Performance Analysis Of Similarity Coefficients In Web Information Retrieval Using Genetic Algorithm

Vikas Thada, Dr. Vivek Jaglan

Abstract: Crawling is a process in which web search engines collect data from the web. Focused crawling is a special type of crawling process where crawler look for information related to a predefined topic[1].In this paper a method for finding out the most relevant document among a set of documents for the given set of keyword is presented. Relevance checking is done with the help of Rogers-Tanimoto, MountFord and Baroni-Urbani/Buser similarity coefficients. The method uses genetic algorithm to show that the average similarity of documents to the query increases when Probability of mutation is taken as low and Probability of crossover is taken as high. The method does the performance analysis of different similarity coefficients on the same set of documents and selects the best combination of ProC and ProM to achieve maximum relevancy using of Rogers-Tanimoto, MountFord and Baroni-Urbani/Buser similarity coefficients.

Index Terms: Algorithm, Coefficients, Crawling, Focused, Genetic ,Information, Ranking ,Retrieval ,Web.

1 INTRODUCTION

Design of most focused crawlers is based on the vector space model. The model is used to judge the evenness of web pages and general web search algorithms. The relevance in turn work as guide in following target links [2]. One of the most important module of search engine is ranking module. The task of ranking module is to assign some ranking score to relevant pages using some criterion. Output of ranking module is an ordered set of pages according to their rank i.e. pages with high rank are near the top of the list and low rank pages are at the bottom of the list. These pages are then presented to the user in their ranking order. A GA based approach using Rogers-Tanimoto, MountFord and Baroni-Urbani/Buser similarity coefficients is taken in this paper for ranking the retrieved documents.

2 GENETIC ALGORITHM

GAs are search algorithms that follow the concept of natural selection and genetics[3]. GA are powerful and very efficient search and optimization techniques motivated by the natural selection theory of Darwin [4]. Genetic Algorithms [5] are based on the principle of heredity and evolution which claims “in each generation the stronger individual survives and the weaker dies”. Therefore, each new generation would contain stronger (fitter) individuals in contrast to its ancestors. The process of GA’s is iteration based of constant population size of candidate solutions. In each generation/iteration each chromosome’s fitness in the current population is evaluated and new population evolves. Chromosomes with higher fitness values goes through reproduction phase in which selection, crossover and mutation operators are applied to get new population. Chromosomes with lower fitness values are discarded. Again this generated new population is evaluated and selection, crossover, mutation operators are applied. This process continues until we get an optimal solution for the given problem.

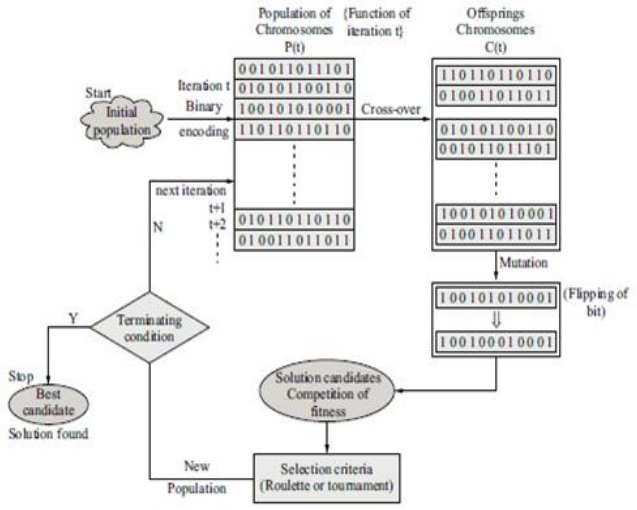

Fig. 1. Basic Operation of Genetic Algorithm [14]

2.1 Fitness Evaluation

Fitness function is a function which is responsible for evaluating some value to indicate among number of solutions which one is optimum. It can also be considered as a measure of performance or fitness to show how fit is the candidate solution. The problem of IRS using GA is to retrieve documents using this fitness function. For finding the relevant document on the basis of some similarity measures we can have number of relevancy methods.

Table 1

Coefficients Used As Fitness Function In Research [6]

S.N	Cofficient Name	Similarity Formula
1.	Rogers & Tanimoto	(p+s) / (p+2*(q+r)+s)
2.	Baroni-Urbani/Buser	(p+sqrt(p*s)) / (p+q+r+sqrt(p*s))
3.	Mountford	2*p/(2*q*r+p*q+p*r))

For the calculation of similarity metric we define few parameters p,q,r and s as (n = p+q+r+s).

p= (x=1 and y=1) (total match)
q= (x=1 and y=0) (single match)
r= (x=0 and y=1) (single match)
s= (x=0 and y=0) (no match)

This is shown in table 2, where

Table 2

Variables Used to Calculate Binary Similarities/Dissimilarities[7,6,8]

	y=1	y=0
x=1	p=1/1 in both A and B	q=1/0 only in A
x=0	r=0/1 only in B	s=0/0 in none of A and B

Where A and B may be any query or document represented in binary form.

2.2 Selection

Once the fitness evaluation process is done next step is to perform selection operation. Process of selection operation is based on the principle of ‘‘survival of the fittest’. Higher fitness valued chromosomes goes through reproduction. Lower fitness valued chromosomes are discarded. There are number of ways to implement this operator, but all relies on the concept that candidates with good fitness values are to be preferred over poor fitness values. The idea is to give preference to better individuals. This selection operation does the replication of candidate chromosomes with good fitness values and eliminating those with poor fitness values [9].

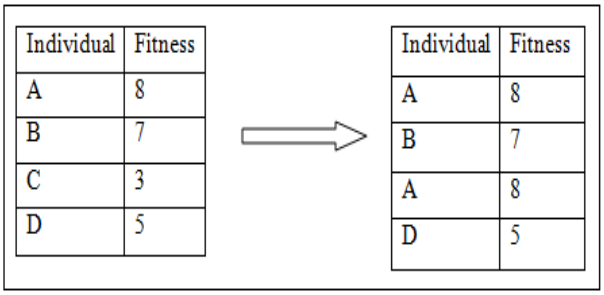

Fig. 2. Selection Operator on a Population of 4 Individuals [9]

The research work uses roulette wheel selection method as selection operator [10]. It is also known as fitness proportionate selection method.

2.3 Crossover[3,11]

In the crossover operation mating of two chromosomes is performed that gives birth to two new offspring. This operation of crossover always happens with one parameter that is known as probability of crossover (ProC). When ProC is say 0.8 it means only 80% of the total population goes for crossover operation. Rests 20% chromosomes remain abstain from this operation and has no effect of crossover. Motive behind performing crossover operation is to explore new solutions and exploit use of old solutions. GA forms an optimum solution by mating two fit chromosomes together. Chromosomes with higher fitness will always have good selection probability then others with lower fitness values, thus a good solution moves from one generation to next generation.

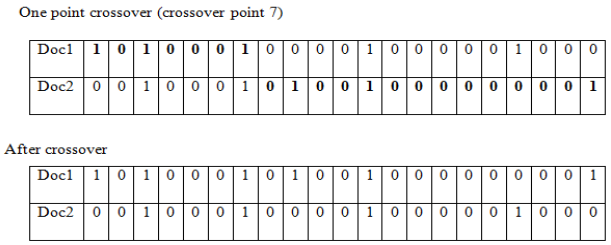

Fig. 3. Single Point Crossover Explained

2.4 Mutation[3,11]

Mutation involves changing one bit of a chromosome from 0 to 1 or viceversa. This is performed under the constraint parameter called probability of mutation (ProM). For example if ProM is 0.10 then 10% genes of total chromosomes will go for mutation. The concept of mutation is based on this natural theory that varying breeds are possible only by varying gene values. After this operation fitness quality of new chromosomes may be high or low then old ones. In case new chromosomes are poor then old ones they are removed during selection process. The motive behind mutation is regaining the lost and discovering varying breeds. For example: randomly mutate chromosome at position 5.

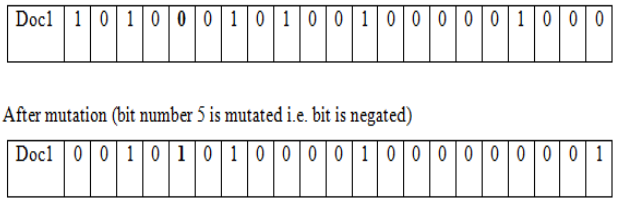

Fig. 4. Mutation Operation Explained

3. Research Tools

This section is about the introduction of tools which helped us to find the top keywords from relevant document. These keywords actually used for making the chromosomes; backbone of GA in which the research implementation work was carried out. We discuss a brief introduction of tools Text Analyzer and Keyword Density Checker. Both the tools were used to find out top most keywords present on webpage with highest frequency.

3.7.1 Textalyser

Textalyser[12] is tool for analyzing text or website online. The tool can be used for calculating detailed statistics of text like frequency of top occurring words in text or webpage etc. The tool has good application for translation purpose and webmasters for ranking webpages, for simply normal users to know about top keywords from a text. Using this tool keyword density can be easily found out. Further relative importance of word or expressions can also be found out. Webmasters can use this tool for the analysis of the links on their pages.

3.7.2 Keyword Density Checker

The keyword density checker tool [13] is an online tool which can be used for finding density of top occurred keywords. These keywords are then displayed at the top . Input to the tool can be URL only against both URL and text for textalyser tool. The tool crawls the given URL , bring out words normally as done by search engine , remove punctuations and return result in the form of density of the top URL keywords in the form of keyword cloud.

4. Experimental Setup & Results

In general the web documents are encoded in strings of 0's and 1's as shown in the figure 1. The documents have been obtained using some search query. The set of documents also contains some non relevant documents too. In the experimental setup of GA 50 documents have been retrieved for each different query. This set of 50 documents serves as document database. The query is also encoded in terms of string of 0 and **1.** Process of experiment is as follows:

- **i.** Query is input to the Google search engine.
- **ii.** Top frequent keywords of each retrieved document are extracted using Textalyser or Keyword Density Checker online tool as discussed in chapter 3 and making list of keywords.
- **iii.** Generate initial population by encoding retrieved documents to chromosomes for each query.
- **iv.** Initial population now consists of these encoded documents. Pass this initial population to GA algorithm.
- **v.** Either some maximum generation or some predefined fitness value is not achieved repeat step 4. At the end of this step we get an optimized document for retrieval.
- **vi.** Optimized document chromosome can be decoded and document can be retrieved from stored documents related to each query.
- **vii.** Rank the documents on the basis of fitness value as obtained in step 4.

4.1 Experimentation

The research work conducted the tests for 15 different queries and 30 documents for each query. The experimentation tests for all 15 queries with RT, MF and BUB coefficients chosen as fitness function. A complete MATLAB code has been written with roulette wheel selection operator, random point crossover with different rates of crossover and mutation. Experiment conducted with various GA parameters as : probability of crossover ProC = 0.7, 0.8 and 0.9 along with probability of mutation (ProM =0.01, 0.10, 0.30) to do performance analysis of this GA based retrieval system. The efficiency parameter is average relevance. Average relevance from three fitness functions are compared and best combination for crossover and mutation probability for any one among three similarity function is selected as best . Using this best selected similarity function the document's rank before and after the algorithm are checked and new ranks to document are assigned. The retrieval process now retrieves documents using this new ranking scheme.

4.2 Results

In this section we show the results in tabular and graphical form for all three similarity coefficients viz. RT, MF and BUB. We vary probability of mutation ProM from 0.01 to 0.10 and 0.30 and keep ProC=0.5, 0.7 and 0.9 for all the values of ProM. Keeping together ProC and ProM we get total nine possible combinations for checking the performance of said similarity coefficients on retrieved documents. Out of nine possible results we got best was for ProC=0.9 and ProM=0.1. Table 3 shows the queries used in research work. Next three table 4 shows maximum average relevancy for Proc=0.5, 0.7 and 0.9 with ProM=0.1. Table 5 shows maximum average relevancy for ProC=0.5, 0.7 and 0.9 with ProM= 0.10. Table 6 shows maximum average relevancy for ProC=0.5, 0.7 and 0.9 with ProM= 0.30.

Table 3
Queries Used in Research Work

S.N	Chromosome Length	Keywords
1.	33	Delhi, Gang, Rape, Case
2.	39	Bomb, Blast, Boston, Marathon
3.	37	Search ,Engine, Optimization,Seo
4.	49	Anna, Hazare, Anti, Corruption
5.	53	Osama, Bin, Laden, Death, Killed
6.	47	Stock, Market, Mutual, Fund
7.	55	Fiber, Optic, Information, Technology
8.	66	Health, Medicine, Medical,Disease
9.	44	Artificial,Intelligence,Neural, Network
10.	40	Rajasthan, Royals,Amity,University
11.	49	Remote, Method, Invocation, RMI
12.	61	Gang, Rape, Kangaroo, Court
13.	69	Aam, Aadmi, Party, Arvind, Kejriwal
14.	50	Genetic,Algorithms,Optimization, Techniques
15.	64	Search, Engine, Web, Crawler

Table 4
Maximum Average Relevancy with ProC = 0.5, 0.7, 0.9 and ProM = 0.01

	ProC=0.5 ProM=0.01			ProC=0.7 ProM=0.01			ProC=0.9 ProM=0.01		
	RT	MF	BUB	RT	MF	BUB	RT	MF	BUB
QR1	0.5636	0.2789	0.5496	0.6113	0.2803	0.6023	0.6309	0.2948	0.6141
QR2	0.5157	0.1589	0.5421	0.579	0.213	0.5502	0.5442	0.2307	0.5446
QR3	0.5628	0.1789	0.5199	0.5862	0.2042	0.5308	0.5729	0.1871	0.5462
QR4	0.564	0.1316	0.5001	0.5818	0.1504	0.4567	0.5541	0.1503	0.4695
QR5	0.5468	0.1422	0.4796	0.5555	0.1364	0.4548	0.5381	0.1258	0.4509
QR6	0.5305	0.1336	0.4688	0.5679	0.1628	0.5044	0.5348	0.1337	0.4702
QR7	0.5182	0.1014	0.4382	0.5291	0.1284	0.4688	0.5462	0.1182	0.4507
QR8	0.4953	0.081	0.4234	0.5221	0.0916	0.4418	0.5268	0.088	0.4113
QR9	0.5953	0.1702	0.4949	0.5376	0.1556	0.503	0.5852	0.1666	0.5198
QR10	0.5542	0.1534	0.5287	0.5692	0.158	0.5418	0.5838	0.2001	0.5171
QR11	0.5292	0.1255	0.4734	0.5405	0.1291	0.4957	0.5604	0.1356	0.4397
QR12	0.5332	0.1022	0.466	0.5216	0.099	0.4529	0.5294	0.0848	0.4641
QR13	0.5169	0.0842	0.4024	0.4957	0.089	0.4357	0.5345	0.0815	0.4241
QR14	0.5887	0.1336	0.4602	0.5593	0.1384	0.4993	0.5552	0.1457	0.4808
QR15	0.5394	0.102	0.4675	0.5228	0.0948	0.4524	0.4987	0.088	0.4518

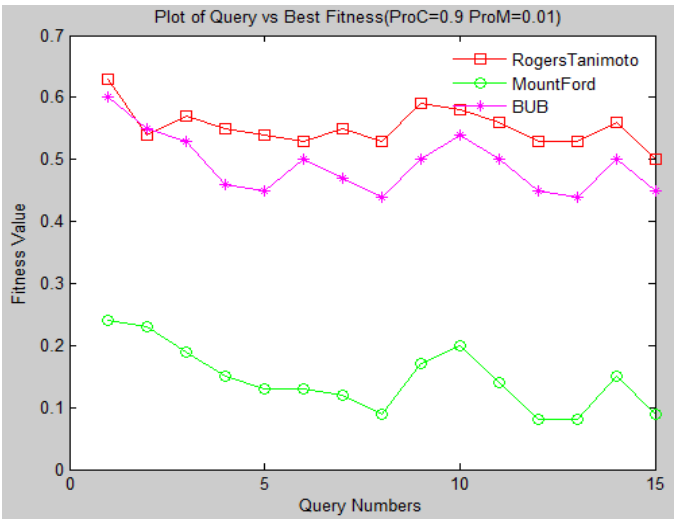

Fig. 5. Query vs best fitness for ProC=0.9,ProM=0.01 using RT,MF and BUB

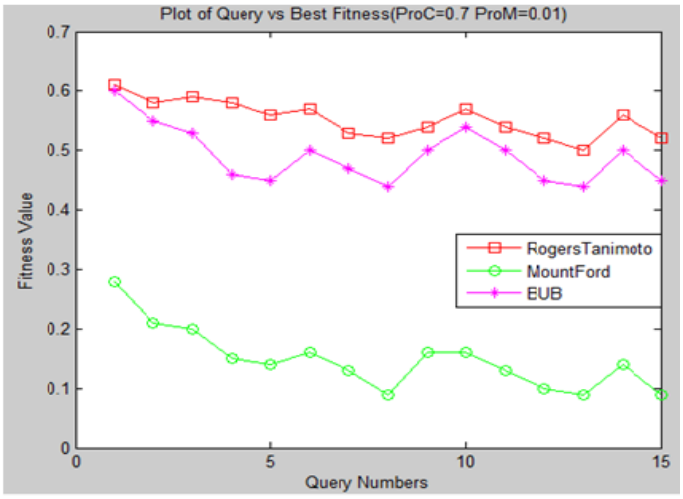

Fig. 6. Query vs Best Fitness for ProC=0.7, ProM=0.01 Using RT, MF and BUB

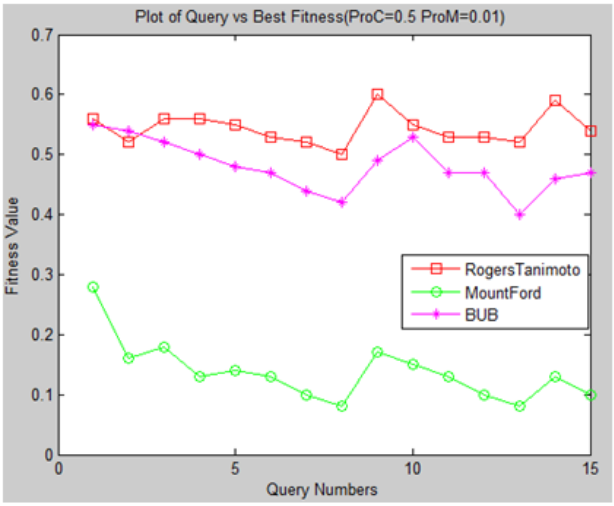

Fig. 7. Query vs best fitness for ProC=0.5, ProM=0.01 using RT,MF and BUB

Table 5
Maximum Average Relevancy with ProC = 0.5, 0.7, 0.9 and ProM = 0.1

	ProC=0.5 ProM=0.10			ProC=0.7 ProM=0.10			ProC=0.9 ProM=0.10		
	RT	MF	BUB	RT	MF	BUB	RT	MF	BUB
QR1	0.5286	0.1247	0.5325	0.4976	0.1643	0.4969	0.5043	0.1575	0.4643
QR2	0.4872	0.1337	0.4884	0.5153	0.1257	0.448	0.5045	0.1809	0.4694
QR3	0.4878	0.1481	0.5064	0.5015	0.1459	0.4795	0.4928	0.1385	0.514
QR4	0.4654	0.0873	0.4313	0.4787	0.0952	0.4281	0.459	0.09	0.4353
QR5	0.469	0.088	0.4192	0.4707	0.077	0.392	0.4791	0.089	0.3748
QR6	0.4795	0.088	0.4329	0.4833	0.08	0.4141	0.4929	0.1142	0.4165
QR7	0.4576	0.08	0.4088	0.4873	0.07	0.3866	0.477	0.0841	0.4163
QR8	0.4579	0.06	0.346	0.4496	0.069	0.3657	0.4417	0.064	0.3436
QR9	0.4871	0.1005	0.4223	0.4844	0.0961	0.4436	0.4746	0.105	0.4408
QR10	0.4829	0.11	0.441	0.5163	0.1177	0.4609	0.5022	0.1181	0.4538
QR11	0.4463	0.0878	0.4215	0.4611	0.1059	0.4276	0.4745	0.093	0.411
QR12	0.4671	0.073	0.3927	0.4623	0.06	0.3891	0.4477	0.06	0.3945
QR13	0.4411	0.055	0.3519	0.466	0.059	0.3661	0.4444	0.057	0.3689
QR14	0.4756	0.1013	0.4139	0.4603	0.089	0.4313	0.4757	0.0944	0.4066
QR15	0.4566	0.063	0.3885	0.4504	0.065	0.394	0.4304	0.067	0.3804

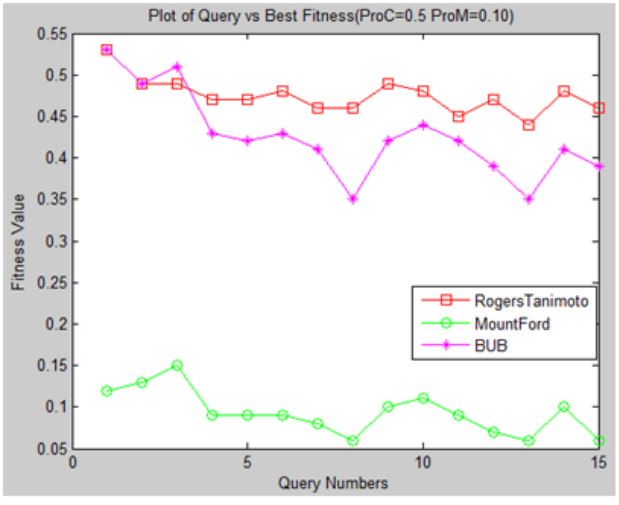

Fig. 8. Query vs best fitness for ProC=0.5, ProM=0.10 using RT, MF and BUB

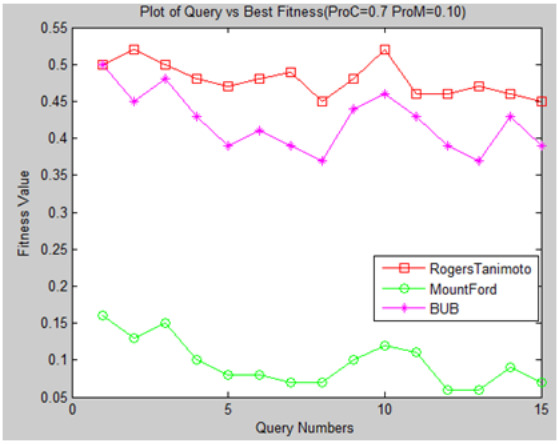

Fig. 9. Query vs best fitness for ProC=0.7, ProM=0.10 using RT, MF and BUB

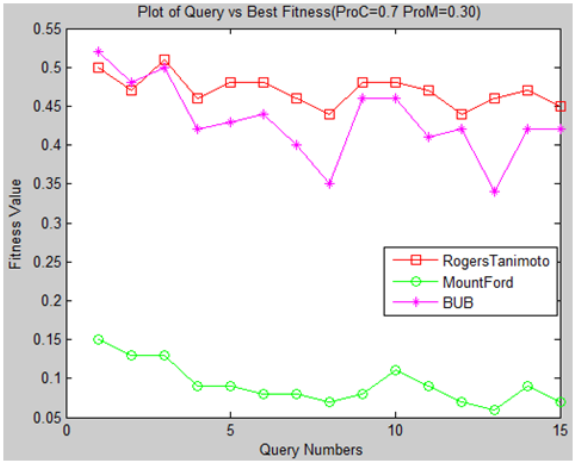

Fig.10. Query vs Best Fitness for ProC=0.7, ProM=0.30 Using RT, MF and BUB

Table 6
Maximum Average Relevancy with ProC = 0.5, 0.7, 0.9 and ProM = 0.30

	ProC=0.5 ProM=0.30			ProC=0.7 ProM=0.30			ProC=0.9 ProM=0.30		
	RT	MF	BUB	RT	MF	BUB	RT	MF	BUB
QR1	0.5254	0.1582	0.4817	0.5024	0.1467	0.5204	0.5225	0.1437	0.4947
QR2	0.4942	0.1473	0.4957	0.4669	0.1266	0.4753	0.4923	0.1344	0.4844
QR3	0.4995	0.1283	0.5006	0.5147	0.1332	0.4966	0.5045	0.1324	0.4463
QR4	0.479	0.0914	0.4207	0.4644	0.0912	0.423	0.4572	0.0859	0.4088
QR5	0.4463	0.078	0.3967	0.48	0.088	0.4279	0.4593	0.077	0.4129
QR6	0.476	0.0956	0.4277	0.4816	0.079	0.4444	0.4542	0.0949	0.4109
QR7	0.4578	0.0745	0.4082	0.4613	0.079	0.3952	0.431	0.072	0.3945
QR8	0.468	0.067	0.3612	0.4448	0.068	0.3514	0.4436	0.061	0.3823
QR9	0.4862	0.1035	0.4574	0.4843	0.08	0.463	0.495	0.1076	0.4224
QR10	0.4939	0.1101	0.4532	0.4797	0.1133	0.4582	0.4827	0.1128	0.4802
QR11	0.479	0.0947	0.4097	0.4656	0.0925	0.4109	0.4629	0.086	0.3986
QR12	0.4303	0.064	0.4001	0.4357	0.07	0.416	0.465	0.06	0.4013
QR13	0.4544	0.056	0.3415	0.4565	0.062	0.3423	0.4514	0.062	0.3569
QR14	0.4872	0.088	0.3987	0.4665	0.0872	0.4197	0.4957	0.094	0.4013
QR15	0.4584	0.054	0.4112	0.4506	0.066	0.421	0.4501	0.064	0.3874

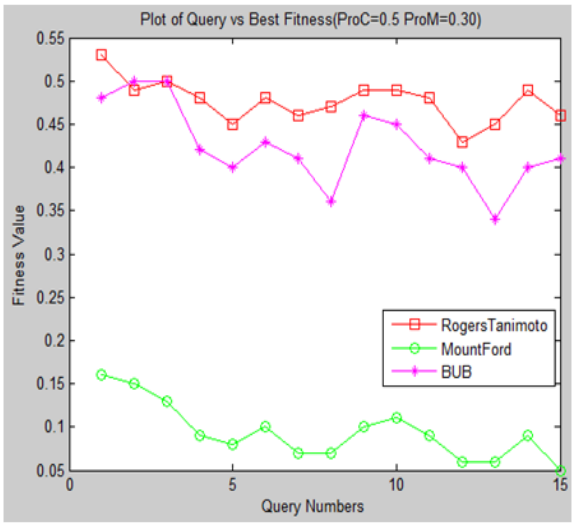

Fig. 11. Query vs Best Fitness for ProC=0.9, ProM=0.30 Using RT, MF and BUB

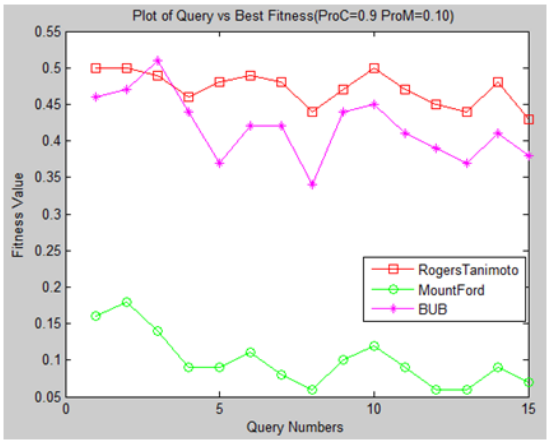

Fig. 12. Query vs best fitness for ProC=0.9, ProM=0.10 using RT, MF and BUB

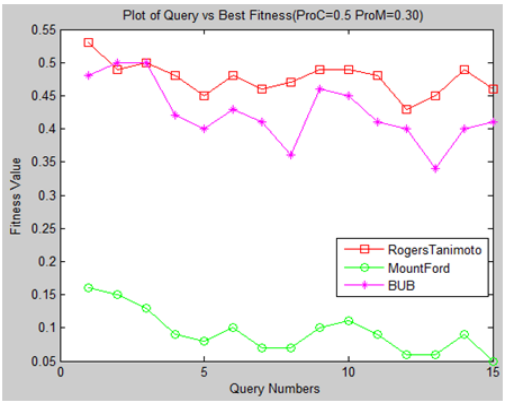

Fig.13. Query vs Best Fitness for ProC=0.5, ProM=0.30 Using RT, MF and BUB

4.3 Analysis

From tables 4 to 6 and figures 5 to 13 following observations can be noted:

- For each query Queryi(i=1 o 15) maximum average fitness of RT is better than MF and BUB
- Maximum average fitness for RT and BUB are very close to each other and they can be used for fusion.
- Performance of MF is very poor as compare to RT and BUB for all queries.
- Maximum average fitness is better for lower values of ProM and higher values of ProC. Best values are obtained at ProM=0.01 and ProC=0.9.
- Changing the values of ProM from 0.01 to 0.10 and 0.30 increases the fitness value for a fixed ProC but we does not see much difference for fitness value at ProM=0.10 and ProM=0.30.

5. CONCLUSIONS

The research work has calculated average maximum fitness value using RT, BUB and MF similarity coefficients. In the work ProC and ProM was varied from {0.5,0.7,0.9} and {0.01, 0.10, 0.30} respectively. It has been shown that average relevancy of retrieved document increases when PC is increased and ProM is decreased. In this way searching has been made easy, so more relevant document or web pages can be retrieved easily. Results show that average relevance of document increased up to 63%. In this way if focused crawler have key set with more relevancy then the retrieved data is more relevant for local collection of a search engine which improves the crawling performance.

References

[1]. B. Novak “A Survey Of Focused Web Crawling Algorithms”, Proceedings of SIKDD, pp. 55–58, 12-15 Oct 2004.

[2]. http://www.wikipedia.org/Web_Crawler

[3]. B.Klabbankoh, O.Pinngern. “applied genetic algorithms in information retrieval” Proceeding of IEEE ,pp.702-711,Nov 2004

[4]. S.S.Satya and P.Simon, "Review on Applicability of Genetic Algorithm to Web Search," International Journal of Computer Theory and Engineering, vol. 1, no. 4, pp. 450-455, 2009.

[5]. Shokouhi, M.; Chubak, P.; Raeesy, Z “ Enhancing focused crawling with genetic algorithms”Vol: 4-6, pp.503-508,2005.

[6]. www.sequentix.de/gelquest/help/distance_measures.htm

[7]. V.Consonni and R. Todeschini ,“New Similarity Coefficients for Binary Data”, Communications in Mathematical and in Computer Chemistry, pp.581-592, 2012

[8]. H.Wolda, “Similarity Indices, Sample Size and Diversity”, Oecologia-Springer-Verlag ,pp. 296-302,1981

[9]. M.A.Kauser, M. Nasar, S.K.Singh, “A Detailed Study on Information Retrieval using Genetic Algorithm”, Journal of Industrial and Intelligent Information vol. 1, no. 3, pp.122-127 Sep 2013.

[10]. http://en.wikipedia/wiki/Fitness_Proportionate_Selection

[11]. J.R. Koza, “ Survey Of Genetic Algorithms And Genetic Programming”, Proceedings of the Wescon, pp.589-595,1995

[12]. http://textalyser.net/

[13]. http://www.webconfs.com/keyword-density-checker.php.

[14]. V.Thada, V.Jaglan, “Use of Genetic Algorithm in Web Information Retrieval”, International Journal of Emerging Technologies in Computational and Applied Sciences, vol.7,no.3,pp.278-281, Feb,2014