 6.7 Cumulative Gain for PageRank, scRank, gRank and BeeRank

The discounted cumulative gain value shown in Figure 6.8 for PageRank is approximately 46, scRank has 59.5, gRank has 59.6 and BeeRank has 72 depicting continuous improvement.

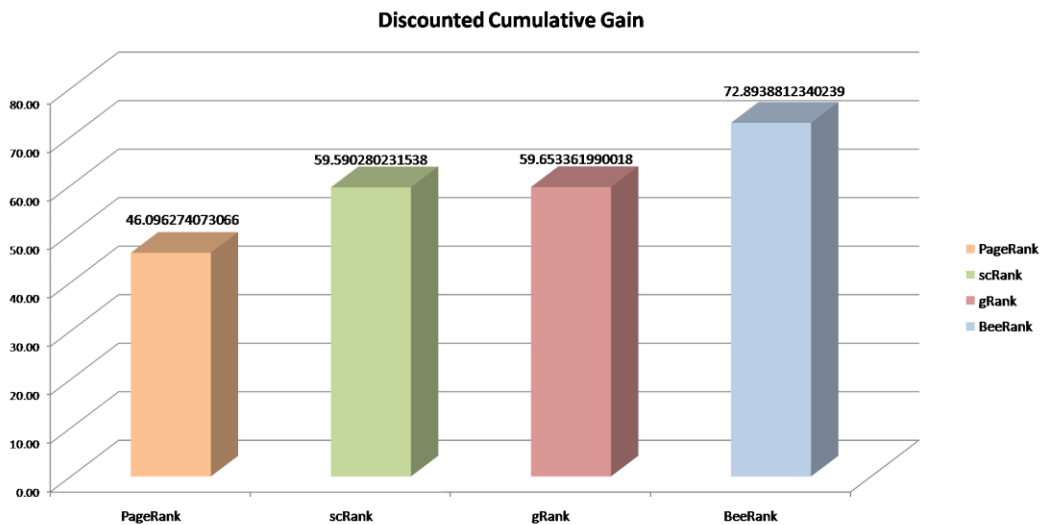


Figure 6.8 Discounted Cumulative Gain for PageRank, scRank, gRank and BeeRank

The idealized discounted cumulative gain value shown in Figure 6.9 for PageRank is approximately 55, scRank has 65, gRank has 68 and BeeRank has 79 depicting continuous improvement.

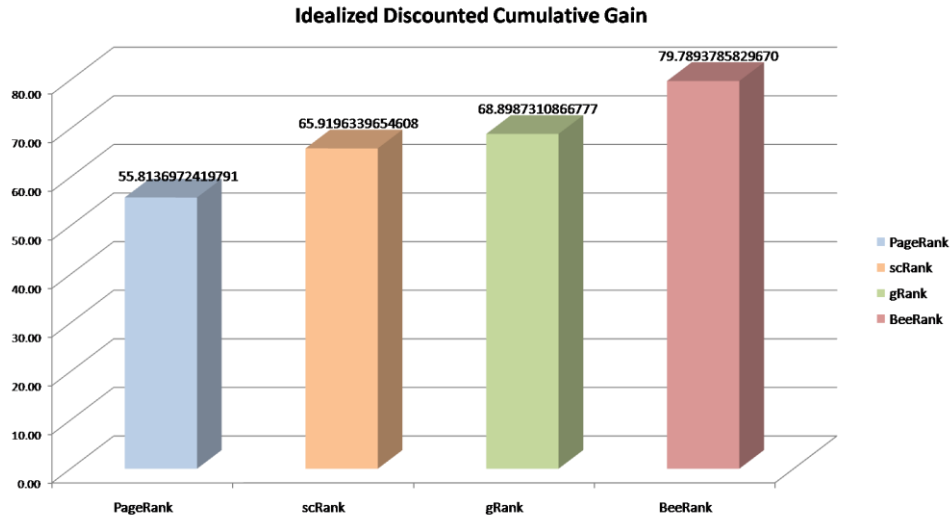


Figure 6.9 Idealized Discounted Cumulative Gain for PageRank, scRank, gRank and BeeRank

The normalized discounted cumulative gain value shown in Figure 6.10 for PageRank is approximately 0.82, scRank has 0.9, gRank has 0.86 and BeeRank has 0.91 that show an improvement in the ranking quality of the proposed algorithm.

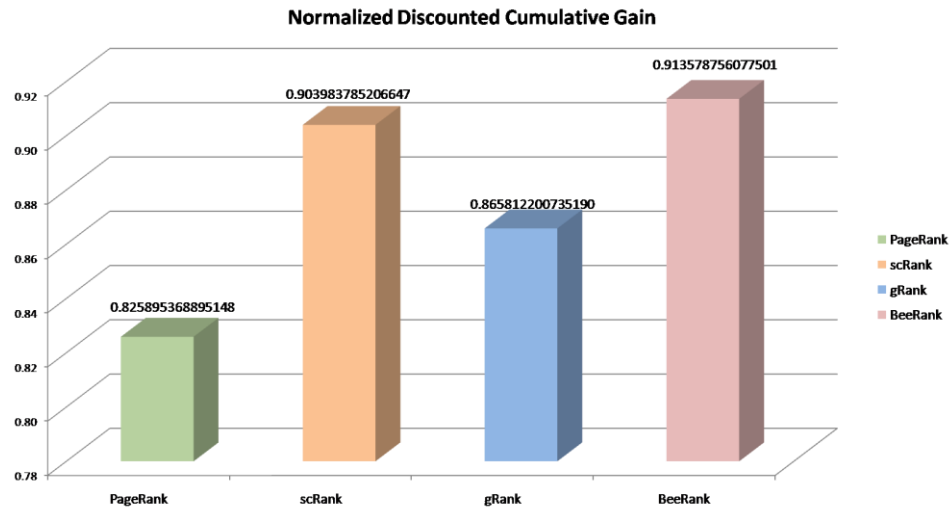


Figure 6.10 Normalized Discounted Cumulative Gain for PageRank, scRank, gRank and BeeRank

Figure 6.11 depicts the graphical analysis of the top 10 ranked URLs by using the proposed algorithms.

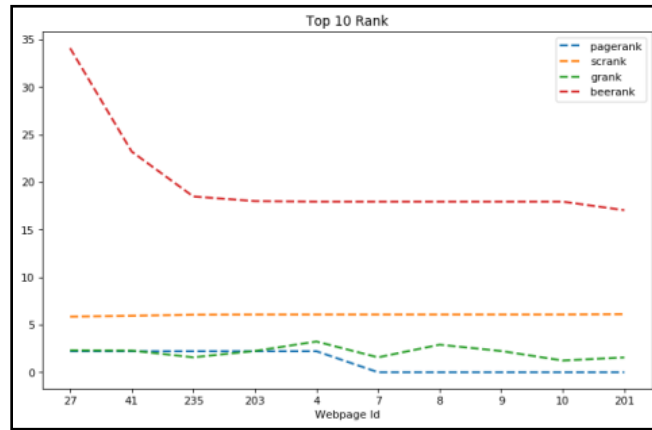


Figure 6.11 Graphically Analysis of PageRank, scRank, gRank and BeeRank

The Final Comparative Result Analysis for PageRank, scRank, gRank and BeeRank is summarized as-

Table 6.1 Comparative Result Analysis for PageRank, scRank, gRank and BeeRank

Parameter	PageRank	scRank	gRank	BeeRank
Number of Iteration	25	19	10	4
Execution Time	228.56207	84.555836	0.25601458	1.052348
Mean Average Precision	43.289	57.817	61.148	74.515
Normalized Discounted Cumulative Gain	0.82589	0.90398	0.8658	0.91357

It has been found by observing the result that the PageRank algorithm is unable to achieve the result till 25 iterations while other algorithms reach the desired result in much less iteration. The scRank needs 19 iterations giving 24% improvement in result, gRank needs 10 iterations providing 60% improvement to achieve the result and BeeRank needs 4 iterations giving 84% improvement. The mean average precision value for PageRank is approximately 43, scRank has value of 57, giving 33% better results,

gRank has 61 value providing 41% better results and BeeRank has 74 value which is providing 72% better results thus showing enhancement in the algorithm performance.

The normalized discounted cumulative gain (NDCG) value for PageRank is approximately 0.82, scRank has a value of 0.9, an improvement of 9%, gRank has 0.86 value giving an improvement of 4% and BeeRank has 0.91 value providing an improvement of 10% in the ranking quality of the proposed algorithm. The high value of NDCG shows the degree of relevancy of retrieved web pages.

The proposed works are compared with PageRank algorithm, as it is considered as the base of ranking algorithm and most of the algorithms are based on this and it is still being used by Google by adding various parameters to enhance its efficiency. The computational being involved in the PageRank algorithm is enormous, so by using different approach in the proposed work I tried to minimize the complexity of the procedure so that the results can be obtained in minimum time duration.

6.4 Summary

The proposed work in the dissertation provides ranking for the web pages by encapsulating both the structural links among the nodes and content within the web pages thus providing high precision value. By considering the proposed approach we are getting high-quality results and the user related information can be obtained in minimal time duration. Using the above-specified algorithms we are clustering the web pages at the initial step of processing by applying different conditions which help in reduced computational effort.

The procedure of Information Retrieval is a complex task that takes time in retrieving relevant web pages, so by following this procedure only relevant nodes are considered and we exclude irrelevant nodes that lack content thus improving the efficiency of the process. The result depicts that a node is not considered for ranking if it lacks proper content even though the hyperlink count for that node is more. The evaluation measure of the given approach depicts a rise in the precision value and an improved normalized value is obtained while using the proposed algorithms. The projected work helps in

overall minimizing the computational complexity of the ranking procedure by reducing the number of iterations and increasing the relevancy of the approach. The result shows an improvement in ranking method by using minimum time to achieve the ranking goal.

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

This chapter sums up the entire work and highlight future work in this direction.

There is an incredible change in the World Wide Web and the users face difficulty in accessing the needed information as per their need. Internet normally comprises of an abundance of information and people explore their areas to get the relevant information. It is found that the core component of any search engine is the ranking framework that helps in ranking the web pages solely based on user queries. Different algorithms are devised at each step of the information retrieval process, and it is observed that ranking is one of the core ingredients of any search engine that has a crucial part in arranging the information. A good ranking system should not be based on single criteria but can take input from multiple points to create a comprehensive ranking system. In this regard, different actions are adopted for web pages ranking that uses content, structure, or log data. Nature-inspired evolutionary measures like Genetic algorithm and Artificial Bee Colony are used to optimize the retrieved information that efficiently handle the complex environment of the web and help in optimizing the ranking algorithm for efficient search.

This thesis aims to develop an efficient page ranking algorithm for improving the search process and help the user to facilitate effective content based on their query. The proposed work provides web page ranking by encapsulating both structural links of the nodes and content within the web pages, thus providing high precision value. By considering the proposed approach we are getting high-quality results and the user related information can be obtained in minimal time duration. Using the above-specified algorithms we are clustering the web pages at the initial step of processing by applying different conditions which help in reduced computational effort.

7.1 Conclusion

The work developed in the thesis can be summarized into three categories-

1. scRank algorithm is proposed that efficiently rank web pages. The web pages are clustered based on user queries that undergo certain conditions, and thereafter ranking is applied to them. The experiment conducted demonstrates the working of the ranking model and assists in minimizing the complexity of the computation and thereby reduces the number of iterations which help to rank documents in least time duration.
2. gRank algorithm is proposed that rank the result of the query using a genetic algorithm. Web pages are clustered together on applying certain conditions and selection is performed which follows crossover and mutation. The working procedure of gRank is illustrated properly in the chapter that assists in optimizing the retrieved information by following a limited set of iterations in minimal time duration.
3. BeeRank algorithm inspired by the Artificial Bee Colony algorithm is proposed for ranking web pages and uses both the structural and content approach for calculating the rank value and provides better results. The user query extracts random web pages and abandoned the web pages having a single query term that is then evaluated to calculate the fitness value and thereby probability is calculated. The concentration of the food source is updated and finally updated fitness and BeeRank is calculated. The working is illustrated and the result shows an improvement in ranking and provides user-specific results.

7.2 Future Scope

Research work illustrated in this thesis can be outstretched in many directions that will help in enhancing the results thus obtained. The algorithms help in ameliorating the ranking of the web pages that assists the user in getting the related and important pages easily. In the proposed research work, a hybrid approach is given by using the structural and content link for ranking web pages and does not employ the usage mining approach that considers log files. If log files can also be include with this work then more accurate results can be achieved.

In gRank and BeeRank, nature-inspired algorithms are used that help in escalating the effectiveness of the ranking process. In the future, the algorithms likely enhanced to encapsulate a similar nature-inspired optimization algorithm that can be used to check ranking process efficiency. Firefly algorithm likely be employed to attract web pages that are rich in content, thus improving the ranking of the web pages. Machine learning methods can be amalgamated with nature-inspired algorithms that will help in providing user-oriented results.

Further, the research work may be extended and analyzed with categorical datasets. This can be extended by evaluation criterion measures for finding the relevant features and for refining the accuracy of prediction.

This is to summarize the final outline of my research work. It also mentions the scope of my research and the areas in need of further research.

REFERENCES

- [1] Croft, W. B. (1993), Chapter 1-Combining Approaches to Information Retrieval, *Advances in Information Retrieval*, Springer.
- [2] Irfan, S., Ghosh, S., (2019), "Ranking Web Pages Using Cosine Similarity Measure", International Conference on Computing, Power and Communication Technologies (GUCON), IEEE.
- [3] Kosala, R., and Blockeel, H., (2000), "Web Mining Research: A Survey", Volume 2, Issue 1, SIGKDD, ACM.
- [4] Roshdi, A., and Roohparvar. A., (2015), "Review: Information Retrieval Techniques and Applications", International Journal of Computer Networks and Communications Security, Vol. 3, NO. 9.
- [5] Irfan, S., and Babu, B.V., (2016), "Information Retrieval in Big Data Using Evolutionary Computation: A Survey", International Conference on Computing, Communication and Automation (ICCCA), IEEE.
- [6] Witten, I.H. and Frank, E., (2000), "*Data Mining, Practical Machine Learning Tools and Techniques*", Morgan Kaufmann, Elsevier.
- [7] Chakrabarti, S., (2002), "*Mining the Web: Discovering Knowledge from Hypertext Data*", Morgan Kaufmann, 2002.
- [8] Hiemstra, D., (2009), "Towards an Information Retrieval Theory of Everything, Information Retrieval Searching in the 21st century", John Wiley and Sons.
- [9] Sharma, P.S., Yadav, D., Garg, P., (Jan 2020), "A systematic review on page ranking algorithms", Int. j. inf. tecnol. , Springer.
- [10] Chand, D., (2015), "Study & Analysis of Web Informational Retrieval", PhD Thesis, Savitribai Phule Pune University, Maharashtra.
- [11] Manning, D.C., Raghavan, P. and Schütze, H., (2009), "*An Introduction to Information Retrieval*", Cambridge University Press.
- [12] Vishwakarma, S. K., (2014), "Improving Performance of Information Retrieval System Based on Static Inverted Pruning & Rank Model ", PhD Thesis, Sir Padampat Singhania University.

- [13] Hiemstra, D., (2009), "Information Retrieval Models, Information Retrieval: Searching in the 21st Century", John Wiley and Sons.
- [14] Kobayashi, M., and Takeda, K., (2000), "Information Retrieval on the Web", ACM Computing Surveys, Vol. 32, No. 2.
- [15] Acid, S., Campos, L.M., Fernandez-Luna, J.M., Huete, J.F., (2003), "An Information Retrieval Model Based on Simple Bayesian Networks", International Journal of Intelligent Systems, Vol. 18, 251–265, Wiley Periodicals.
- [16] Huibers, T.W.C., Lalmas, M., and Rijsbergen, V. (1997), "Information Retrieval and Situation Theory", ACM SIGIR Forum, 30(1), 11–25.
- [17] Mashagba, E.A., Mashagba, F.A., and Nassar, M.O., (2011), "Query Optimization Using Genetic Algorithms in the Vector Space Model", IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 5, No 3.
- [18] Salton, G., Wong, A., and Yang, C.S., (1975), "A Vector Space Model for Automatic Indexing", ACM.
- [19] https://en.wikipedia.org/wiki/Evolutionary_computation.
- [20] Kumar, S., "Chapter 1- Web Mining and Information Retrieval", A study of web mining tools for query optimization, PhD Thesis, University of Jammu, 2015.
- [21] Roberston, S.E., (1977), "The probability Ranking Principle in IR", Journal of Documentation, Vol 33, N0.4.
- [22] Cooper, G.E., and Herskovits, E.,(1992),"A Bayesian Method for the Induction of Probabilistic Networks from Data", Machine Learning, 9, 309-347 ,2 Kluwer Academic Publishers.
- [23] Campos, L. M., Fernandez-Luna, J.M., Huete, J.F., (2004), "Bayesian networks and information retrieval: an introduction to the special issue", Information Processing and Management, Elsevier.
- [24] Zhai, C. and Lafferty, J.(2004), "A Study of Smoothing Methods for Language Models Applied to Ad Hoc Information Retrieval", ACM.
- [25] Cordon, O., Herrera-Viedma, E., Lopez-Pujalte, C., Luque, M., and Zarco, C., (2003), "A Review on the Application of Evolutionary Computation to Information Retrieval", International Journal of Approximate Reasoning, Elsevier.

- [26] Irfan, S., & Ghosh, S., (2017), "Optimization of Information Retrieval Using Evolutionary Computation: A Survey", International Conference on Computing, Communication and Automation, IEEE.
- [27] Li, Y., Zhong, N., (2004), "Web Mining Model and Its Applications for Information Gathering", Knowledge-Based Systems, Elsevier.
- [28] Lewandowski, D. (2005), "Web Searching, search engines and Information Retrieval", Information Services & Use, IOS Press.
- [29] Rajan, A.P., (2014), "Design of Analysis of Web Mining Algorithm", PhD Thesis, Manomaniam Sundaranar University, Tirunelveli, Tamil Nadu.
- [30] Jyoti, (2011), "A Framework for extracting relevant web pages from www using web mining", PhD Thesis, Jyoti Maharshi Dayand University.
- [31] Yates, R.B., (2003), "Information retrieval in the Web: Beyond Current Search Engines", International Journal of Approximate Reasoning, 97–104, Elsevier, (2003).
- [32] Agbele, K.K. , Ayetiran, E.F. , Aruleba, K.D. and Ekong, D.O., "Algorithm for Information Retrieval Optimization", International Journal of Computer, Electrical, Automation, Control & Info. Engineering, 2016.
- [33] Cooley, R., Mobasher, B., Srivastava, J., (1997), "Web Mining: Information and Pattern Discovery on the World Wide Web", ICTAI, pp. 558–567.
- [34] Irfan, S., and Ghosh, S., (2018), "Web Mining for Information Retrieval", International Journal of Engineering Science and Computing, Volume 8, Issue No.4, 2018.
- [35] Vasavi, G., and Sudha, T., (2017), "Information Extraction from Online Shopping Sites using Web Content Mining Methods and Techniques", International Conference on Innovative Applications in Engineering and Information Technology (ICIAEIT).
- [36] Kumar, A., and Singh, R.A., (2017), "A Study on Web Structure Mining", International Research Journal of Engineering and Technology (IRJET), Volume: 04 Issue 1.
- [37] Irfan, S., Ghosh, S., and Dwivedi, G., (2017), "Optimization of K-Means clustering Using Genetic Algorithm", IEEE.
- [38] Irfan, S. and Ghosh, S., (2019), "Efficient Ranking Framework for Information Retrieval Using Similarity Measure", Computational Vision and Bio-Inspired Computing, ICCVBIC, Springer.
- [39] Neelima, G., and Rodda, S., (2015), "An Overview on Web Usage Mining ", Advances in Intelligent Systems and Computing, Springer.

- [40] Irfan, S., and Ghosh, S., (2018), “Web Usage Mining in Ranking Web Pages”, International Journal of Innovative Research in Science, Engineering and Technology, Vol. 7, Issue 10, October 2018.
- [41] Hirsch, L. and Nuovo, A.D., (2017), “Document Clustering with Evolved Search Queries”, IEEE.
- [42] Runkler, T.A., and Bezdek, J.C., (2003), “Web Mining with Relational Clustering”, Elsevier.
- [43] Rafi, M., Shaikh, M.S., and Farooq, A., (2010), “Document Clustering based on Topic Maps”, International Journal of Computer Applications.
- [44] Shah, N., and Mahajan, S., (2012), “Document Clustering: A Detailed Review”, International Journal of Applied Information System.
- [45] Kumar, V., (2015), “Retrieval of Ranking of pages in Semantic Web”, PhD Thesis, GS, St.Peter’s Institute of Higher Education of Research, St.Peter’s University, Chennai.
- [46] Lewandowski, D., (2008), “Search engine User behavior: how user be guided to quality content?”, Information Services and Use.
- [47] Jansen, J., Spink, A., and Saracevic, T., (2000), “Real life, Real users, and Real needs: A Study and Analysis of User Queries on the web”, Information Processing & Management.
- [48] Hian, D.K. (2017), “Learning to Rank for Retrieval and Recommendation”, PhD Thesis, University of California, San Diego.
- [49] Hjørland, B., (2010), “The Foundation of the Concept of Relevance”, Journal of the American Society for Information Science and Technology.
- [50] Grady, C., and Lease, M.,(2010), “Crowd sourcing Document Relevance Assessment with Mechanical Turk”, In proceedings of the NAACL with 2010 workshop on Creating Speech and Language Data with Amazon Mechanical Turk, Los Angelis, California.
- [51] Brin, S. & Page, L., (1998), “The anatomy of a Large-scale Hypertextual Web Search Engine”, In Proceedings of the Seventh International World Wide Web Conference.
- [52] Sharma, D.K. and Sharma, A.K., (2010), "A Comparative Analysis of Web Page Ranking Algorithms", International Journal on Computer Science and Engineering.
- [53] Langville, A.M., Meyer, C.D., (2006), “Information Retrieval and Web Search”, Handbook of Linear Algebra.
- [54] Xing, W. and Ghorbani, A. (2004), “Weighted PageRank Algorithm”, Proceedings of the Second Annual Conference on Communication Networks and Services Research, IEEE.

- [55] Kleinberg, J.M., (1999), "Authoritative Sources in a Hyperlinked Environment", ACM, 46(5):604–632.
- [56] Sanjay and Kumar, D., (2015), "A Review Paper on Page Ranking Algorithms", International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume 4 Issue 6.
- [57] Fatima, R.S., M.K., (2017), "Analysis of Different Page Ranking Algorithms", International Journal of Technical Research and Applications, Volume 5, Issue 2 , PP. 81-89.
- [58] Yates, R.B. and Davis, E. (2004), "Web Page Ranking using Link Attributes", May 17–22, ACM.
- [59] Fujimura, K., Inoue, T. and Sugisaki, M., (2005), "The EigenRumor Algorithm for Ranking Blogs", WWW.
- [60] Bidoki, M. Z., Yazdeni,N., (2008), "DistanceRank: An intelligent algorithm for web pages", Information Processing & Management, Elsevier.
- [61] Jiang, H., Ge , Y.X., Zuo ,D., Hen,B., (2008), "TimeRank: A method of improving ranking scores by visited time", Proceeding of seventh International Conference on machine leaning and cybernetics, Kunming, IEEE.
- [62] Jie, S., Chen, C., Zhanghui, Rong-Shuang, S., ZhuYan and Hekun,(2008), "TagRank: A New Rank Algorithm for Webpage based on Social Web", International Conference on Computer Science and Information Technology.
- [63] Lamberti, F., Sanna , A., and Demartini , C., (2009), " A Relation-Based Page Rank Algorithm for Semantic Web Search Engines", IEEE Transactions on knowledge and Data Engineering, Vol.21, No.1.
- [64] Lee, L.W. , Jiang ,J.Y., Wu , C., Lee , S.J., (2009), "A Query-Dependent Ranking Approach for Search Engines", Second International Workshop on Computer Science and Engineering.
- [65] Vojnovic, M., Cruise, J., Gunawardena, D., and Marbach, P.,(2009), "Ranking and Suggesting Popular Items", IEEE Transactions on knowledge and data Engineering, Vol. 21, No.8.
- [66] Bhamidipati , N.L. and Pal ,S.K., (2009), "Comparing Scores Intended for Ranking", IEEE Transactions on Knowledge and Data Engineering, Vol. 21, No. 1.
- [67] Irfan, S., and Ghosh, S., (2018), "A Review on Different Ranking Algorithms", ICACCCN, IEEE.

- [68] Lian, X., Chen, L., (2010), "Ranked query Processing in Uncertain Databases", IEEE Transactions on Knowledge and Data Engineering, Vol. 22, No. 3.
- [69] Irfan, S., Ghosh, S., (2018), "Analysis & Challenges of Web Ranking Algorithms", 4th International Conference on Computing Communication and Automation, IEEE.
- [70] Katerenchuk, D., and Rosenberg, A., (2016), "RankDCG: Rank-Ordering Evaluation Measure".
- [71] Wang, Y. ,Wang, L. ,Li, Y., He, D. ,Chen, W., and Liu ,T.Y.,(2013), "A Theoretical Analysis of NDCG Ranking Measures", JMLR: Workshop and Conference Proceedings Volume ,1–30
- [72] <https://opensourceconnections.com/blog/2018/02/26/ndcg-scorer-in-quepid/>
- [73] A. Astolfi, (2002), "OPTIMIZATION An Introduction".
- [74] Lim, S.M. and Leong, K.Y., (2018), "A Brief Survey on Intelligent Swarm-Based Algorithms for Solving Optimization Problems", Nature-inspired Methods for Stochastic, Robust and Dynamic Optimization.
- [75] Jr Fister, I., Yang, X., Fister, I., Brest, J. and Fister, D., (2013), "A Brief Review of Nature-Inspired Algorithms for Optimization".
- [76] Binitha, S., Sathya, S.S, (2012),"A Survey of Bio inspired Optimization Algorithms", International Journal of Soft Computing and Engineering (IJSCE), Volume-2, Issue-2.
- [77] Bollegala, D., Noman, N. and Iba, H., (2011), "RankDE: Learning a Ranking Function for Information Retrieval using Differential Evolution", GECCO, ACM.
- [78] Ksibi, A., Ammar, B.A. and Amar, B.C., (2014) "Enhancing relevance re-ranking using nature-inspired meta-heuristic optimization algorithms", IEEE Congress on Evolutionary Computation (CEC).
- [79] Mirjalili, S., Dong, S.J. and Lewis, A. , (2019), " Ant Colony Optimizer: Theory, Literature Review, and Application in AUV Path Planning", Nature-Inspired Optimizers, Studies in Computational Intelligence, Springer.
- [80] Kumar, A., (2015), "Optimization of Artificial Bee Colony Algorithm for Clustering in Data Mining", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 5, Issue 7.
- [81] Spears, W.M., Jong, K.A.D, Back, T., Fogel, D.B., and Garis, H.D., (1993), "An Overview of Evolutionary Computation", European Conference on Machine Learning, Springer.

- [82] Simon, P., (2009), "Two Stage Approach to Document Retrieval using Genetic Algorithm", International Journal of Recent Trends in Engineering, Vol. 1, No. 1.
- [83] Beasley, D., Bull, D.R., Martin, R.R., (1993), "An Overview of Genetic Algorithms: Part 1, Fundamentals".
- [84] Oyelade, J., Isewon, I., Oladipupo, F., Aromolaran, O., Uwoghien, E., Ameh, F., Achas, M., and Adebisi, E., (2016), "Clustering Algorithms: Their Application to Gene Expression Data", Bioinformatics and Biology Insights.
- [85] Holland, J.H., (1992), "Genetic Algorithms", Scientific American, 267(1), pp 66-72.
- [86] Hruschka, E.R., Campello, R.J.G.B, Freitas, A.A., and Carvalho, A.C.P.L.F.de, (2009), "A Survey of evolutionary algorithm for clustering" IEEE Transactions on Systems, Man & Cybernetics, Part C: Application & Reviews, 39(2), pp 133-135.
- [87] Karaboga, D. and Akay, B., (2009), "A comparative study of Artificial Bee Colony algorithm", Applied Mathematics and Computation, Elsevier.
- [88] Bansal, C.J., Sharma, H. and Jadon, S.S., (2013), "Artificial bee colony algorithm: a survey". Int. J. Advanced Intelligence Paradigms, (Vol. 5, Nos. 1/2).
- [89] Shen, C., Kim, J., Wang, L. and Anton Van Den Hengel, A.V.D, (2012), "A positive semi definite metric learning using boosting-like algorithms", Journal of Machine Learning Research.
- [90] Anjusha, I.T., and Nizar, M.A., (2019), "Combining Hyperlink Structure and Content of Webpage for Personalization of Search Engine", Information and Communication Technology for Intelligent Systems, Smart Innovation, Systems and Technologies, Springer.
- [91] Koo, J., Chae, D.K., Kim, D.J. and Kim, S.W., (2019), "Incremental C-Rank: An effective and efficient ranking algorithm for dynamic Web environments", Knowledge-Based Systems, Elsevier.
- [92] Gao, Y. and Xu, K., (2018), "pRankAggreg: A fast clustering based partial rank aggregation", Information Sciences, Elsevier.
- [93] Choudhary, J., Tomar, D.S., and Singh, D.S., (2018), "An Efficient Hybrid User Profile Based Web Search Personalization through Semantic Crawler", Springer.
- [94] Goyal, S., Kumar, R., Kumar, M. and Chopra, V., (2019), "An efficient page ranking approach based on vector norms using sNorm(p) algorithm", Information Processing and Management, Elsevier.

- [95] Sethi, S., and Dixit, A. (2019), "A Novel Page Ranking Mechanism Based on User Browsing Patterns", Software Engineering, Advances in Intelligent Systems and Computing, Springer.
- [96] Sheetal K. Sahu, Rajendra Pal Gupta and Amit Dutta "An Analysis of Web User Behavior using Hybrid Algorithm based on Sequential Pattern Mining", International Journal of Applied Engineering Research ISSN 0973-4562 Volume 14, Number 10, 2019.
- [97] Guo, J., Fan, Y., Pang, L., Yang, L., Ai, Q., Zamani, H., Wua, C., Croft, W.B. and Cheng, X. "A Deep Look into Neural Ranking Models for Information Retrieval", Journal of Information Processing and Management, 2019.
- [98] Overland, I., and Juraev, J., (2019), "Algorithm for Producing Rankings Based on Expert Surveys", MDPI, Algorithms.
- [99] Rathod, S.D., Shah, A., and Solanki, N.D., (2017), "Personalize Page Ranking Using Genetic Algorithm and Dynamic Profile", IJARIE, Volume 3 Issue 3.
- [100] Yates, R.B., Cuzzocrea, A., Crea, D., Bianco, G.L., (2019), "An Effective and Efficient Algorithm for Ranking Web Documents via Genetic Programming", In Proceedings of 34th ACM/SIGAPP Symposium on Applied Computing (SAC'19).
- [101] Chen, L.C., (2018), "A novel page clipping search engine based on page discussion topics", Springer.
- [102] Yan, L., Gui, Z., Dub, W., Guo, Q., (2011), "An Improved PageRank Method based on Genetic Algorithm for Web Search", Advanced in Control Engineering and Information Science, Elsevier.
- [103] Mueller, J., (2017), "Combining Aspects of Genetic Algorithms with Weighted Recommender Hybridization", In Proceedings of The 19th International Conference on Information Integration and Web-based Applications & Services, ACM.
- [104] Subramanian, S., Sivaprakasam, (2014), "Efficient Algorithm for Removing Duplicate Documents", International Journal of Soft Computing and Engineering (IJSCE).
- [105] Thada, V., Jaglan, V., (2014), "Design of Web Ranking Module using Genetic Algorithm", International Journal of Computer Applications.
- [106] Singh, S.P., and Solanki, S., (2019), "Recommender System Survey: Clustering to Nature Inspired Algorithm", Proceedings of 2nd International Conference on Communication, Computing and Networking, Lecture Notes in Networks and Systems, Springer.
- [107] Semenikhin, S.V., Denisova, L.A., (2016), "Learning To Rank Based on Modified Genetic Algorithm".

- [108] Radwan,A.A.A., Latef, B.A.A., Ali, A.M.A. and Sadak, O.A., (2008), “Using Genetic Algorithm to Improve Information Retrieval Systems”, International Scholarly & Scientifics Research & Innovation.
- [109] Yeh, J.Y., Lin, J.Y., Ke, H.R., Yang, W.P.,(2007), “ Learning to Rank for Informational Retrieval Using Genetic Programming”, ACM.
- [110] Nuovo, D., A. G and Catania, V.,(2008),” An Evolutionary fuzzy c-means approach for clustering of bio-informatics databases”, In Proceedings of the IEEE International Conference on Fuzzy Systems (FUZZ-IEEE) pp 2077-2082, IEEE.
- [111] Lambora, A., Gupta,K., and Chopra,K. ,(2019), "Genetic Algorithm- A Literature Review", International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (Com-IT-Con) .
- [112] Hasan, M.J.A., and Ramakrishnan, S., (2011), "A survey: hybrid evolutionary algorithms for cluster analysis", Artif Intell Rev, Springer.
- [113] Deo, A., Gangrade. J., Gangrade, S. ,(2019) “A PSO Algorithm Based Web Page Retrieval System”, International conference on Recent Advances in Interdisciplinary Trends in Engineering & Applications, SSRN-ELSEVIER.
- [114] Karaboga, D., and Ozturk, C., (2011), “A novel clustering approach: Artificial Bee Colony (ABC) algorithm”, Applied Soft Computing, Elsevier.
- [115] Xu,Y., Fan, P., and Yuan, L., (2013), “A Simple and Efficient Artificial Bee Colony Algorithm “, Mathematical Problems in Engineering Volume.
- [116] Anuradha, G., and Lavanya, D. G., (2014), “Artificial Bee Colony (ABC) Approach for Ranking Web Pages”, International Journal of Computer Applications, Volume 99– No.1.
- [117] Rani, B.M., Sujata, B., and Priyanka, M. ,(2017), “ Ranking Web Pages using Ant Rank and Bee Colony Algorithm”, International Journal of Advanced Research in Computer and Communication Engineering, Vol. 6, Issue 7.
- [118] Sakulin, S., Alfimtsev, A., Solovyev, D., and Sokolov, D., (2018), “Web Page Interface Optimization Based on Nature-Inspired Algorithms”, International Journal of Swarm Intelligence Research, Volume 9, Issue 2.
- [119] Gupta, A., Dixit, A. and Sharma, A.K. (2017), “An Efficient User Preference and Page Relevance based Page Ranking Algorithm”, Indian Journal of Science and Technology, Volume 10(37).

- [120] Kavitha, C., Sadasivam, S.G., and Kiruthika,S., (2014), “ Semantic similarity based web document classification using Artificial Bee Colony (ABC) algorithm”, WSEAS TRANSACTIONS on COMPUTERS.
- [121] Anuar,S., Selamat, A., and Sallehuddin, R., (2016), “A modified scout bee for artificial bee colony algorithm and its performance on optimization problems”, Journal of King Saud University, Computer and Information Sciences, Elsevier.
- [122] Ju, C. & Xu, C., (2013), “A New Collaborative Recommendation Approach Based on Users Clustering Using Artificial Bee Colony Algorithm”, The Scientific World Journal, Hindawi.
- [123] Abdullah, H.S., and Hadi, M.J., (2014), “Artificial Bee Colony based Approach for Web Information Retrieval”, Eng. & Tech. Journal, Vol. 32, Part (B), No. 5.
- [124] Gao, H., Shi. Y., Pun, C. and Kwong, S., (2018), “An Improved Artificial Bee Colony Algorithm with its Application”, IEEE.
- [125] Xiao, S., Wang, W., Wang, H., Tan, D., Wang, Y., Yu, X. and Wu, R. ,(2019), “ An Improved Artificial Bee Colony Algorithm Based on Elite Strategy and Dimension Learning”, Mathematics.
- [126]Gadekallu,T.R., Khare,N., Bhattacharya,S., Singh,S., Maddikunta,P.K.R., and Alazab,M., (2020), “Early Detection of Diabetic Retinopathy Using PCA-Firefly Based Deep Learning Model” , MDPI, Electronics.
- [127] Bhattacharya,S., Krishnan,S.R., Maddikunta,P.K.R., Kaluri,R., Singh,S., Gadekallu,T.R., Alazab,M., and Tariq,U. ,(2020), “A Novel PCA-Firefly Based XGBoost Classification Model for Intrusion Detection in Networks Using GPU” , MDPI, Electronics.
- [128]Gadekallu,T.R., Khare,Reddy, M.P.K., Lakshmana,K., Rajput,D.S., Kaluri,R., and Srivastava,G., (2019), “Hybrid genetic algorithm and a fuzzy logic classifier for heart disease diagnosis”, Evolutionary Intelligence, Springer Nature.
- [129] Iwendi,C.,Maddikunta,P.K.R.,Gadekallu,T.R.,Lakshmana,K.,Bashir,A.K., and Piran,J., (2020), “A metaheuristic optimization approach for energy efficiency in the IoT networks”, John Wiley & Sons.
- [130]Gadekallu,T.R.,Reddy,P.K., Lakshmana,K., Kaluri,R., Rajput,D.S., Srivastava,G., and Baker, T., (2020), “ Analysis of Dimensionality Reduction Techniques on Big Data", Vol 8, IEEE Access.
- [131] Baykasoglu, A., Ozbakir, L., and Tapkan, P., (2007), "Artificial Bee Colony Algorithm and Its Application to Generalized Assignment Problem", Swarm Intelligence: Focus on Ant and Particle Swarm Optimization.

- [132] Cummins, R., (2017) "Information Retrieval", Natural Language and Information Processing (NLIP) Group.
- [133] Jarvelin, K. and Kekelainen, J., (2002), "Cumulated Gain-Based Evaluation of IR Techniques", ACM Transactions on Information Systems, Vol. 20, No. 4.
- [134] Jarvelin, K. and Kekelainen, J., (2017), "IR evaluation methods for retrieving highly relevant documents ", Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, pp. 41–48.
- [135] Chen, W., Liu, T.Y., Lan, Y., Ma, Z., and Li, H., "Ranking Measures and Loss Functions in Learning to Rank".

APPENDICES

A1:Pseudocode of Evolutionary Algorithm:

```
Procedure EA; {  
    t=0  
    initialize population P (t);  
    evaluate P (t);  
    until (done) {  
        t=t+1  
        parent-selection P(t);  
        recombine P(t);  
        mutate P(t);  
        evaluate P(t);  
        survive P(t);  
    }  
}
```

A2:Pseudocode of Standard Genetic Algorithm:

```
BEGIN
    Generate initial population
    Compute fitness of each individual
While NOT finished DO
    BEGIN
        FOR population-size
            BEGIN
                Select two individual from old generation for mating
                Recombine the two individuals to give two offspring
                Compute fitness of the two offspring
                Insert offspring in new generation
            END
        IF population has converged THEN
            Finished=TRUE
        END
    END
END
```


A3:Pseudo-code of ABC Algorithm (Karaboga and Akay, 2009)

1. Load training samples
2. Generate the initial population $z_i= 1 \dots SN$
3. Evaluate the fitness (f_i) of the population
4. Set cycle to 1
5. Repeat
6. **FOR** each employed bee {
 Produce new solution v_i by using (6)
 Calculate the value f_i
 Apply greedy selection process }
7. Calculate the probability values p_i for the solutions (z_i) by (5)
8. **FOR** each onlooker bee {
 Select a solution z_i depending on p_i
 Produce new solution v_i
 Calculate the value f_i
 Apply greedy selection process }
9. **If** there is an abandoned solution for the scout
 then replace it with a new solution which will be randomly produced by (7)
10. Memorize the best solution so far
11. cycle = cycle + 1
12. **until** cycle= MCN

A4: Algorithm scRank

Input: User Query

Output: Ranked Web Pages (WP)/documents

Steps:

1. Enter query Q consisting of 'n' terms that extract 'N' pages by removing stopwords.
2. Count number of terms 'n' of query 'Q'.
3. Calculate cumulative weight of the nodes using the formula-

$$CW(n) = I(w) / (I(w) + O(w))$$

where

CW(n)- cumulative weight of node

I(w)- Indegree weight of node

O(w)- Outdegree weight of node

4. Webpages/ documents are clustered on basis of following criteria-

Case I: If all term 'n' of 'Q' appear in WP_i where i=1, 2, 3,...M group them in C_i, where i=1, 2, 3.

Case II: If 'n-1 <= WP >= n/2' group them in C_i, where i=1, 2, 3.

Case III: If one term 'n' match group them in cluster C_i, where i=1, 2, 3, exclude.

5. Calculate cumulative weight of web pages with respect to the Query which are grouped into clusters using the formulas-

$$W(x) = N(xi) / \sum_{i=0}^n N(x_n)$$

where

W(x) - Weight of node x

N(x_i) - Number of times x_i term of query occur in web page

N (x_n) - Total occurrence of all term in web page

$$CW_q(x) = \sum_{i=0}^n x_i$$

where

CW_q(x) - cumulative weight of query in x page

6. Calculate scRank of the web pages grouped in clusters using Formula-

$$scRank(N) = (1 - \infty) + CW_q(x) + CW(n) + [(rank(N1) / O(N1) + rank(N2) / O(N2) + \dots + rank(Nn) / O(Nn)] * \infty$$

where

$scRank(N)$ - rank of web pages N

α – constant term between 0-1, taken as 0.85 as standard value.

$CW_q(x)$ - cumulative weight of query in x page

$CW(n)$ - cumulative weight of node

7. Finally ranked web pages are retrieved.

A5: Algorithm gRank

Input: User Query

Output: Relevant Ranked Web Pages (RW)

Steps:

1. User entered query Q comprises of 'n' terms that extract 'W' pages by removing stopwords.
2. Count number of terms 'n' of query 'Q'.
3. Calculate Inlink (I) and Outlink (O) weight of the nodes.
4. Cluster Web pages using following criteria-

Case I: If all term 'n' of 'Q' appear in W_i where $i=1, 2, 3, \dots, X$ group them in CL_i , where $i=1, 2, 3$.

Case II: If ' $n-1 \leq W \leq n/2$ ' group them in CL_i , where $i=1, 2, 3, \dots, X$.

Case III: If one term 'n' match group them in cluster CL_i , where $i=1, 2, 3, \dots, X$, exclude.

5. Calculate Fitness value of the web pages $F(X)$ using the formulas-

$$F(X) = (W(X) / Q(n)) + (I(W) / O(W))$$

where

$W(X)$ - Term weight of page X corresponding to Q

$Q(n)$ - Term weight n of query Q

$I(W)$ - Inlink weight of web page X

$O(W)$ - Outlink weight of web page X

6. Calculate probability of the web pages $P(X)$ using Formula-

$$P(X) = F(X) / \sum_{i=1}^X F(X)$$

where

$F(X)$ - Fitness of the web page

7. Calculate Expected Count of the web page $EC(X)$ and Actual Count of the web page $AC(X)$ which is approximate of $EC(X)$ using Formula-

$$EC(X) = F(X) / \overline{F(X)}$$

where

$F(X)$ - Fitness of the web page

$\overline{F(X)}$ – Average of fitness of the web pages

8. Select Mating pool and mate. Apply one- point crossover and Flip mutation to produce new population.
9. Fitness value is again evaluated for the New generation and the steps 5- 8 is repeated till termination condition is reached, which is repetition of same strings of chromosomes.
10. Finally web pages are ranked on the basis of Fitness values.

A6: Algorithm BeeRank:

Input: Query as Bee

Output: Ranked web pages as Food Source (FS)

Steps:

1. Let there are randomly distributed Food Source (Web Pages) comprises of 'S' solution where each solution $S_i = \{i=1, 2, 3 \dots N\}$. $K \in \{1, 2, 3 \dots N\}$ represented in 'i' dimension, where $i! = k$. Query is represented by 'j' dimension vector where $j = \{1, 2 \dots Q\}$.
2. The Query 'q' having 'n' terms, randomly extracts 'W' pages by removing stopwords.
3. Query density 'qd', document density 'dd' and density of nodes having indegree and outdegree is evaluated.
4. Food source containing a single term of query is abandoned by the bee as it has low food amount, and new sources are searched.
5. Fitness of Food Source 'f(FS)' is calculated using-

$$f(FS) = \text{sqr}t[(dd(W) / qd)] / (I(W) / O(W))$$

where

dd- document density of web page

qd- query density

I (W) - Indegree of web page

O (W) - Outdegree of web page

6. Calculate probability p(FS) of food source using-

$$p(FS) = f(FS) / \sum_{i=1}^N f(FS)$$

where

f (FS)- fitness of food source

7. Calculate quantity of food source using-

$$x_{ij} = x_{j \min} + C[-1, 1](x_{j \max} - x_{j \min})$$

where

x_{ij} - i^{th} food source in j^{th} direction

$x_{j \min}$ - lower bound of food source in j^{th} direction

$x_{j \max}$ - upper bound of food source in j^{th} direction

C - random number between (-1, 1)

8. Calculate strength of neighbor food source using-

$$N_{ij} = x_{ij} + C_{ij}(x_{ij} - x_{kj})$$

where

N_{ij} - neighbor food source i^{th} in j^{th} direction

x_{ij} - i^{th} food source in j^{th} direction

x_{kj} - second vector ($i \neq k$)

C - random number between (-1, 1)

9. Old food source (x_{ij}) and new neighbor food source (N_{ij}) is compared and value of x_{ij} is updated by selecting the best value between them.

10. Calculate fitness of updated food source using-

$$f(UFS) = 1 / (1 + f(x_{ij}))$$

where

$f(UFS)$ - fitness value after updation of food sources

$f(x_{ij})$ - fitness value for quantity of food sources

11. Calculate probability and probability percentage of new food source using-

$$p(UFS) = f(UFS) / \sum_{i=1}^N f(UFS)$$

where

$p(UFS)$ – probability of updated food source

$f(UFS)$ - fitness value after updation of food sources

12. BeeRank of food sources is calculated using-

$$BeeRank(W) = (1 - \alpha) + f(UFS) * \alpha$$

where

f (UFS) - fitness value after updation of food sources

α -constant (0-1) standard taken as 0.85

13. Finally ranked web pages.

CURRICULUM VITAE

SHADAB IRFAN

Permanent Address: - 156-B- 18 Shams Nagar
Karelabagh, Allahabad-211016

Mob: - 8299054072, 9336847390

E-mail:- shadab710@gmail.com



Career Objective:

To work with a reputed organization that will provide me a good platform wherein I can utilize my teaching and administrative skills in order to gain recognition and take it to the zenith of success.

Professional Qualification:

- Pursuing **Phd** (CSE) from Galgotias University, Greater Noida.
- **M.Tech** (SE) from Motilal Nehru National Institute of Technology (**MNNIT**), Allahabad, in 2012 with 75%.
- **MCA** from Allahabad Agricultural Institute Deemed University (**AAI-DU**), Allahabad, in 2005 with 80%.
- “**A**” Level (PGDCA) from DOEACC Society, New Delhi, in 2000 with 65%.
- “One Year Adv.Diploma” from UPTEC, Allahabad, in 2000.
- Post Graduation (M.A) in English Literature from Allahabad University (AU), in 2000 with 55%.
- Graduation (B.A) from Allahabad University(AU), in 1998 with 58%.
- 10+2 from ISC Board from GHS, Allahabad in 1995 with 62%.
- 10th from ICSE Board from GHS, Allahabad in 1993 with 67%.

Work Experience: (17 years)

- Presently working at Sri Ramswaroop Memorial University, Lucknow from 23 March 2021 till date.
- 2 year teaching experience as Guest Faculty in Hamidia Girls’ Degree College, Allahabad from 16 Nov 2018 to 16 March 2021.
- 10 years teaching experience as Assistant Professor in SP Memorial Institute of Technology (**SPMIT**), Allahabad (UPTU affiliated- College code-364) from 12 July 2008 to 15 Nov 2018.

- 2.4years teaching experience as a Faculty in United College of Engineering & Research (**UCER**), Allahabad (UPTU affiliated College Code-010) from 2006 to 2008.
- 1 year teaching experience as a Faculty in recognized IGNOU'S IICS study Centre from 2005 to 2006.
- 2 year teaching experience as a Faculty in NEIT Institute from 2000 to 2002.

Subjects Taught:

Graduate Level:

- Computer Programming & Problem Solving
- Object Oriented Paradigm
- Web Technology
- Real Time System
- Software Engineering
- Data Mining & Warehousing
- Computer Architecture & Organization
- Software Testing
- Graph Theory
- Software Project Management
- Information Technology
- C++
- Microprocessor
- Semantic Web & Web Services
- Digital Logic Design
- Selenium
- Big Data (Solr, Flume, Mahout, Pig)
- Data Science using Python
- Big Data Security

Post Graduate Level:

- Web Technology
- Data Mining & Warehousing
- Software Engineering
- Software Project Management

Technical Skills:

Operating System	: MS-DOS, WINDOWS 95 / 98 / 2000 / XP, Linux.
Language	: 'C', 'C++', Java, Assembly Language, Visual Basic.
Scripting Languages	: HTML, DHTML, XML, JavaScript, VBScript, JSP.
Databases	: MS-Access, SQL, PL/SQL, SQL SERVER, Oracle.
Package/Tools	: MS-Office, FoxPro.

Research Publications:

Book Published:

- Book authored title “**Real Time System**”, 2014, ISBN: 978-93-5014-562-3 published by Katson Publication, New Delhi.

Paper Published: 14- (International Conf.-8, National Conf.-2, Journal-5)

- Paper published entitled “**Improving Ranking Procedure In Information Retrieval Process Using scRank Algorithm**” in **International Journal of Advanced Science and Technology**, Vol. 144, No. x, (2020), pp. 1 – 18, NADIA, ISSN(P): 2005-4238,ISSN(O): 2207-6360 IJAST, DOI: 10.33832/ijast.2020.144.01.
- Paper published entitled “**Ranking Algorithm to Optimize the Retrieval Process Using Genetic Algorithm**” in **International Journal of Control and Automation**, Vol. 13, No. 2, (2020), pp. 383 – 396, ISSN: 2005-4297 IJCA, SERSC.
- Paper published entitled “**BeeRank A Heuristic Ranking Model to Optimize the Retrieval Process**”, in International Journal of Swarm Intelligence Research (IJSIR), Volume 12, Issue 2, Article 6, (2021) in Press.
- Paper published entitled “**Efficient Ranking Framework for Information Retrieval Using Similarity Measure**” in 3rd International Conference On Computational Vision and Bio Inspired Computing- **ICCVBIC 2019**, held on 25-26 Sep 2019 and published in **Springer - Computational Vision and Bio-Inspired Computing**.
- Paper published entitled “**Ranking Web Pages Using Cosine Similarity Measure**” at International Conference on Computing, Power and Communication Technologies (GUCON 2019) to be held on 27-28 Sep 2019, at Galgotias University Greater Noida, India
- Published a paper entitled “**Web Usage Mining in Ranking Algorithms**” in International Journal of Innovative Research in Science, Engineering and Technology (**IJRSET**), Volume 7 Issue 10 October 2018, ISSN 22319-8753.
- Presented and published entitled “**Analysis and Challenges of Web Ranking Algorithms**” at 4th International Conference on Computing, Communication and Automation –**ICCCA 2018**) held at Galgotias University on Dec 14-15, 2018 in Greater Noida, India.
- Presented a paper entitled “**A Review on Different Ranking Algorithms**” for International Conference on Advances in Computing, Communication Control and Networking –**ICACCCN (ICAC3N'18)** held at Galgotias College of Engineering and Technology on October 12-13, 2018 in Greater Noida, India.
- Published a paper entitled “**Web Mining for Information Retrieval**” in International Journal of Engineering Science and Computing (**IJESC**), Volume 8 Issue 4 April 2018, ISSN 2250-1371.

- Presented and published a paper entitled “**Optimization of K-Means Clustering Using Genetic Algorithm**” at International Conference on Computing, and Communication Technologies for Smart Nation (**IC3TSN2017**) held at GD Goenka University (12-14 Oct 17), Gurgaon (In Press).
- Presented and published a paper entitled “**Optimization of Information Retrieval Using Evolutionary Computation: A Survey**” at International Conference on Computing, Communication and Automation (**ICCCA2017**) held at Galgotias University (5-6 May 17), Greater Noida.
- Presented and published a paper entitled “**Information Retrieval in Big Data Using Evolutionary Computation: A Survey**” at International Conference on Computing, Communication and Automation (**ICCCA2016**) held at Galgotias University (29-30 April 16), Greater Noida.
- Presented and published a paper entitled “**Security Challenges in Big Data: A Survey**” at National Conference on Emerging Trends in Computational Management & Technologies (**NCETCMT- 2016**) held at SPMIT (26-27 Feb 16), Allahabad, **ISBN: 978-93-85477-65-2**.
- Presented and published a paper entitled “**A Generalization Approach Applied to Security Ontology**” at **IEEE 2nd National Conference** on Emerging Trends in Engineering & Technology (**ETET-2013**) held at SBIT (8-9 Feb 13), Sonapat, **ISSN 2277-8764**.
- Presented and published a paper entitled “**A Concept of Out Degree in CFG for Optimal Test Data Using Genetic Algorithm**” at **IEEE Proceedings in International Conference** on Recent Advances in Information Technology (**RAIT-2012**) held at ISM (15-17 March 12), Dhanbad, **ISBN:978-93-80813-17-2**.

Conference/Seminars/ Workshops/FDP:

- Certification Program in **Python Programming** from 1 Sep-11 Sep 2020 from LetsUpgrade, under National Skill Development Corporation (NSDC) and ITM Edutech Training Pvt Ltd.
- One Week Online Training Program on **ICT Tools for Teaching, Learning and Administration** August 04 – August 10, 2020 Organized By Department of Electronics and Communication, J. K. Institute of Applied Physics & Technology, University of Allahabad, Prayagraj
- Webinar on "**How to Write and Structure Persuasive Research**" on 12 July 2020 organized by Matterhere by Karl Weber.
- Global E-Symposium on "**Global Pandemic Outbreak: Role of Technology & Automation**" from 1-5 June 2020, organized by JSS Academy of Technical Education, India.

- Indo-Muscat one-week Virtual workshop on "**Research Writing Skills**" from 26th May to 30th May, 2020, Organized by University Center of Research and Development(UCRD) Galgotias University, India.
- Workshop on "**Online Basic Optimization Technology**" from 04 May, 2020 to 06 May, 2020 organized by REST Society for Research International (RSRI), Krishnagiri, Tamil Nadu, India.
- FDP on "**Nature Inspired Algorithms for Solving Complex Engineering Problems (FDP-NIASCEP 2018)**", at Galgotias University (23-27 April 2018).
- An International Workshop on "**Publishing in High Impact Journals**" at Zakir Husain College of Engineering & Technology, Aligarh Muslim University, (25 April 2017).
- 1 week workshop on "**Big Data & Soft Computing: Techniques & Applications**" at Zakir Husain College of Engineering & Technology, Aligarh Muslim University, (05-09 Oct 2016).
- International Conference on **Computer and Computational Sciences (ICCCS-2015)** organized by Department of Computer Science & Engineering at ITS Engineering College Gr. Noida (27 Jan 2015).
- FDP on "**Teaching Methodologies**" at SPMIT, (29-31 Aug 2014) with expertise from NITTTR Chandigarh and MNNIT Alld.
- FDP on "**Curriculum Enhancement Awareness in Information Security**" at MNNIT, (14-24 Dec 2009)
- FDP on "**Issues & Design of Distributed Systems and its Applications**" at MNNIT, (27 Jan-7 Feb 2009)
- National Seminar of "**Management of Change-Shifting Paradigm in Business Model**" at UCER (3rd May 2008)
- National Seminar on "**Exploring New Horizons of Computer Application in Modern Society**" at United Institute of Management (May 5th 2008)
- National Seminar on "**Advances in Computing & its Applications**" at UCER, (April 12-13 2008)
- Short course on "**LINUX Installation and Programming**" at UCER by National Resource Centre for Free/Open Source Software (NRCFOSS) Chennai (23-28 July 2007)
- Short **FDP** Programme for Faculty by National Institute of Technical Teachers Training & Research, Chandigarh (8-18 Jan 2007)
- International Conference on "**Science, Technology & Society 21st Century Perspective**" at UCER (23rd Nov 2006)

Administrative Responsibilities:

- Worked as **Departmental Head** of CS/IT department at SPMIT from July 2008 till Nov 2018.

- Worked as **Exam Cell Head** for Internal Examination at SPMIT from 2008-2010.
- Worked as **ACS** for **UPTU Semester Examination** at SPMIT from 2009-2011.
- Worked as **ACS** for **BTE Annual Examination** at SPMIT in 2015-16.
- Performed the **Evaluation** work at Krishna Institute of Engineering & Technology, Ghaziabad (27 Jan -5 Feb 2010)
- **Faculty Mentor** for **IBM Academic Initiative Project INVITE 2007**
- **Faculty Mentor** for **IBM Academic Initiative The Great Mind Challenge 2008**

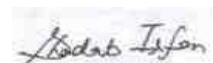
Projects Undertaken:

1. Project Name : Online Teleshopping
Duration : Six Month
Tools Used : SQL Server/Java
2. Project Name : Online Railway Enquiry System
Duration : Six Month
Tools Used : SQL Server/Visual Basic
3. Project Name : Production Planning and Control Of Shervani Industrial Syndicate Limited
Duration : One Month
Tools Used : Oracle/Visual Basic
4. Project Name : Library Management System
Duration : Six Month
Tools Used : FoxPro/ Win/98

Personal Information:

Father's Name : Mr.Irfan-ul Haque.
 Mother's name : (L) Mrs.Shama Irfan.
 Date of Birth : 07-10-1976.
 Gender : Female
 Marital Status : Unmarried
 Nationality : Indian
 Religion : Islam
 Languages known : English, Hindi and Urdu.

Place: Allahabad
 Date:



(SHADAB IRFAN)