Website Reorganization using Physics based Algorithms

Santosh Kumar (santoshg25@gmail.com)
ABES Engineering College
Aman Kumar Goyal
ABES Engineering College
Arjun Sharma
ABES Engineering College
Arpit Verma
ABES Engineering College

Research Article

Keywords: Web Mining, Genetic Algorithm, Big-Bang Big-Crunch, Galaxy Based Search Algorithm

Posted Date: April 20th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1562303/v1

Website Reorganization using Physics based Algorithms

Santosh Kumar, Aman Kumar Goyal, Arjun Sharma, Arpit Verma
Department of Computer Science and Engineering
ABES Engineering College, Ghaziabad, India
santoshg25@gmail.com, aman.18bcs1064@abes.ac.in , arjun.18bcs1087@abes.ac.in , arpit.18bcs1069@abes.ac.in

Abstract

This is the era of the internet and more than 50 percent of the population is dependent on it. Internet is collection of web pages which can be transported across it. Its size is very huge and increasing exponentially day by day. Now, around 30 percent of the overall content is duplicate. Finding relevant and required information from such a huge source of information is a very challenging task and finding top ranked results is a NP-hard problem. Majorly, the overall content can be categorized in three parts: content data, usage data and structure data. Web mining techniques and metaheuristic algorithms have been successfully applied in the literature to extract the useful information. This paper aims to provide ranking to the web links which could be utilized for restructuring of websites for business intelligence. Different features like keywords frequency, user's navigation behaviour like unique visitors, duration stayed, access frequency, hub, and authority information has been utilized for prioritizing top-T web links. These results may be useful in different applications that include web personalization, website reorganization, recommendation systems, search engine optimization, etc. Other than this, it will also help in improving the user's experience on websites. Based on different types of data, algorithms WLRGA, WLRBBBC, and WLRGbSA have been proposed for finding top-T web links and experimentally it has been seen that WLRGbSA is able to select top quality web links useful specially for website reorganization.

Keywords: Web Mining, Genetic Algorithm, Big-Bang Big-Crunch, Galaxy Based Search Algorithm

1. Introduction

Nowadays, every single piece of information is stored on the internet and with this the amount of data on the servers is increasing every day. People can now easily access the web for any information that is required. Web reorganization is needed for sorting the data present on the internet for the customers. It also assists with acquiring bits of knowledge into their inclinations and purpose through information, so that the users can be offered with custom-made encounters.

Web reorganization refers to the process of gathering visitor's interactions and navigation in a website in order to deliver them with more organized and accurate results on the web. The process of reorganization includes making the website more appealing, providing accurate results to the query and delivering only the relevant information to the user. This website can be any one of general website ranging from e-commerce to a stock trading website. Instead of providing a unit or wider experience, web reorganization permits the companies to give their users something special and distinctive according to their requirements and desires.

The main problem for every user who is accessing the internet is to find the useful information according to their requirements from the bulk of data stored on multiple servers across the globe. It costs the time and effort of the user to search for their requirements. If these web pages could be ranked according to the user's relevance then it would reduce the user's effort in finding the required web link. Other problem is that a large number of users do not move to the subsequent

page of the results shown. Many of the times the relevant data on the subsequent page is left untraversed by the user.

Optimization is a process of finding the minimum and maximum points based on some objective function. There are many different types of algorithm used for optimization problems. Optimization has find its place with many important applications and has been successfully implemented with many optimization algorithms. The stochastic algorithm is divided into two parts: heuristic and metaheuristic algorithm. Both the algorithms are used to find solutions by some guided trial and error method. Heuristic algorithms are mostly problem-dependent and different types of heuristics are defined for many problem statements. While metaheuristic algorithms almost have no prior assumptions about the problem and can integrate multiple heuristics inside it. Therefore, metaheuristics are applied to a wide range of problems which they treat as black-boxes. Popular examples of metaheuristic algorithms include Swarm Optimization (PSO), Ant Colony Optimization (ACO) and Harmony Search(HS).

The efficiency of an algorithm is mainly related to time and space, which includes speed and the rate of convergence in the direction of global optimum and the effectiveness is the measurement related to the solutions that is being returned by an algorithm. It mainly represents the capability of the algorithm given by the number of optimal solutions and the statistical analysis of the significance of the obtained result.

Among the various optimization algorithms present today, the modern metaheuristics's popularity is increasing day-by-day which is resulting in formation of new branch of optimization termed as metaheuristic optimization. Most of the metaheuristic algorithms are natured-based like the stimulated annealing, ant colony optimization and particle swarm optimization. The metaheuristic algorithm are mainly used in the field of optimization, data mining, machine learning, design, schedule and mining and many more such fields. The major problems in metaheuristic algorithm are that the metaheuristic algorithms are highly complex, non-linear and stochastic.

In this paper, the algorithms used are physics-based metaheuristic algorithm for web reorganization in order to provide optimized result to user. Generally recommendation systems are used by websites to provide optimized results but these systems use collaborating filtering techniques to find related content between customers having similar usage history. But these systems are not good enough to provide a real personalized experience. A few meta heuristic, physics based optimization algorithms have been proposed to provide personalized web experience in the most optimized way. The paper presents an existing review that belongs to the following -based metaheuristic algorithm:

1. Genetic algorithm
2. Galaxy based search algorithm
3. Big bang big crunch algorithm

This will give the reader a basic idea about how optimization algorithm works in web surfing. This research paper will also improve the knowledge of readers in the field of web structuring and reorganization.

1.1 Organization of Paper

Section 2 discusses about the related work. The proposed approach is discussed in Section 3 and the general idea of the Genetic Algorithm is given in subsection 3.1; Subsection 3.2 explains the general idea of the Big Bang Big Crunch Algorithm and the Center of Mass is explained in 3.2.1; The Galaxy Based Search Algorithm is explained in subsection 3.3. The problem representation is given section 4; The subsection 4.1 represents the candidate solution; Parameters for the

evaluation of the candidate solution is given in subsection 4.2 as keywordsFrequency in 4.2.1, accessFrequency in 4.2.2, numOfVisitors in 4.2.3, totalDuration in 4.2.4, numOfHubs in 4.2.5, numOfAuthorities in 4.2.6. The Fitness Function is given in 4.3. The algorithms are explained in detail in section 5; WLRGA is explained in subsection 5.1; WLRBBBC algorithm is explained in 5.2 and the WLRGbSA is explained in 5.3. Experimentation and Results has been discussed in Section 6. At last, Conclusions are given in Section 7 and the Future Directions in subsection 7.1.

2. Related Work

Web mining is a field that has been getting a lot of focus nowadays because of the increase in the data. Many researchers have tried to contribute a lot of studies and new algorithms for the same.

Chattopadhyay, et. al. (2022) in [1] tells that metaheuristic calculations are strategies formulated to take care of computationally testing enhancement issues effectively. He also tells that this part centers around meta-heuristic calculations demonstrated upon non-straight actual peculiarities having a substantial streamlining worldview, having shown impressive investigation and double-dealing capacities for such improvement issues which centers around a few famous material science based metaheuristic as well as depicting the hidden one of a kind actual cycles related with every calculation.

Zeidabadi, et. al. (2022) in [2] introduces a new algorithm called as Mutated Leader Algorithm(MLA) which is expected to provide quite optimal solution for various optimization problems. The principle thought in the proposed MLA is to update individual member of the algorithm in the search space which will be based on the guidance of the mutated leader. The MLA algorithm will include information of all the members of the population be it the best or the worst member of the population. The author had then compared its implementation result against eight other algorithms to analyze the performance of MLA algorithm to give the most optimal result.

Manik Sharma, et. al. (2021) in [3] presents a thorough examination of Nature-Inspired Meta-Heuristic Techniques for Feature Selection Problem which give an ideal arrangement by investigating and taking advantage of the whole inquiry space iteratively. A precise survey procedure has been utilized for amalgamation and investigation of one hundred and 76 articles. The rundown of nature-propelled meta-heuristic strategies and their variations alongside datasets, execution is likewise portrayed. Moreover, the itemized distribution pattern of meta-heuristic element determination approaches has likewise been introduced.

Dola Gobinda Padhan1, et. al. (2020) in [4], tells about the Gray Wolf Optimization which is generally used for PID optimization. It is a well known meta-heuristic algorithm for optimization problems and also compares gray wolf with big bang big crunch optimizations to provide us a better understanding of which algorithm is better in a particular scenario.

Amine Khalil (2019) in [5] tells about the simulated annealing method, its different variants, and the state of art of the simulated annealing method. Asymptotic convergence makes simulated annealing a powerful tool but the algorithm processes slow for some of the cooling schemes.

Demirol, et. al. (2018) in [6] compares the performance of various Physics Based Meta-Heuristic Optimization Algorithms. He also tells the aim of optimization process with objective function. Many different algorithms have been developed for optimization problems. In result of the quantity of expressed populace, size, run and cycle, after the base, greatest, standard deviation, and their mean qualities were laid out, their prevalence over one still up in the air.

Hossein Mousavi, et. al. (2017) in [7] explains a more general optimization algorithm that takes inspiration from the life of a comet. This algorithm is used to solve a variety of optimization problems. Analogous to how only a few comets persist while approaching the sun, it converges

to a high fitness function in a global maximum. Promising results in solving all the types of optimization problems are shown by this algorithm.

Yan Kang, et. al. (2017) in [8], presents a hybrid Big Bang Big Crunch algorithm which is used for optimization in the energy sector and is basically inspired from the Big-Bang and the Big-Crunch algorithm. This approach can be used in web optimization problems and is expected to give promising results.

Nazmul Siddiqui, et. al. (2017) in [9], tells us about the Chemical Reaction Optimisation (CRO). The algorithm is inspired from the process of different chemical reactions and is classified as a population based meta-heuristic algorithm. The general mechanism in any chemical reaction is the transformation of the reactants into products through a sequence of steps.

Can, et. al. (2015) in [10], talks about the optimization technique and has introduced all the possible metaheuristic optimization algorithms that are physics based. The author had also distinguished between the biology-based and physics-based algorithm and had also talked about the evolution of physics-based algorithm and their importance in the field of optimization.

Sariman, et. al. (2014) in [11], explains the application of the meta-heuristic algorithms in solving various optimization problems is discussed. These algorithms have been built by observing various natural phenomena and are being tested on different test cases by changing the input parameters. The change in the results of the algorithms can be seen by changing the input parameters for each optimization.

Sivanandam, et. al. (2008) in [12] describes about the genetic algorithm as a concise prologue optimization technique for solving various optimization problems in a easy way and its application in significant fields of optimization, for ex. Fuzzy, combinatorial and multi objective optimizations.

Jiao, et. al. (2006) in [13], explains about the Small-World Optimization Algorithm (SWOA) . As per the comparisons made with different algorithms like the Genetic Algorithm, the comparison results indicate that SWOA can be used to avoid the GA deceptive problem, enhance the diversity of the population and also avoid the prematurity of the population at a faster rate. Complex tasks can be more effectively solved using the SWOA.

H. Edwin Romeijn, et. al. (1994) in [14] explains about the simulated annealing algorithm and its usefulness in optimization problems by implementing an idealized version of simulating annealing called adaptive annealing. As the dimension of the problem increases, the number of the corresponding record values is also increased. A natural choice for the cooling schedule of a problem is also suggested by the algorithms of this class.

Richard Formato in [15], tells about a new deterministic metaheuristic algorithm which is inspired from the gravitational kinematics and is deterministic and multi-dimensional and uses the central force optimization technique. Different “probes” that are observed to “fly” and move under the gravitational pull and move through the decision space by providing analogy to masses. As per the particle motion under the influence of gravitational pull, different equations are developed for the positions, accelerations and probes.

H. Adeli, et. al.in [16] tells about various physics-based optimization algorithms that are currently being used, these are inspired by natural phenomena. Author has presented a detailed study and review of several physics-based algorithms in a detailed manner. Through this paper we get to know about various physics-based algorithms and we have identified a few algorithms that can be used for our optimization problem.

Shah-Hosseini in [17], presents a new nature inspired algorithm belonging to the class of meta heuristic algorithm and uses it to perform multilevel thresholding for segmentation of gray level

images. Galaxy based search algorithm imitates the spiral pattern arms of the galaxy to find the most optimum solutions, author uses GbSA for multilevel thresholding of images and then he compares the results based on the Otsu's criterion of an exhaustive search. The results indicate that GbSA proves to be highly efficient for the problem specified and it can be used for other optimization tasks.

Hamed Shah-Hosseini in [18], uses principal component analysis and uses a galaxy based search algorithm to explore the search space by imitating the arms of the galaxy and enhancing it by introducing chaos to escape the local optimum. This paper introduces and modifies the galaxy based search for PCA estimation. We get to explore GbSA more and we get a better understanding of how it can be modified and can be made more suitable as per our problem statement.

The writing work which has previously done utilizes the page positioning, information mining and AI calculations to recover data. The creators utilize the web information as structure information or the substance information while certain creators likewise utilize different calculations like hereditary calculations, mimetic calculations, and so forth to work on the outcome. The creators have moreover utilized a few arbitrary calculations however the proposed calculations utilize the boundaries which are of most extreme significance to rearrange the sites in light of client's question. The utilization of these boundary assists the proposed calculations with contrasting and other randomized calculations which were at that point proposed and taking a gander at the outcomes, proposed calculations were showing better execution which likewise help in adding to the exploration space.

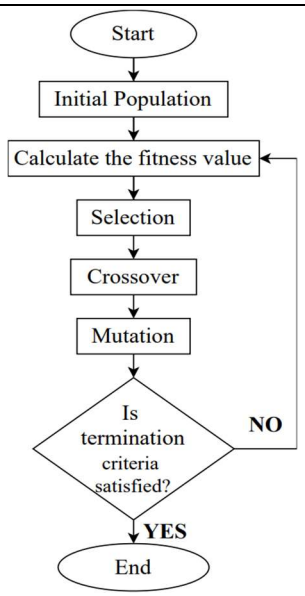
3. Proposed Approach

It has been found that the evolutionary genetic algorithm has been successfully applied to solve optimization problems like the Voyage Optimisation[20] , Crack detection in bulk superconductor[21], two-dimensional Knapsack Packing problem[22], Electric vehicle charging stations emplacement[23], Moonpool dimensions and position optimisation[24], Energy Saving of Electric Refrigerated Truck[25]. The BB-BC algorithm has been applied on wide variety of problem like passive building design[19], Construction-Engineering Design Optimisation[26], Sliding Mode Control[27], Voltage and Frequency Regulation[28], Diminution of LTI systems[29], Design of Schwedlerand Ribbed Domes[30]. The GbSA algorithm has been applied on a wide variety of problems like Minimize Real Power Losses[31], Object-tracking[32].

This paper aims to provide web link ranking algorithms. The algorithms are evaluated on different values of the dataset and then evaluated based on their results. These results will be beneficial in examining different use cases of the algorithms and solving other problems.

3.1 Genetic Algorithm

This is a classical evolutionary algorithm inspired from random selection. Uncertain or random changes are applied to the solutions to generate the new ones. Darwin's theory of evolution is the core idea behind the Genetic Algorithm. Genetic Algorithm moves forward by making slight changes until the algorithm is not converged. The convergence of the algorithm occours when the fitness scores remains unchanged for several consecutive generations. Figure 1. shows the flowchart and the Genetic Algorithm.



Step 1 : Initialize t = 0

Step 2 : Initialize population P(t = 0)

Step 3 : Calculate fitness of the population

Step 4 : Evaluate the population P(t = 0)

Step 5 : while not terminated do

Step 6 : Pp(t) = P(t) , select Parents

Step 7 : Pε(t) = reproduction(Pp)

Step 8 : mutate(Pε(t))

Step 9 : evaluate(Pε(t))

Step 10 : P(t+1) = build next generation from (Pε(t), P(t))

Step 11 : t = t + 1

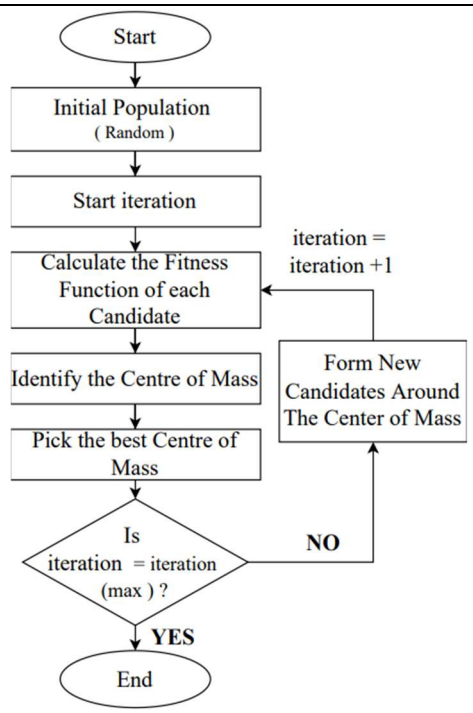
Step 12 : end while

Fig.1 Genetic Algorithm

3.2 Big Bang Big Crunch Algorithm

This algorithm is inspired from the theory proposed behind the beginning and end of the universe, The Big Bang and Big Crunch. In 1927, an astronomer named Georges Lemaître said that the universe started from a single point and with time it stretched and expanded to become as big as it is now. Further from more observations it was noticed that our galaxy is still expanding. This whole theory about the formation of the universe was termed The Big Bang. Moving forward with this theory of the Big Bang, there is also a hypothetical theory as the Big Crunch. This theory says that the expansion of the universe will eventually reverse after a point of time and the complete universe will collapse as the ultimate fate.

In context with the internet and the problem of ranking the web pages resembles this theory. In the Big Bang phase, we have the complete bunch of information that is there on the internet in the form of different websites, urls, resources that all have emerged from one point and with the advancements of the internet and the scaling it is getting scaled up day by day. This piling up of information can be termed as the Big Bang phase here. Moving further with the Big Crunch phase, this can be generalized as the process that is required to reach a final destination or the result starting from that bulk of information. To reach a final destination is same as finding the best T web links from a thousand of links available up there. Figure 2. shows the flowchart and the Big Bang Big Crunch Algorithm.



Step 1 : *Initializing the variables:*

- *Nr : Normal Random Number*
- *Pp : Population Size*
- *Ub : Upper Boundary*
- *max_iteration : maximum number of iterations*
- *it : iterator*

Step 2 : *Generate the initial population Xi, of size Pp*

Step 3 : *The fitness value for each candidate of the population is calculated*

Step 4 : *Calculate the center of mass xc*

Step 5 : *Generate new solutions around the center of mass*

Step 6 : *it = it + 1*

Step 7 : *return to step 3, until stopping criteria, (it = max_iteration)*

Fig. 2 Big Bang Big Crunch Algorithm

The generation of dataset for this algorithm is the Big Bang phase. In this the unique Url Id's can be taken as the discrete and widely distributed masses in the Galaxy. Now in the Big Crunch phase, these distributed masses will start coming closer to their Centre of Mass (COM). The Centre of Mass for any two particle system having equal masses separated by any distance x(say) lies halfway between them[33].

3.2.1 Center of Mass

In a distributed system of masses, the density of mass at a particular point varies with distance x and length L is represented as given below:

$\lambda(x) = \lambda_\circ \left(\frac{x}{L} + 1\right)$ where, λ is the mass density, $\lambda_\circ$ is the mass density when x = 0 and L is the total length.

Now, for an object whose length is l=0 at start and l = L at the end, the centre of mass can be calculated as

$$\lambda(x) = -\left(\frac{x}{L} + 1\right) x$$

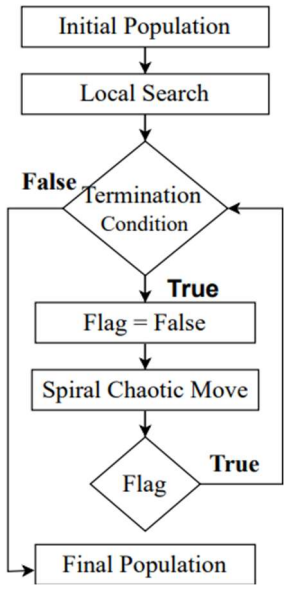
$$x = \frac{\int_{x=0}^{x=L}\left[\lambda_\circ \left(\frac{x}{L}+1\right)\right]x}{\int_{x=0}^{x=L}\left[\lambda_\circ \left(\frac{x}{L}+1\right)\right]dx} = \frac{\lambda_\circ \int_{x=0}^{x=L}\left(\frac{x^2}{L}+x\right)dx}{\lambda_\circ \int_{x=0}^{x=L}\left(\frac{x}{L}+1\right)d} = \frac{5}{9}L$$

From the above formula it is observed that the centre of mass does not depend upon the density.

3.3 Galaxy Based Search Algorithm

The GbSA is inspired from nature and is a continuous search algorithm. The algorithm searches in its surroundings through imitating the spiral form of arm as the spiral galaxies. This spiral movement basically helps the local search to escape from the local optimum. The results obtained after using the spiral movement approach of the GbSA algorithm are then further

adjusted using a local search algorithm. The local search is done to find other candidates with better fitness scores. Figure 3. shows the flowchart and the Galaxy Based Search Algorithm.



Gen =1 , maxG = 1000
***Step 1**: Generate initial seed points (candidate solution)*
***Step 2** : while (gen < maxG) do*
***Step 3** : S` = perform spiral chaotic move (mutate 1st Ith gene of S)*
Perform local search using Hill Climbing
***Step 4** : Find N neighbors of S`*
***Step 5** : if (fitness(N[i]) > fitness(S` ) ) S` = N[i]*
***Step 6** : goto Step 3*
***Step 7** : gen=gen+1*

Fig. 3 Galaxy Based Search Algorithm

4. Problem Representation

Our major problem is to find out the best top-k web pages which are suitable for that specific query. Inputs are represented by the web pages which consist of some selected properties. Web pages are addressed by their URLs which are represented as a string and so they can't be pre-processed. So, unique Id's are given to the url's as shown in Table 1.

Table 1. Mapping of URL ID to unique IDs

Web Users	WebUrl	UniqueId
User1	192.168.31.26	1
User2	10.21.30.15	2
User3	192.158.30.56	3
User4	100.0.1.100	4
..	..	…
UserN	20.98.56.78	N

4.1 Candidate Solution Representation

The unique url Id's are now mapped with the dataset features of the respective url as shown in Table 2.

Table 2. Mapping of unique Ids to their feature values

WebUsers	Unique Id	WebUrl	AccessCount	UniqueVisitor	StayedTime	Hubs	Authorities
User1	1	192.168.32.25	43	129	200	7	9
User2	2	192.168.32.52	23	27	120	6	8
User3	3	10.21.30.15	18	75	65	9	5
User4	4	10.0.0.10	48	38	48	12	12
..	…	..	..	…	…	14	6
UserN	N	10.21.70.93	30	73	90	6	17

The dataset values can now be accessed and operated upon by using the unique Id's. Table 3. shows the dataset that can now be pre-processed easily.

Table 3. Arranging the dataset features

Parameters/Urls	Url1	Url2	Url3	Url4	Url5	Url6	Url_i	Url10000
Document ids	1	2	3	4	5	6	…	10
Access frequency	113	126	436	145	144	152	…	149
Visitors	4569	4374	127	836	5310	2254	…	476
Duration Stayed	2138	2300	1578	1845	345	2451	…	1597
Hubs	44	5	89	13	9	3	…	17
Authorities	1	65	11	17	6	19	…	21

The example of a candidate solution having top 10 web links is shown in table 4.

Table 4. Showing an example of a candidate solution

Candidate Solution									
Top-10 web Pages									
Randomly generated Unique Ids									
5	10	7	19	25	16	43	61	77	21

4.2 Parameters for evaluation of candidate

4.2.1 **keywordsFrequency:** The number of times a particular keyword occours in a web document. It is the sum of total occurences of all the unique keywords of a query present in any web document.

4.2.2 **accessFrequency**: The number of times a particular website is visited by users. Whenever any user open a particular web document, it is being noted down as one visit and the total number of visits is the access frequency.

4.2.3 **numOfVisitors**: The number of unique users that visit the website. It counts only the unique visitors who have opened that particular web document.

4.2.4 **totalDuration**: The total time duration for which a particular website is looked upon or opened by a user. This is calculated in milli seconds.

4.2.5 **numOfHubs**: The number of hyperlinks in a particular website. The total number of different websites that this particular webpage routes to or forwards its users for more information.

4.2.6 **numOfAuthorities**: The number of different websites that route to this website. This shows the credibility of a particular website. More the number of authorities, more is the value of that particular web page.

The maximum values of dataset features that are used to evaluate the fitness scores are listed in Table 5.

Table 5. Maximum values of different dataset features

Parameters	Interpretation	Value
maxAccessFrequency	Maximum access frequency	1000
maxVisitors	Maximum visitors	500
maxDuration	Maximum duration stayed by visitor on any web document	1800000
maxHubs	Maximum number of links pointed by any document	50
maxAuthorities	Maximum number of links pointed to any document	50

maxKeywordValue	Maximum number of keyword found in any document	20
maxKeywordFrequency	Keyword Frequency	50

4.3 Fitness Function

The fitness function used in this algorithm is as

$$fitness_{score} = c1 \times \sum_{i=1}^{n} AccessFreq_i + c2 \times \sum_{i=1}^{n} Duration_i + c3 \times \sum_{i=1}^{n} numOfVisitors_i + c4 \times \sum_{i=1}^{n} numOfHubs_i + c5 \times \sum_{i=1}^{n} numOfAuthorities_i + c6 \times \sum_{i=1}^{n} freqOfKeyword_i$$

where, $c1 = 1$, $c2 = \frac{maxAccessFreq}{maxDuration}$, $c3 = \frac{maxAccessFreq}{maxVisitors}$, $c4 = \frac{maxAccessFreq}{maxHubs}$,

$c5 = \frac{maxAccessFreq}{maxAuthoritie}$, $c6 = \frac{maxAccessFreq}{maxKeywordFreq}$

5. Proposed Algorithms

5.1 Web Links Ranking using Genetic Algorithm, WLRGA

The important parameters used in this algorithm are the crossover rate that varies from 0.5 to 0.9 and mutation rate that varies from 0.5 to 0.9 and the algorithm starts by initializing a population where pop size varies as 100,150, 200 and 250. A sequence of steps is followed until convergence is reached which is when the whole population becomes same for several generations and these steps are selection, crossover and mutation. In selection, binary tournament selection method has been used and child population is generated from these selected candidates and then the child population is mutated according to mutation rate and then this process is repeated till convergence.

```
Procedure WLRGA(N, K, CR, MR)
Initalization:
Population size, N∈{100, 150, 200, 250}
Number of top ranked web documents, K∈{5, 6, 7, 8, 9, 10}
Crossover rate, CR∈{0.5, 0.6, 0.7, 0.8, 0.9}
Mutation rate, MR∈{0.5, 0.6, 0.7, 0.8, 0.9}
Evaluation Function
fitness_value(chromosome)
Method:
Generate initial population of Top-K web pages of size N, randomly
For i = 1 to N
        For j = 1 to k
                POP[i][j] = random(MAX_URL_ID)
        End for
End for
Select the set of chromosomes from Pop[] for crossover using binary tournament selection
        Child_pop = []
        For i in range(pop_size)
                best, second_best = POP[random(0, pop_size)], POP[random(0, pop_size)]
                r = random(0, 1)
                if r < crossover_rate
                        Child_pop.append(best)
                else
                        Child_pop.append(second_best)
```

```
        POP = Child_pop
Perform crossover among selected Top-K web pages
        Fori=1 to pop_size/2
                (Parent1, Parent2) = POP[random(0, pop_size)], POP[random(0, pop_size)]
                (Child1, Child2)=cyclic_crossover(Parent1, Parent2)
                POP  (Off1, Off2)
        End for
Perform mutation
        Set mutation rate, MR
        R  = pop_size x MR
        For i=1 to R
                For j in range(num_of_mutations)
                        page = get_page_id()
                        while page is in POP[i]
                                page = get_page_id()
                        POP[i][j] = page
                End for
        End for
```

The parameter values that varies in the algorithm are listed in Table 6.These are the standard values that are used to generate results of this algorithm. These parameters are varied so that the algorithms can be evaluated in different situations and the results can be observed.

Table 6. Parameter Settings for WLRGA

Parameter	Value
Initial Population Size	100, 150, 200, 250
Crossover rate	50%, 60%, 70%, 80%, 90%
Mutation rate	0.05, 0.10
Size of Chromosome(Web Document)	5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15
Selection method (selection of population for crossover	Binary Tournament Selection
Crossover type	Cyclic crossover

Table 7. shows the fitness score values obtained when the algorithm is executed for different parameter values.

Table 7. Selection of suitable crossover and mutation rate

Popsize=100, SM=BTS, Top-T=10				
Crossover Rate(%)	Fitness Score			
	MR=0.05	MR=0.05	MR=0.10	MR=0.10
50	34989	34818	40114	30416
60	39794	36883	35189	35754
70	39865	39345	36551	35245
80	33744	36422	35647	36858
90	35149	32499	35583	39095
Average=	36708.2	35993.4	36616.8	35473.6
Net Average =	36350.8		36045.2	

The algorithm is implemented over different crossover rates {50,60,70,80,90} and different mutation rates {0.05 , 0.1}. The results show that the algorithm gives the best results for 0.05 %

mutation rate when the crossover rate is 70%.Now, for all further analysis of this algorithm the crossover rate will be 70% and the mutation rate will be 0.05%.
Table 8. shows the fitness scores when the algorithm is executed for different population sizes and different Top T web links.

Table 8. Finding suitable Population Size for WLRGA

CR=70, MR=0.05, SM=BTS				
Top-T Web Links	Fitness Score, WLRGA			
	PopSize=100	PopSize=150	PopSize=200	PopSize=250
5	18741	21447	18248	20559
6	23108	23293	19497	22826
7	26656	25715	28598	24241
8	29611	27228	32889	26793
9	32874	37158	33231	35061
11	37532	38449	42074	45042
12	41365	44572	44925	46899
13	50297	45051	53092	45260
14	44899	54424	53507	49188
15	55715	59251	53963	55440
Average =	36079.8	37658.8	38002.4	37130.9

Now, upon taking the average of the fitness scores for different population sizes, it is clear that the algorithm shows best results when the population size is 200.

5.2 Web Links Ranking using Big-Bang Big-Crunch, WLRBBBC

This algorithm is inspired from the theory of the evolution and the destruction of universe. It works in two phases, the big bang phase and the big crunch phase. In the big bang phase, the population is generated which is then used to calculate the center of mass. The centre of mass can be described as the central value around which the complete mass is evenly distributed. In the second phase, new population is generated around that center of mass and this process keeps on repeating until convergence is reached which is when the results remains constant for several consecutive generations.

```
Initialization
Population size, n ∈{100, 150, 200, 250}
Number of top ranked documents, k ∈{5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15}
Pop_limiter = 200
Evaluate Fitness
fitness_value(chromosome)
Method
Generate initial population
For i = 1 to N
        For j = 1 to k
                POP[i][j] = random(MAX_URL_ID)
        End for
End for
Find center of mass from the population
Center_of_mass = 0
For i in population
        Current_fitness += Fitness of array i
        Center_of_mass += 5*(Current_fitness)/9
Center_of_mass /= len(Population)
Find the instance nearest to center of mass value
```

```
minDif = INT_MAX
        For instance in population:
                Dif = fitness of instance - center_of_mass
                if(dif < 0)
                        Continue
                if(dif < minDif)
                        Res = instance
                        minDif = dif
            return Res
Generate new population from center of mass
New_population = []
For instance in population:
New_center = center_of_mass + (pop_Size*random(0,Pop_limiter))/iteration
for i in population:
        if(Fitness of i - center) < 100):
            New_instance =  i

        New_population.append(new_instance)
Return New_population
Assign new population to population
Population =  new_population
Increment iterator and repeat till convergence
```

The parameter values that varies in the algorithm are listed in Table 9. These are the standard values that are used to generate results of this algorithm. These parameters are varied so that the algorithms can be evaluated in different situations and the results can be observed.

Table 9. Parameter settings for WLRBBBC

Parameter	**Value**
Initial Population Size	100, 150, 200, 250
Size of Chromosome(Web Document)	5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15
Pop_limiter	2000

Table 10. shows the fitness scores when the algorithm is executed for different population sizes and different Top T web links.

Table 10. Finding suitable Population Size for WLRBBBC

Pop_limiter=2000				
Top-T Web Links	Fitness Score, WLRBBBC			
	PopSize=100	PopSize=150	PopSize=200	PopSize=250
5	21496	22268	23729	30274
6	22834	23701	24544	31157
7	25579	28241	25855	31981
8	31957	29463	33587	33776
9	37323	37113	34967	35081
11	44298	44211	42797	43190
12	50700	45923	48287	47134
13	48491	51033	48118	50349
14	51669	52048	54220	54112
15	58394	56218	59315	59871
Average=	39274.1	39021.9	39541.9	41692.5

Now, upon taking the average of the fitness scores for different population sizes, the data shows that the algorithm shows best results when the population size is 250.

5.3 Web Links Ranking using Galaxy based Search Algorithm, WLRGbSA

This algorithm is inspired from the spiral pattern of galaxy. The algorithm imitates the spiral movement through the subsequent generations. The algorithm contains two major phases, one is spiral chaotic move and other is the local search. In spiral chaotic move, an initial candidate solution is taken and then the algorithm tries to find a better solution while moving in spiral manner across the data. In local search phase, several neighbours around the candidate solution are generated and the best solution is searched using the hill climbing algorithm. This process keeps on repeating until convergence is reached which is when the results remains constant for several consecutive generations.

```
Initialization
Population size, n ∈{100, 150, 200, 250}
Number of top ranked documents, k ∈{5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15}
Num_Of_iter_for_Spiral_move = 100
Evaluate Fitness
fitness_value(chromosome)
Method
Generate initial candidate solution i.e. SG_instance
chromosome = []
For i = 1 to k
        t = random(MAX_URLs)
        if  t not in done:
                chromosome.append(t)
        End if
End for
Find new SG instance using spiral chaotic move
i = (i+1)%k
while (iter< Num_Of_iter_for_Spiral_move):
        ran = random(MAX_URLs)
        while(ran in SG_instance)
                ran = random(MAX_URLs)
        SG_instance[i] = ran
Do local search by generating n neighbours and find a solution with better ftness among that population
neighbour = []
for i in range(len(Sg_instnace)
        for j in range(len(SG_instance)
                if j!= i
                        SG_instance[j] = random(MAX_URLs) not in SG_instnace
                neighbour.append(SG_instance)
SG_instance = max in neighbour
```

The parameter values that varies in the algorithm are listed in Table 11.These are the standard values that are used to generate results of this algorithm. These parameters are varied so that the algorithms can be evaluated in different situations and the results can be observed.

Table 11. Parameter Settings for WLRBBBC

Parameter	Value
Initial Population Size	100, 150, 200, 250
Size of Chromosome(Web Document)	5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15
Num_Of_iter_for_Spiral_move	100
Local Search	Hill Climbing Algorithm

Table 12. shows the fitness scores when the algorithm is executed for different population sizes and different Top T web links.

Table 12. Finding suitable Population Size for WLRGbSA

#iter_Spiral_move=100, Local Search=HCA				
Top-T Web Links	Fitness Score, WLRGbSA			
	PopSize=100	PopSize=150	PopSize=200	PopSize=250
5	29894	30028	30072	31244
6	33556	34821	35136	36790
7	37509	37414	37220	38050
8	39117	39111	44085	41178
9	44111	44648	44686	44577
11	52962	51303	51956	51971
12	53079	56531	54515	60085
13	57449	59250	57884	58665
14	65272	64471	62932	59912
15	63608	68090	65800	69252
Average =	47655.7	48566.7	48428.6	49172.4

Now, upon taking the average of the fitness scores for different population sizes, the data shows that the algorithm shows best results when the population size is 250.

6. Experimentation and Results

The algorithms have been implemented in python language using the Jupyer IDE and their execution has been done on a Ubuntu 20.04.4 LTS machine with 8 GB RAM , Intel Core i5 7th gen processor with a speed of about 3.4 Ghz and supported by an integrated NVIDIA GeForce MX130 , 2GB graphics card.

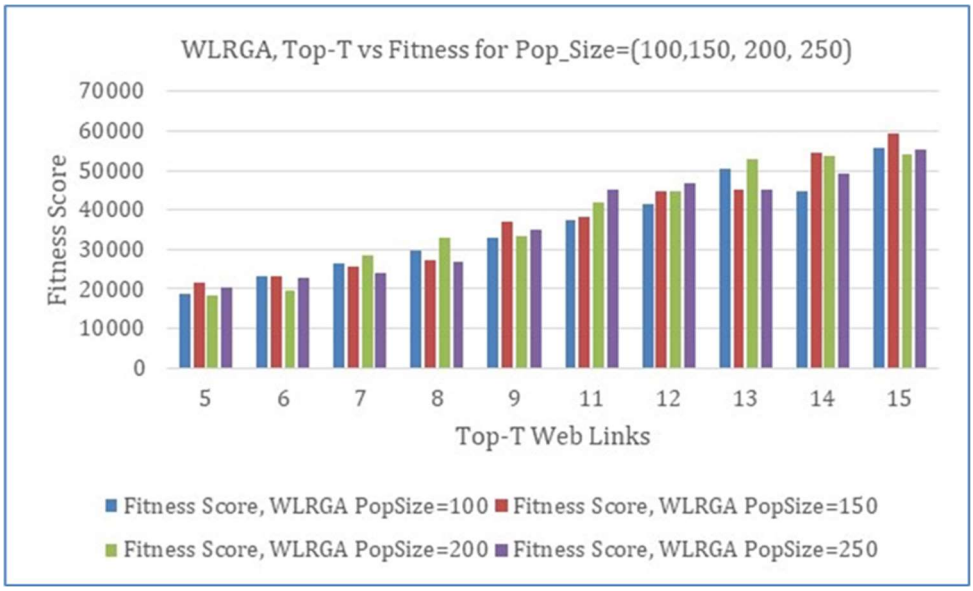


Fig. 4(a) WLRGA, Top-T vs Fitness for Pop_size = {100,150,200,250}

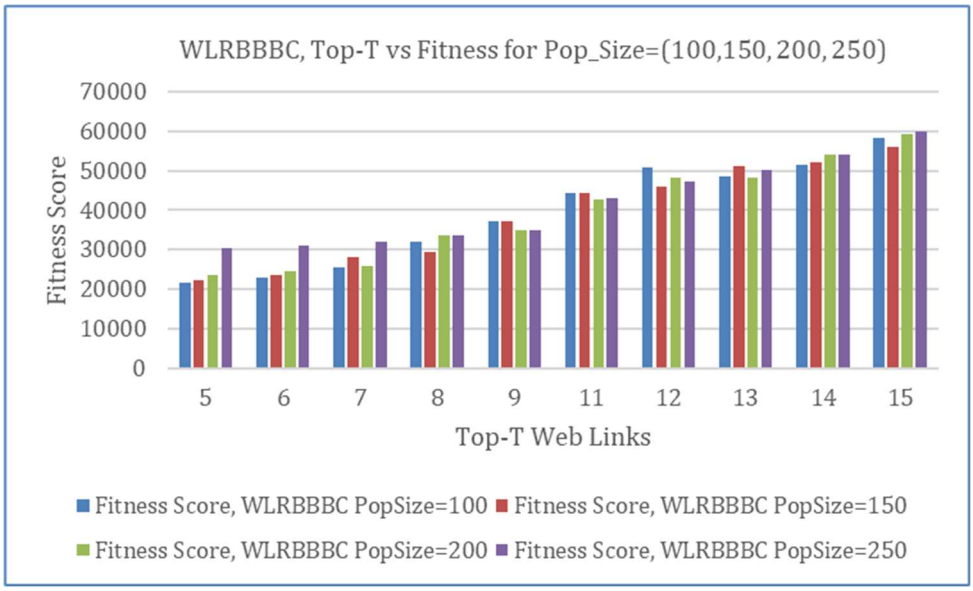


Fig. 4(b) WLRBBBC, Top-T vs Fitness for Pop_size = {100,150,200,250}

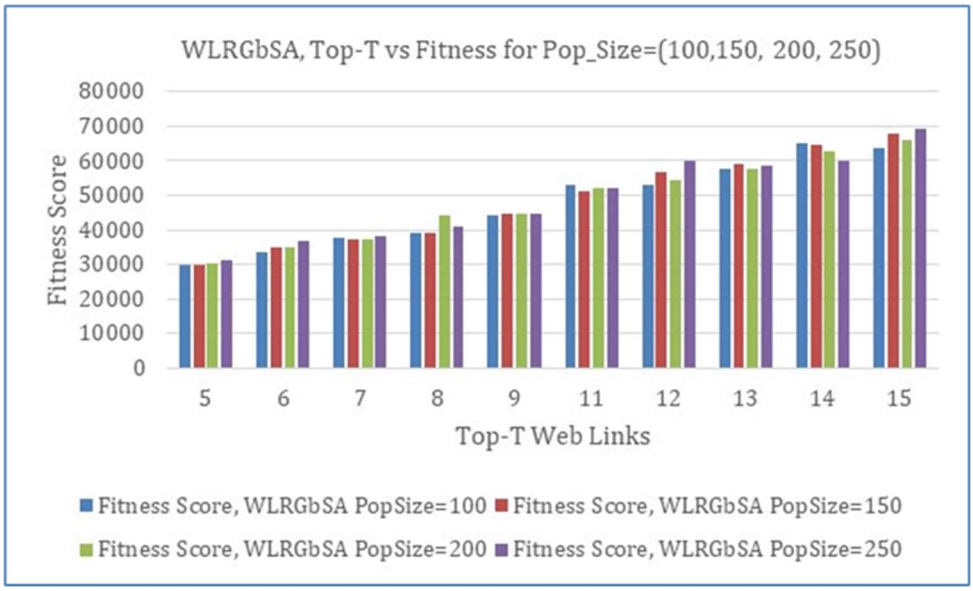


Fig. 4(c) WLRGbSA, Top-T vs Fitness for Pop_size = {100,150,200,250}

The above graphs shows a comparative study between the Top-T weblinks vs Fitness scores for different Pop_size {100,150,200,250}. Fig. 4(a) represents the WLRGA , Fig. 4(b) represents the WLRBBBC and Fig. 4(c) represents the WLRGbSA analysis graph. From the graphs, this can be seen that the Fitness scores for all population sizes varies but the difference is not much for the same Top-T web links. Table 13. represents the fitness scores when the three algorithms are compared as Top-T web links vs Generations.

Table 13. Comparing Iterative Results among the three Algorithms

Generations	Fitness Score(GA)						Fitness Score(BB-BC)						Fitness Score(GbSA)					
	Top-5	Top-6	Top-7	Top-8	Top-9	Top-10	Top-5	Top-6	Top-7	Top-8	Top-9	Top-10	Top-5	Top-6	Top-7	Top-8	Top-9	Top-10
1	16761	20253	23733	27208	31254	34017	15698	19316	21702	21533	28598	32347	22394	19062	22260	28362	32905	36982
51	15474	23641	26386	29416	33282	34314	20696	21969	25397	30762	36216	39811	29760	31658	36399	39478	43972	47781
101	15018	24905	26394	28267	34154	34149	20696	23994	26453	32227	36216	39811	29760	32497	36684	39516	44315	48059
151	15419	25067	26219	28972	34375	34278	20696	23994	26453	32227	36216	39811	29820	32621	36684	39777	44418	48593
201	15389	25065	26275	28794	34375	35059	20696	23994	26453	32227	36216	39811	29997	32741	36702	39777	44557	48593
251	15762	24855	26215	28214	34375	35085	20696	23994	26453	32227	36216	39811	30023	32911	36990	39872	44557	48593
301	16391	24879	26215	29002	34375	35200	20696	23994	26453	32227	36216	39811	30023	32911	36990	39943	44557	48689
351	16391	24889	26345	29004	34375	35269	20696	23994	26453	32227	36216	39811	30027	32911	37056	39943	44557	48689
401	16391	24863	26242	29725	34375	34997	20696	23994	26453	32227	36216	39811	30027	32911	37135	39954	44636	48920
451	16391	24884	26222	30251	34375	34989	20696	23994	26453	32227	36216	39811	30028	32911	37135	39954	44636	48920
501	16391	25032	26393	30363	34392	34989	20696	23994	26453	32227	36216	39811	30028	32911	37183	39954	44636	48920
551	16391	25792	26233	30286	34406	34989	20696	23994	26453	32227	36216	39811	30028	32911	37183	39954	44636	48920
601	16391	25902	26222	30249	34375	34989	20696	23994	26453	32227	36216	39811	30028	32911	37183	39974	44636	48920
651	16391	26473	26256	30249	34375	34989	20696	23994	26453	32227	36216	39811	30072	32911	37183	40022	44636	48920
701	16391	26473	26255	30249	34382	34989	20696	23994	26453	32227	36216	39811	30072	32911	37183	40022	44647	48920
751	16391	26473	26375	30252	34862	34989	20696	23994	26453	32227	36216	39811	30072	32911	37183	40081	44647	48920
801	16391	26473	26773	30249	35088	34989	20696	23994	26453	32227	36216	39811	30072	32911	37183	40081	44647	48920
851	16391	26473	26777	30249	35088	34989	20696	23994	26453	32227	36216	39811	30072	32911	37183	40081	44651	48965
901	16391	26473	26782	30249	35088	34993	20696	23994	26453	32227	36216	39811	30072	32911	37183	40081	44651	48965
951	16391	26473	26769	30249	35088	34989	20696	23994	26453	32227	36216	39811	30072	32911	37183	40081	44651	48965
1001	16391	26473	26769	30249	35088	34989	20696	23994	26453	32227	36216	39811	30072	32911	37183	40081	44651	48965

This data shows how fast an algorithm convers for different values of Top-T web links at a population size of 100. Along with the convergence rate, the fitness scores also signifies the performance of the algorithm.

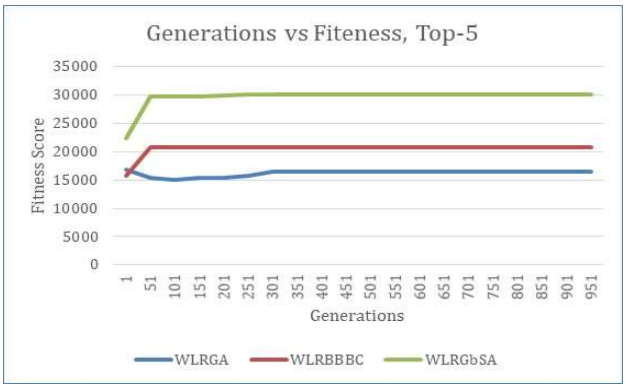


Fig. 5(a) Generations vs Fitness, Top-5

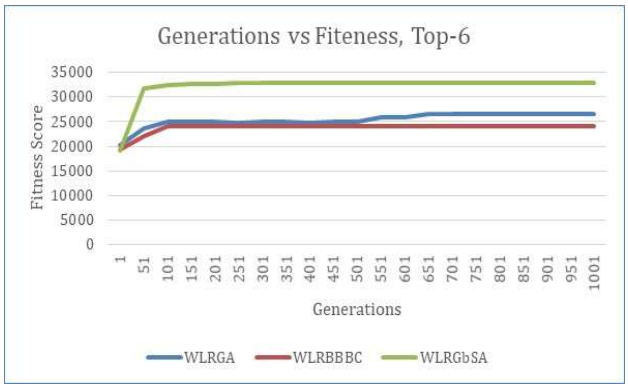


Fig. 5(b) Generations vs Fitness, Top-6

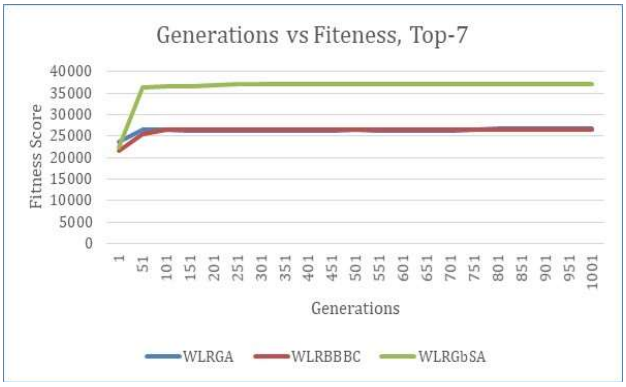


Fig. 5(c) Generations vs Fitness, Top-7

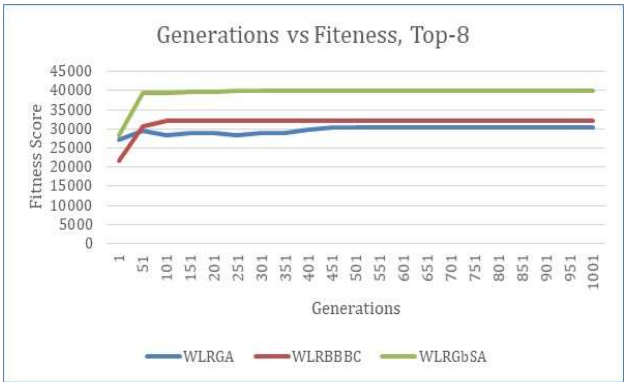


Fig. 5(d) Generations vs Fitness, Top-8

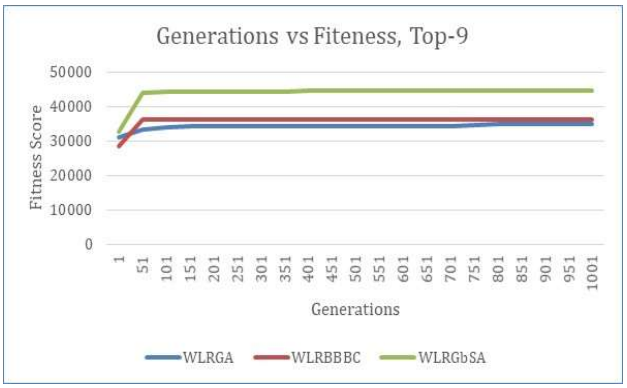


Fig. 5(e) Generations vs Fitness, Top-9

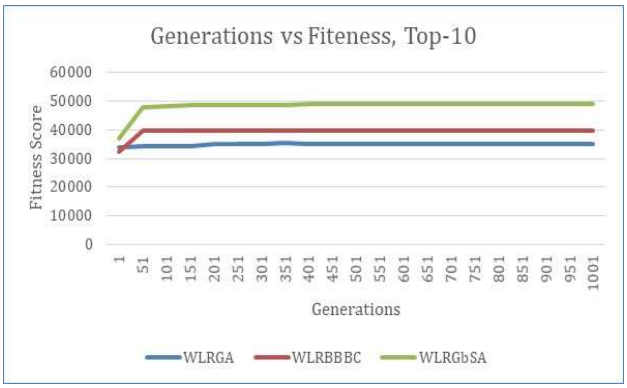


Fig. 5(f) Generations vs Fitness, Top-10

The above Figures represent the graphical view of the data in Table 13. These graph simply summarizes the performance of the algorithms for different Top-T web links. The above figures compares the algorithms for Generations vs Fitness Scores for different values of Top-T web links. Figure 5(a) for Top-5, Figure 5(b) for Top-6, Figure 5(c) for Top-7, Figure 5(d) for Top-8, Figure 5(e) for Top-9 and Figure 5(f) for Top-10. Table 14. gives the comparative analysis among these algorithms for different population sizes vs Top-T web links.

Table 14. Comparative analysis for different population size

Top-T Web Links	Popsize=100			Popsize=150			Popsize=200			Popsize=250		
	Fitness Score											
	GA	BB-BC	GbSA	GA	BB-BC	GbSA	GA	BB-BC	GbSA	GA	BB-BC	GbSA
5	18741	21496	29894	21447	22268	30028	18248	23729	30072	20559	30274	31244
6	23108	22834	33556	23293	23701	34821	19497	24544	35136	22826	31157	36790
7	26656	25579	37509	25715	28241	37414	28598	25855	37220	24241	31981	38050
8	29611	31957	39117	27228	29463	39111	32889	33587	44085	26793	33776	41178
9	32874	37323	44111	37158	37113	44648	33231	34967	44686	35061	35081	44577
11	37532	44298	52962	38449	44211	51303	42074	42797	51956	45042	43190	51971
12	41365	50700	53079	44572	45923	56531	44925	48287	54515	46899	47134	60085
13	50297	48491	57449	45051	51033	59250	53092	48118	57884	45260	50349	58665
14	44899	51669	65272	54424	52048	64471	53507	54220	62932	49188	54112	59912
15	55715	58394	63608	59251	56218	68090	53963	59315	65800	55440	59871	69252

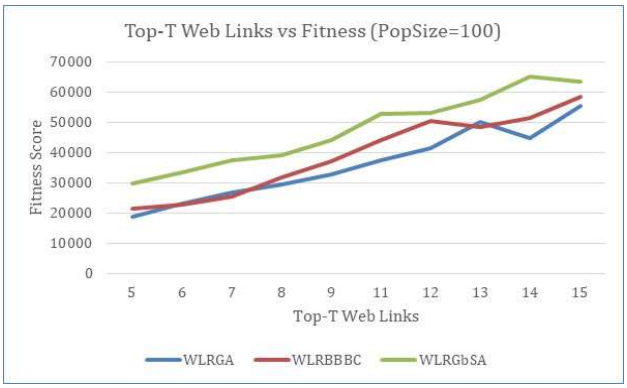


Fig. 6(a) Top-T Web Links vs Fitness (PopSize=100)

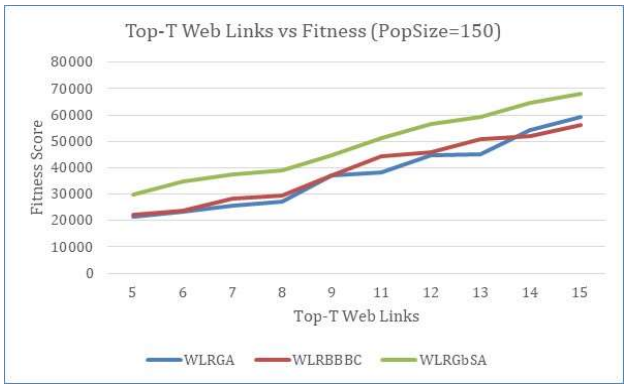


Fig. 6(b) Top-T Web Links vs Fitness (PopSize=150)

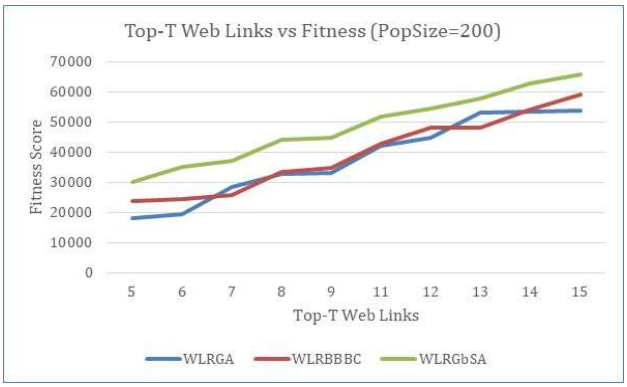


Fig. 6(c) Top-T Web Links vs Fitness (PopSize=200)

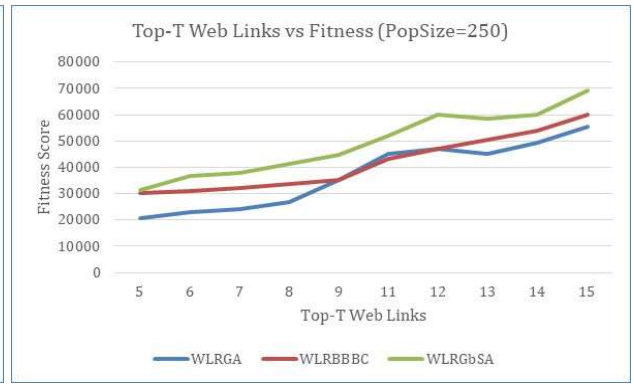


Fig. 6(d) Top-T Web Links vs Fitness (PopSize=250)

The graphs in figure 6(a), figure 6(b), figure 6(c) and figure 6(d) shows the performance of the algorithms and from the graphs it can be easily seen that the WLRGbSA performs best in all given conditions of population sizes and Top-T web links.

Further ANOVA test has been performed to measure the improvement in the performance of algorithms with respect to each other among WLRGA, WLRBBBC, and WLRGbSA. Results are shown in Table 15 and it is proved from the tests that the algorithm WLRGbSA is able to select better quality Top-T web links in comparison to other two algorithms. The algorithm WLRBBBC shows almost similar performance as WLRGA. it can be clearly seen that the improvement in selecting Top-T web links by WLRGbSA is significant in all cases because p-value is < 0.00001. The results are significant at $p<0.05$.

Table 15. ANOVA analysis for validation of results

Summary of Data for Top-5 web links					
	Treatments				The *f*-ratio value is 715.34001. The *p*-value is < .00001. The result is significant at *p* < .05.
	WLRGA	WLRBBBC	WLRGbSA	Total	
N	21	21	21	63	
∑X	339688	429618	622519	1391825	
Mean	16175.619	20458	29643.7619	22092.46	
$\sum X^2$	5498896862	8812915524	18509185969	32820998355	
Std.Dev.	460.0641	1090.653	1664.0455	5781.1705	
Result Details					
Source	***SS***	***df***	***MS***		
Between-treatments	1988755163	2	994377581.4	*F* = 715.34001	
Within-treatments	83404610.76	60	1390076.846		
Total	2072159774	62			
Summary of Data for Top-6 web links					
	Treatments				The *f*-ratio value is 103.2461. The *p*-value is < .00001. The result is significant at *p* < .05.
	WLRGA	WLRBBBC	WLRGbSA	Total	
N	21	21	21	63	
∑X	531811	497171	675155	1704137	
Mean	25324.3333	23674.8095	32150.2381	27049.794	
$\sum X^2$	13508828565	11794273501	21887889275	47190991341	
Std.Dev.	1432.9957	1091.8969	3012.4348	4201.4535	
Result Details					
Source	***SS***	***df***	***MS***		
Between-treatments	848027558.6	2	424013779.3	*F* = 103.2461	

Within-treatments	246409579.7	60	4106826.329		
Total	1094437138	62			
Summary of Data for Top-7 web links					
	Treatments				The *f*-ratio value is 179.6266. The *p*-value is < .00001. The result is significant at *p* < .05.
	WLRGA	WLRBBBC	WLRGbSA	Total	
N	21	21	21	63	
∑X	551850	549706	763048	1864604	
Mean	26278.5714	26176.4762	36335.619	29596.889	
$\sum X^2$	14509611142	14411447384	27934877482	56855936008	
Std.Dev.	623.7587	1050.7465	3233.0793	5189.0993	
Result Details					
Source	***SS***	***df***	***MS***		
Between-treatments	1430539697	2	715269848.4	*F* = 179.6266	
Within-treatments	238918901.3	60	3981981.689		
Total	1669458598	62			
Summary of Data for Top-8 web links					
	Treatments				The *f*-ratio value is 131.83786. The *p*-value is < .00001. The result is significant at *p* < .05.
	WLRGA	WLRBBBC	WLRGbSA	Total	
N	21	21	21	63	
∑X	621746	664608	826988	2113342	
Mean	29606.9524	31648	39380.381	33545.111	
$\sum X^2$	18424500348	21142981784	32695161598	72262643730	
Std.Dev.	908.1887	2339.5277	2530.4062	4701.3236	
Result Details					
Source	***SS***	***df***	***MS***		
Between-treatments	1116328476	2	558164238.2	*F* = 131.83786	
Within-treatments	254023047.9	60	4233717.465		
Total	1370351524	62			
Summary of Data for Top-9 web links					
	Treatments				The *f*-ratio value is 170.81064. The *p*-value is < .00001. The result is significant at *p* < .05.
	WLRGA	WLRBBBC	WLRGbSA	Total	
N	21	21	21	63	
∑X	721547	752918	924199	2398664	
Mean	34359.381	35853.2381	44009.4762	38074.032	
$\sum X^2$	24805667569	27049818724	40803504761	92658991054	
Std.Dev.	829.4372	1662.3839	2549.4203	4635.3859	
Result Details					
Source	***SS***	***df***	***MS***		
Between-treatments	1133161170	2	566580585	*F* = 170.81064	
Within-treatments	199020600	60	3317010		
Total	1332181770	62			
Summary of Data for Top-10 web links					
	Treatments				The *f*-ratio value is 304.83577. The *p*-value is < .00001. The
	WLRGA	WLRBBBC	WLRGbSA	Total	
N	21	21	21	63	
∑X	732240	828567	1012119	2572926	
Mean	34868.5714	39455.5714	48196.1429	40840.095	

$\sum X^2$	25534606826	32744642829	48914218655	1.07193E+11	result is significant at $p < .05$.
Std.Dev.	349.5771	1628.7783	2588.3078	5840.5223	
Result Details					
Source	***SS***	***df***	***MS***		
Between-treatments	1925436227	2	962718113.3	$F = 304.83577$	
Within-treatments	189489202.9	60	3158153.381		
Total	2114925429	62			

7. Conclusions

This work presented in this paper will be helpful in ranking web links based on web usage, content and structure data using parameters of importance like keywords frequency, access frequency, number of unique visitors, duration stayed, number of hubs, and number of authorities. GA based algorithm WLRGA, and physics based algorithms WLRBBBC and WLRGbSA have been proposed for selecting Top-T web links for T∈{5,6,7,8,9,10,11,12,13,14,15} and for different population sizes {100,150,200,250}. Experimental results have shown that the WLRGbSA performs better, in comparison to other algorithms, in terms of selecting better quality Top_T web links. WLRBBBC performs better from WLRGA in general. In terms of convergence rate, the WLRGbSA has performed best among all proposed algorithms. Then algorithm WLRBBBC has better convergence rate than WLRGA. As per the performance and the quality of results, WLRGbSA performs best in all conditions and there is a trade-off seen between the rest two algorithms at some values but the WLRBBBC has an upper edge in quality while the WLRGA has a faster convergence rate. These algorithms can be successfully applied in the scenarios where response time is not a big constraint like website structure reorganization, web content personalization, target marketing and many more. Even these approaches can be appropriately used in recommendation systems, search engine optimization, and business intelligence solutions.

7.1 Future Directions

The algorithms can also be compared on space complexities. Higher values of Top-T and bigger population sizes can also be used for better and further analysis. Real and published datasets can also be used to assess the performance. As per new researches, a better and more optimised selection method can be used to improve the performance. The demands and the requirements of the users over the internet vary day to day. As per the demands in the future, there can be need to discover new and more effective algorithms. New data is being created and added on the internet daily. New and more complex problems will be there that need to be analysed and examined for these algorithms.

Declarations

Ethical Approval and Consent to participate Not applicable

Consent for publication Yes

Availability of supporting data The datasets generated randomly similar to the real data and is available from the corresponding author on reasonable request.

Competing interests The authors have no competing interests to declare that are relevant to the content of this article.

Funding No funding was received to assist with the preparation of this manuscript.

Authors' contributions Related work was carried out by all the authors. Algorithms analysis and discussion was done by Santosh Kumar and implementation was done by Arjun Sharma and Aman Kumar Goel. Experiments and analysis was carried out by all authors. All authors drafted the manuscript.

Acknowledgments: Not applicable.

Authors' information Dr. Santosh Kumar is Professor at ABES Engineering College, Ghaziabad, India. Aman Kumar Goyal, Arjun Sharma, and Arpit Verma are undergraduate scholar at the same College.

References

1. Chattopadhyay, S., Marik, A. and Pramanik, R., 2022. A Brief Overview of Physics-inspired Metaheuristic Optimization Techniques. arXiv preprint arXiv:2201.12810.
2. Zeidabadi, F.A., Doumari, S.A., Dehghani, M., Montazeri, Z., Trojovsky, P. and Dhiman, G., 2022. MLA: a new mutated leader algorithm for solving optimization problems. Computers, Materials & Continua, 70(3), pp.5631-5649.
3. Manik Sharma, Prableen Kaur . A Comprehensive Analysis of Nature-Inspired Meta-Heuristic Techniques for Feature Selection Problem. Arch Computat Methods Eng 28, 1103–1127 (2021).
4. Dola Gobinda Padhan1, Syed Sarfaraz Nawaz, and Padmanabhuni Ravikanth, A Fractional Order Control Strategy for LFC via Big Bang Big Crunch & Grey Wolf Optimization Algorithms, E3S Web of Conferences 184, 01016 (2020).
5. Amine, Khalil. (2019). Multiobjective Simulated Annealing: Principles and Algorithm Variants. Advances in Operations Research. 2019. 1-13. 10.1155/2019/8134674.
6. DEMIROL, Doygun; OZTEMIZ, Furkan; KARCI, Ali (2018). [IEEE 2018 International Conference on Artificial Intelligence and Data Processing (IDAP) - Malatya, Turkey (2018.9.28-2018.9.30)] 2018 International Conference on Artificial Intelligence and Data Processing (IDAP) - Performance Comparison of Physics Based Meta-Heuristic Optimization Algorithms. , (), 1–5. doi:10.1109/IDAP.2018.8620742
7. Hossein Mousavi, Seyed Muhammad et al. “Galaxy gravity optimization(GGO) an algorithm for optimization, inspired by comets life cycle.” *2017 Artificial Intelligence and Signal Processing Conference (AISP)* (2017): 306-315.
8. Yan Kang , Hao , Chunhui Wang and Li Da, A hybrid Big Bang-Big Crunch algorithm for energy optimization on heterogeneous distributed systems, MATEC Web of Conferences 119, 01047 (2017).
9. Nazmul Siddiqui, Hojjat Adeli, “Nature-Inspired Chemical Reaction Optimisation Algorithms. Cognit Comput. 2017; 9(4): 411–422”. Doi: 10.1007/s12559-017-9485-1 , June 17, 2017.
10. Can, Ümit, and Bilal Alataş. "Physics based metaheuristic algorithms for global optimization." (2015).
11. Sariman, Guncel & Kucuksille, Ecir Ugur. (2014). Web Based Educational Tool for Metaheuristic Algorithms. Pamukkale University Journal of Engineering Sciences. 20. 46-53. 10.5505/pajes.2014.15870.
12. Sivanandam, S.N. and Deepa, S.N., 2008. Genetic algorithm optimization problems. In Introduction to genetic algorithms (pp. 165-209). Springer, Berlin, Heidelberg.

13. Jiao, Licheng, Wang, Lipo, Gao, Xinbo, Liu, Jing, Wu, Feng (2006). *[Lecture Notes in Computer Science] Advances in Natural Computation Volume 4222 || Small-World Optimization Algorithm for Function Optimization. , 10.1007/11881223(Chapter 33), 264–273.* doi:10.1007/11881223_33
14. H. Edwin Romeijn and Robert L. Smith (1994). Simulated Annealing and Adaptive Search in Global Optimization. Probability in the Engineering and Informational Sciences, 8, pp 571-590 doi:10.1017/S0269964800003624
15. Richard Formato(2007) . "Central Force Optimization: a New Metaheuristic with Applications in Applied Electromagnetics," Progress In Electromagnetics Research, Vol. 77, 425-491. Doi: 10.2528/PIER07082403
16. N. Siddique and H. Adeli, "Physics-based search and optimization: inspirations from nature," Expert Systems, vol. 33, no. 6, pp. 607–623, 2016.
17. Shah-Hosseini, Hamed. "Otsu's criterion-based multilevel thresholding by a nature-inspired metaheuristic called Galaxy-based Search Algorithm." 2011 Third World Congress on Nature and Biologically Inspired Computing (2011): 383-388.
18. Hamed Shah-Hosseini. 2011. Principal components analysis by the galaxy-based search algorithm: a novel metaheuristic for continuous optimisation. Int. J. Comput. Sci. Eng. 6, 1/2 (July 2011).
19. Robic, F., Micallef, D., Borg, S. P., & Ellul, B. (2020). Implementation and fine-tuning of the Big Bang-Big Crunch optimisation method for use in passive building design. Building and Environment, 106731. doi:10.1016/j.buildenv.2020.106731 10.1016/j.buildenv.2020.106731
20. Saima Khan, Przemyslaw Grudniewski, Yousaf Shad Muhammad, Adam J. Sobey(2022). The benefits of co-evolutionary Genetic Algorithms in voyage optimisation,Ocean Engineering,Volume 245, 110261, ISSN 0029-8018.
21. Hao Chen, Huadong Yong, Youhe Zhou(2022). Crack detection in bulk superconductor using Genetic Algorithm, Engineering Fracture Mechanics, Volume 265, 108372, ISSN 0013-7944.
22. Qiang Luo, Yunqing Rao, Xiaoqiang Guo, Bing Du(2022). A biased genetic algorithm hybridized with VNS for the two-dimensional knapsack packing problem with defects, Applied Soft Computing, Volume 118, 108479, ISSN 1568-4946.
23. Jaume Jordán, Javier Palanca, Pasqual Martí, Vicente Julian(2022). Electric vehicle charging stations emplacement using genetic algorithms and agent-based simulation, Expert Systems with Applications, Volume 197, 116739, ISSN 0957-4174.
24. Lucas do Vale Machado, Antonio Carlos Fernandes(2022). Moonpool dimensions and position optimization with Genetic Algorithm of a drillship in random seas, Ocean Engineering, Volume 247, 110561, ISSN 0029-8018.
25. Haiying Song, Minghua Cai, Jian Cen, Chenhua Xu, Qingmeng Zeng(2022). Research on Energy Saving Optimization Method of Electric Refrigerated Truck Based on Genetic Algorithm, International Journal of Refrigeration, ISSN 0140-7007.
26. Doddy Prayogo, Min-Yuan Cheng, Yu-Wei Wu, Albertus Arief Herdany, Handy Prayogo(2018). Differential Big Bang - Big Crunch algorithm for construction-engineering design optimization, Automation in Construction, Volume 85, ISSN 0926-5805.

27. Jia-Jun Wang, Tufan Kumbasar(2018). Big Bang-Big Crunch optimized hierarchical sliding-mode control of X-Z inverted pendulum, Simulation Modelling Practice and Theory, Volume 86, ISSN 1569-190X.
28. Mostafa Sedighizadeh, Masoud Esmaili, Amir Eisapour-Moarref(2017). Voltage and frequency regulation in autonomous microgrids using Hybrid Big Bang-Big Crunch algorithm, Applied Soft Computing, Volume 52, ISSN 1568-4946.
29. S.R. Desai, Rajendra Prasad(2017). A novel order diminution of LTI systems using Big Bang Big Crunch optimization and Routh Approximation, Applied Mathematical Modelling, Volume 37, Issues 16–17, ISSN 0307-904X.
30. A. Kaveh, S. Talatahari(2010). Optimal design of Schwedler and ribbed domes via hybrid Big Bang–Big Crunch algorithm, Journal of Constructional Steel Research, Volume 66, Issue 3, ISSN 0143-974X.
31. B. Sravan Kumar, M. Suryakalavathi, G.V. Nagesh Kumar(2016). Optimal Power Flow with Static VAR Compensator Using Galaxy Based Search Algorithm to Minimize Real Power Losses, Procedia Computer Science, Volume 92, ISSN 1877-0509.
32. Faegheh Sardari, Mohsen Ebrahimi Moghaddam(2016). An object tracking method using modified galaxy-based search algorithm, Swarm and Evolutionary Computation, Volume 30, ISSN 2210-6502.
33. https://phys.libretexts.org/Courses/University_of_California_Davis/UCD%3A_Physics_9A__Classical_Mechanics/4%3A_Linear_Momentum/4.2%3A_Center_of_Mass