Global Journals LaTeX JournalKaleidoscope™
Artificial Intelligence formulated this projection for compatibility purposes from the original article published at Global Journals. However, this technology is currently in beta. *Therefore, kindly ignore odd layouts, missed formulae, text, tables, or figures.*

An Enhanced Cuckoo Search for Optimization of Bloom Filter in Spam Filtering

Premalatha[1], Dr.. Arulanand Natarajan[2] and ubramanian S[3]

[1] Bannari Amman Istitute of Technology

Received: 9 April 2012 Accepted: 30 April 2012 Published: 15 May 2012

Abstract

Bloom Filter (BF) is a simple but powerful data structure that can check membership to a static set.The trade-off to use Bloom filter is a certain configurable risk of false positives. The odds of a false positive can be made very low if the hash bitmap is sufficiently large. Spam is an irrelevant or inappropriate message sent on the internet to a large number of newsgroups or users. A spam word is a list of well-known words that often appear in spam mails. The proposed system of Bin Bloom Filter (BBF) groups the words into number of bins with different false positive rates based on the weights of the spam words. An Enhanced Cuckoo Search (ECS) algorithm is employed to minimize the total membership invalidation cost of the BFs by finding the optimal false positive rates and number of elements stored in every bin. The experimental results have demonstrated for CS and ECS for various numbers of bins.

Index terms— Bin Bloom Filter, Bloom Filter, Cuckoo Search, Enhanced Cuckoo Search, False positive rate, Hash function, Spam word.

1 INTRODUCTION

spam filter is a program that is used to detect unsolicited and unwanted email and prevent those messages from getting into user's inbox. A spam filter looks for certain criteria on which it stands decisions. For example, it can be set to look for particular words in the subject line of messages and to exclude these from the user's inbox. This method is not effective, because often it is omitting perfectly legitimate messages and letting actual spam through.

The strategies used to block spam are diverse and includes many promising techniques. Some of the strategies like black list filter, white list /verification filters rule based ranking and naïve bayesian filtering are used to identify the spam.

A BF presents a very attractive option for string matching (Bloom 1970). It is a space efficient randomized data structure that stores a set of signatures efficiently by computing multiple hash functions on each member of the set.

It queries a database of strings to verify for the membership of a particular string. The answer to this query can be a false positive but never be a false negative. The computation time required for performing the query is independent of the number of signatures in the database and the amount of memory required by a BF for each signature is independent of its length (Feng et al 2002). This paper presents a BBF which allocates different false positive rates to different strings depending on the significance of spam words and gives a solution to make the total membership invalidation cost minimum. BBF groups strings into different bins via smoothing by bin means technique. The number of strings to be grouped and false positive rate of each bin is identified through GA which minimizes the total membership invalidation cost. This paper examines different number of bins for given set of strings, their false positive rates and number of strings in every bin to minimize the total membership invalidation cost.

The organization of this paper is as follows. Section 2 deals with the standard BF. Section 3 presents the CS technique. Section 4 reports optimized BBF using ECS. Performance evaluation of CS and ECS for the BBF is discussed in section 5.1

2 II.

3 BLOOM FILTER

Bloom filters (Bloom 1970) are compact data structures for probabilistic representation of a set in order to support membership queries. This compact representation is the payoff for allowing a small rate of false positives in membership queries which might incorrectly recognize an element as member of the set.

Given a string S the BF computes k hash functions on it producing k hash values and sets k bits in an m-bit long vector at the addresses corresponding to the k hash values. The value of k ranges from 1 to m. The same procedure is repeated for all the members of the set. This process is called programming of the filter. The query process is similar to programming, where a string whose membership is to be verified is input to the filter. The bits in the m-bit long vector at the locations corresponding to the k hash values are looked up. If at least one of these k bits is not found in the set then the string is declared to be a nonmember of the set. If all the bits are found to be set then the string is said to belong to the set with a certain probability. This uncertainty in the membership comes from the fact that those k bits in the m-bit vector can be set by any other n-1 members. Thus finding a bit set does not necessarily imply that it was set by the particular string being queried. However, finding a bit not set certainly implies that the string does ? + ? ? ? ? ? ? ? ? ? =**(2)**

The derivative equals 0 when kmin=(1n2)(m/n). In this case the false positive probability f is:f(k min) = (1 ? p) k min = (1 2) k min = (0.6185) m /n (3) of course k should be an integer, so k is ?ln 2 . (m n) ? ?

The BF has been widely used in many database applications (Mullin 1990 **??**Fan et al,2000). BFs have great potential for representing a set in main memory **??**Peter and Panagiotis, 2004) in stand-alone applications. BFs have been used to provide a probabilistic approach for explicit state model checking of finite-state transition systems **??**Peter and Panagiotis, 2004). It is used to summarize the contents of stream data in memory **??**Jin et al,2004; **??**eng and Rafiei,2006), to store the states of flows in the on-chip memory at networking devices **??**Bonomi et al,2006)

4 CUCKOO SEARCH

Cuckoo search is an optimization algorithm inspired by the brood parasitism of cuckoo species by laying their eggs in the nests of other host birds proposed by Yang and Deb (2009). If a host bird discovers the eggs are not their own, it will either throw these foreign eggs away or simply abandon its nest and build a new nest elsewhere. Each egg in a nest represents a solution, and a cuckoo egg represents a new solution. The better new solution (cuckoo) is replaced with a solution which is not so good in the nest. In the simplest form, each nest has one egg. When generating a new solution Levy flight is performed. The rules for CS are described as follows:

? Each cuckoo lays one egg at a time, and dumps it in a randomly chosen nest ? The best nests with high quality of eggs will carry over to the next generations; ? The number of available host nests is fixed, and a host can discover an foreign egg with a probability p a ?[0, 1]. In this case, the host bird can either throw the egg away or abandon the nest so as to build a completely new nest in a new location

The algorithm for CS is given below:

Generate an initial population of n host nests; not belong to the set. In order to store a given element into the bit array, each hash function must be applied to it and, based on the return value r of each function (r1, r2, ? , rk), the bit with the offset r is set to 1. Since there are k hash functions, up to k bits in the bit array are set to 1 (it might be less because several hash functions might return the same value). Figure 1 is an example where m=16, k=4 and e is the element to be stored in the bit array. A BBF is a date structure considering weight for spam word. It groups spam words into different bins depending on their weight. It incorporates the information on the spam word weights and the membership likelihood of the spam words into its optimal design. In BBF a high cost bin lower false positive probability and a low cost bin has higher false positive probability. The false positive rate and number of strings to be stored is identified through optimization technique GA which minimize the total membership invalidation cost. Figure 2 shows Bin BF with its tuple <n,f,w> configuration.

5 Evaluate its fitness Fi

? ? ? ? ? ? ? × ? ? ? ? ? ? ? = = j i 1 j j i r n m r ln2 i 2 1 f ( ) i i f ln r = (1 ? i ?L)

The objective function f(L) taken as standard for the problem of minimization is? ? ? ? < ? = max max max C F(L) if 0 C F(L) if F(L) C f(L) (5)

where Cmax is a large constant.

which has an infinite variance with an infinite mean. Here the consecutive jumps of a cuckoo essentially form a random walk process which obeys a power-law step-length distribution with a heavy tail. The representation of egg (solution) is given in figure **??**.

where nij, fij and wij refer respectively the number of words, false positive rate of and the weight of the jth bin of ith egg. The triplet <n,f,w> encodes a single bin. The false positive rate fij can be obtained from equation

(1) where nij is drawn from the ith egg in the nest, m is known in advance and k is calculated from equation (3). One egg in the nest represents one possible solution for assigning the triples <n, f, w>. At the initial stage, each egg randomly chooses different <n, f, w> for L Bins based on the given constraints. The fitness function for each egg can be calculated based on the equation (5).

6 VII. EXPERIMENTAL RESULTS

Cuckoo Search employs Levy flight for finding new solutions from equation (7). CS and ECS consider 10 nests and 50 iterations. The parameters pa,? and ? are set as 0.3, 1 and 1.5 respectively. The total number of strings taken for testing is 250, 500, and 1000. The string weights are varying from 0.0005 to 5. The size of the BF is 1024. This experimental setup is applied for number of bins from 4 to 7.

Figures **??**a, 4b, 4c and 4d correspondingly show the total membership invalidation cost obtained from BBF for bin sizes from 4 to 7 for 1000 strings using CS and ECS algorithm. In this experimental setup the ECS performs better than CS. Figures **??**a, 5b, 5c and 5d show the total membership invalidation cost obtained from BBF for bin sizes from 4 to 7 respectively for 500 strings. Figures **??**a, 6b, 6c and 6d show the cost of BBF from bin sizes 4 to 7 for 250 strings. For all the string sizes the ECS outperforms CS.

In CS, 10 nests which equals to number of nests in ECS and 40 nests which equals to number of eggs in ECS are taken to find the total membership invalidation cost for 1000 strings. Figure **??** shows the total membership invalidation cost obtained from BBF for the bin sizes ranging from 4 to 10 using CS and ECS. It shows that the cost is decreased when the numbers of bins are increased. The results obtained from ECS outperform CS for all bin sizes from 4 to 10.

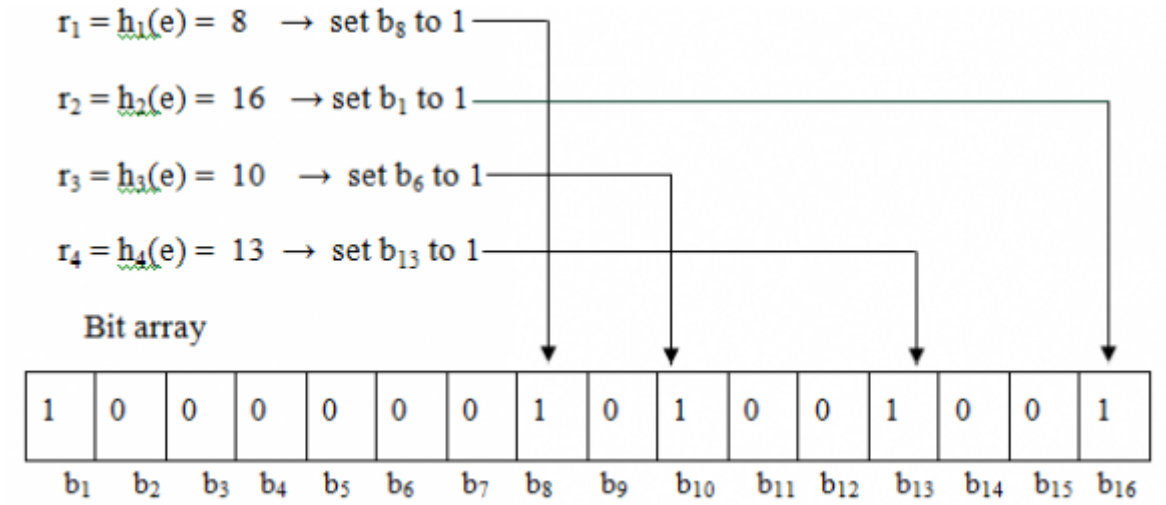
(A) (B) [1] [2]

2012

Figure 1: A © 2012

[1]January 2012© 2012 Global Journals Inc. (US)

[2]© 2012 Global Journals Inc. (US) Global Journal of Computer Science and Technology Volume XII Issue I Version I



1

Figure 2: Fig. 1 :

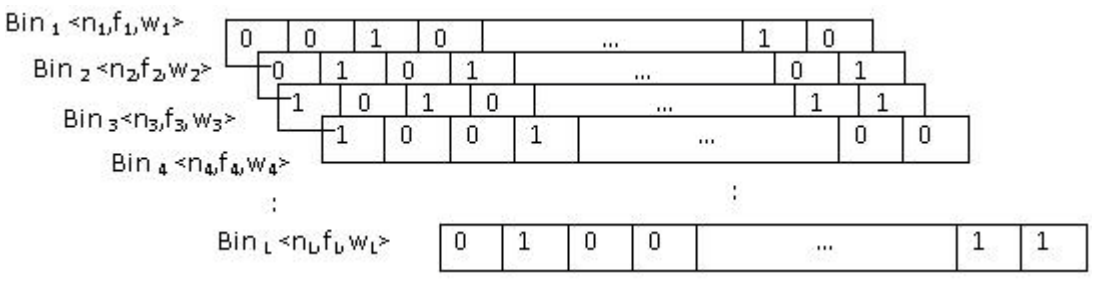


Figure 3:

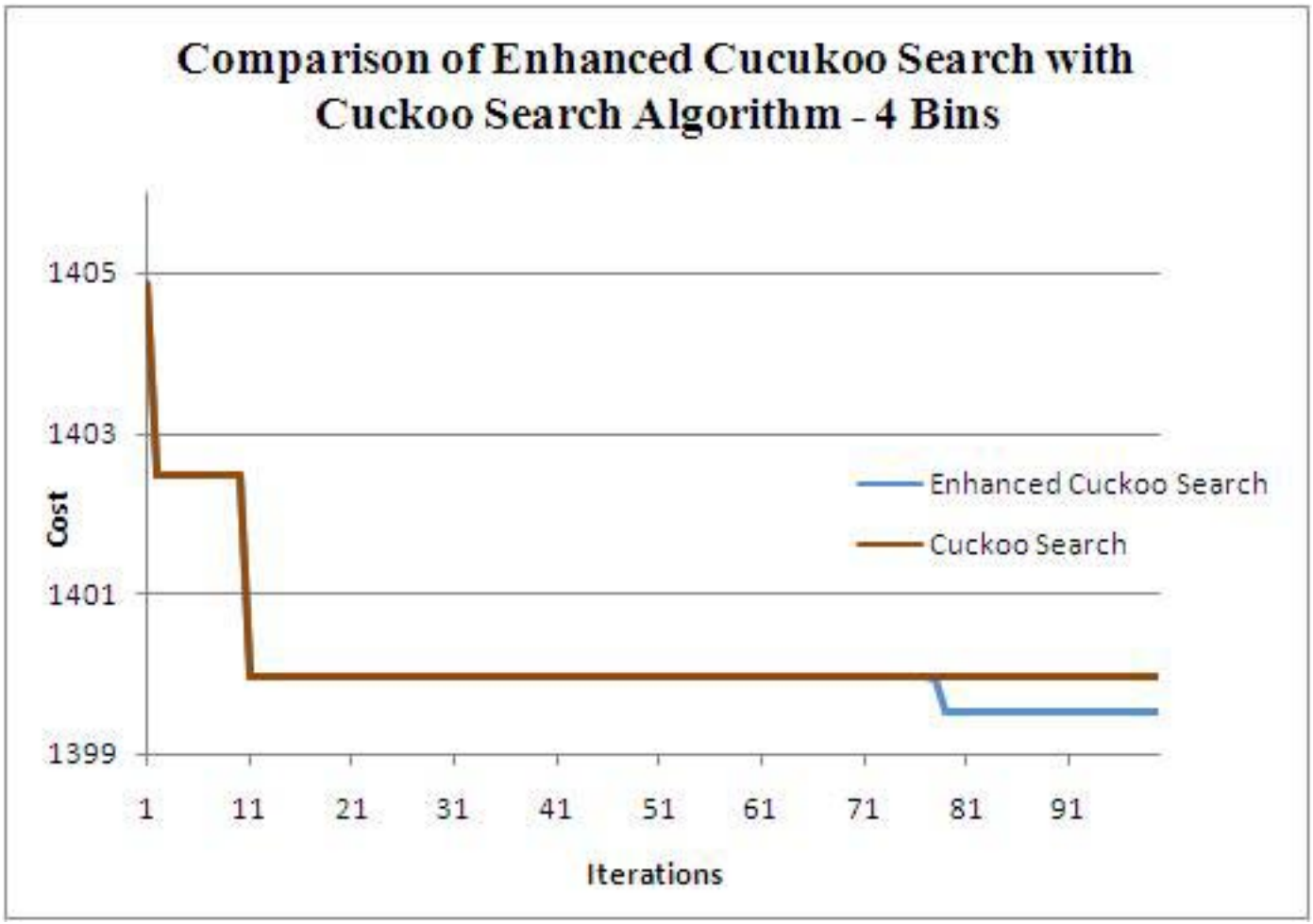

Comparison of Enhanced Cucukoo Search with Cuckoo Search Algorithm - 4 Bins
1405
1403
1401
1399
Cost
1
11
21
31
41
51
61
71
81
91
Iterations
Enhanced Cuckoo Search
Cuckoo Search

Figure 4:

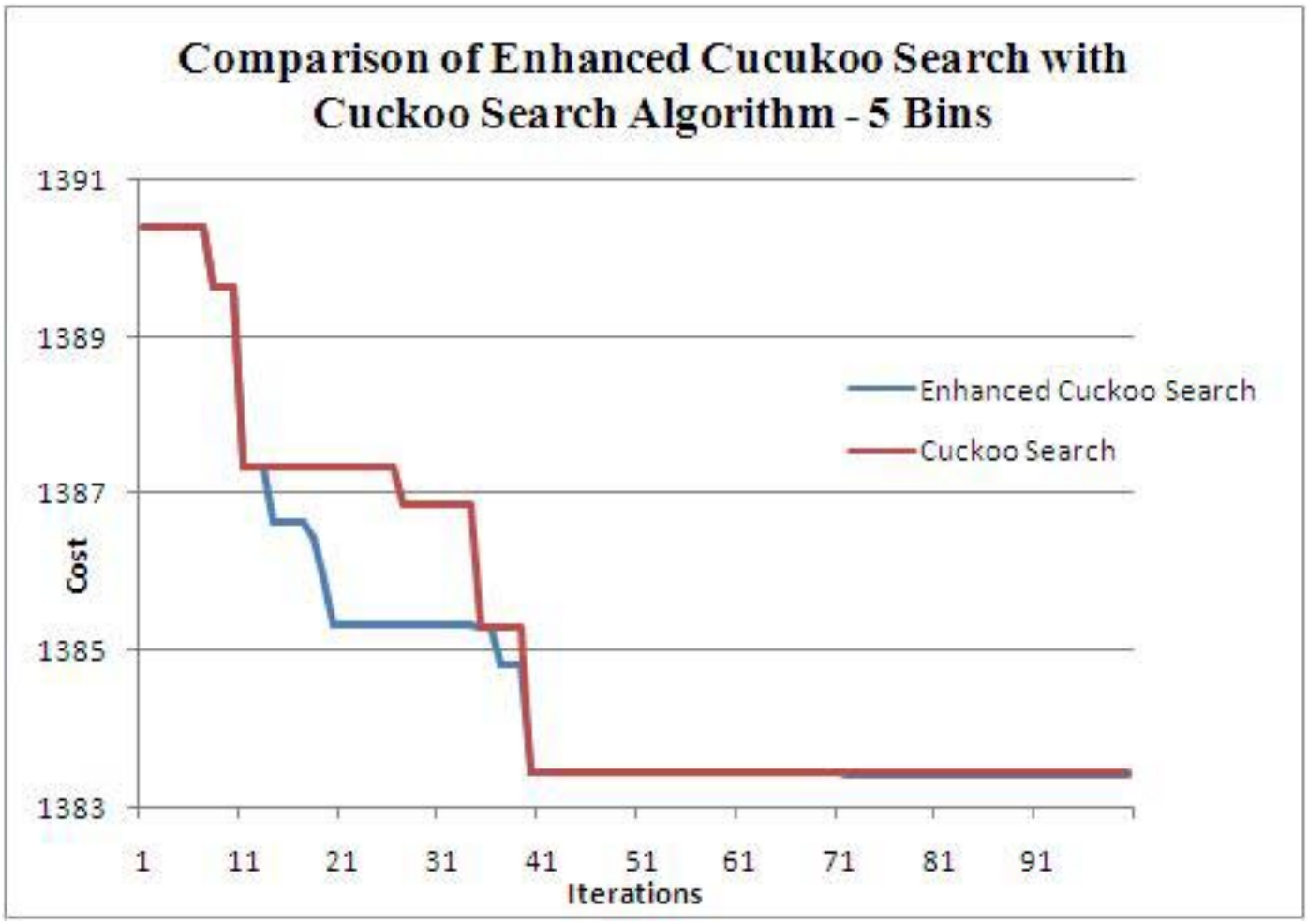


Figure 5:

$x_i =$ | n_{i1} | f_{i1} | w_{i1} | n_{i2} | f_{i2} | w_{i2} | … | n_{ij} | f_{ij} | w_{ij} | … | n_{iL} | f_{iL} | w_{iL} |

2

Figure 6: Fig. 2 :F

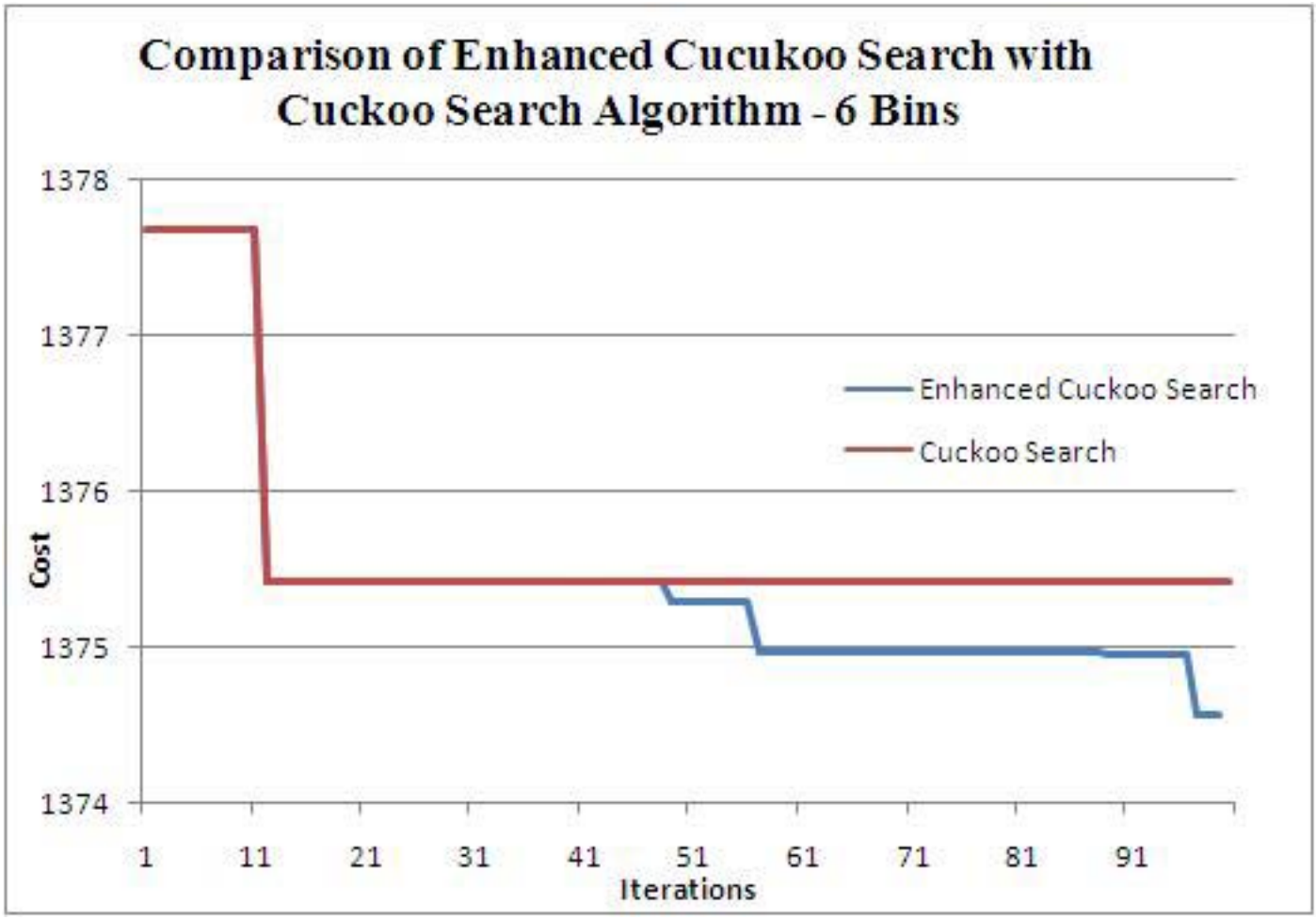

Comparison of Enhanced Cucukoo Search with Cuckoo Search Algorithm - 6 Bins
1378
1377
1376
1375
1374
Cost
1
11
21
31
41
51
61
71
81
91
Iterations
Enhanced Cuckoo Search
Cuckoo Search

Figure 7: ?

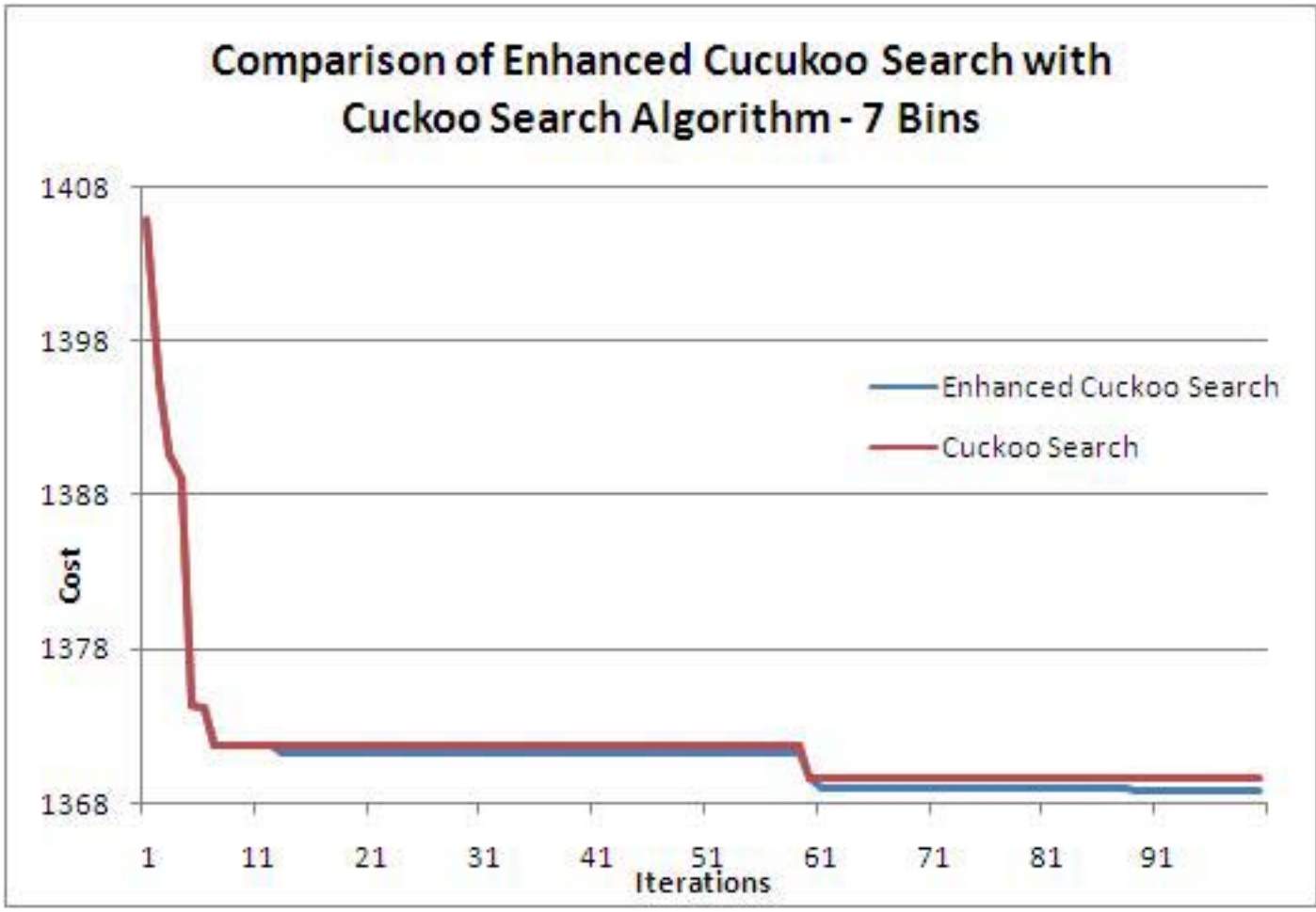


Figure 8:

It is applied in networking literature (Brooder and Mitzenmacher, 2005). A BF can be used as a summarizing technique to aid global collaboration in peer-to-peer networks (Kubiatowicz et al., 2000 ; Li et al, 2002 ; Cuena-Acuna et al, 2003). It supports probabilistic algorithms for routing and locating resources (Rhea and Kubiatowicz 2004; Hodes et al,2002 ; Reynolds and Vahdat, 2003; Bauer et al, 2004) and share Web cache information

Figure 9:

[Peter and Panagiotis] , C D Peter , M Panagiotis .

[Laufer et al. ()] , R P Laufer , P B Velloso , O C M Duarte , Filters . GTA-05-43. 2005. Univ. of California, Los Angeles (UCLA (Technical Report Research Report)

[Fan et al. ()] 'A Scalable Wide Area Web Cache Sharing Protocol'. L Fan , P Cao , J Almeida , A Broder , Summary Cache . *IEEE/ACM Trans. Networking* 2000. 8 (3) p. .

[Hodes ()] 'An Architecture for Secure Wide Area Service Discovery'. T D Hodes , CzerwinskiS E , ZhaoB . *Wireless Networks* 2/3, 2002. 8 p. .

[An Enhanced Cuckoo Search for Optimization of Bloom Filter in Spam Filtering Global Journal of Computer Science and Techn 'An Enhanced Cuckoo Search for Optimization of Bloom Filter in Spam Filtering Global Journal of Computer Science and Technology Volume XII Issue I Version I 80'. *Global Journals Inc* January 2012. 2012. US.

[Jin et al. ()] 'Analysis and Management of Streaming Data: A Survey'. C Jin , W Qian , A Zhou . *J. Software* 2004. 15 (8) p. .

[Deng and Rafiei ()] 'Approximately Detecting Duplicates for Streaming Data Using Stable Bloom Filters'. F Deng , D Rafiei . *Proc. 25th ACMSIGMOD*, (25th ACMSIGMOD) 2006. p. .

[Xie et al. ()] 'Basket Bloom Filters for Membership Queries'. K Xie , Y Min , D Zhang , J Wen , G Xie , J Wen . *Proceedings of IEEE Tencon'05*, (IEEE Tencon'05) 2005. p. .

[Bonomi et al. ()] 'Beyond Bloom Filters: From Approximate Membership Checks to Approximate State Machines'. F Bonomi , M Mitzenmacher , R Panigrahy , S Singh . *Proc. ACM SIGCOMM*, (ACM SIGCOMM) 2006. p. .

[Bloom ()] B Bloom . *Space/time tradeoffs in hash coding with allowable errors*, 1970. 13 p. .

[Bauer et al. ()] 'Bringing Efficient Advanced Queries to Distributed Hash Tables'. D Bauer , P Hurley , R Pletka . *Proc. IEEE Conf. Local Computer Networks*, (IEEE Conf. Local Computer Networks) 2004. p. .

[Kirsch and Mitzenmacher ()] *Building a Better Bloom Filter*, A Kirsch , M Mitzenmacher . tr-02-05.pdf. 2006. Dept. of Computer Science, Harvard Univ (Technical Report)

[Hao and Kodialam ()] 'Building High Accuracy Bloom Filters Using Partitioned Hashing'. F Hao , M Kodialam , LakshmanT . *Proc. SIGMETRICS/Performance*, (SIGMETRICS/Performance) 2007. p. .

[Mitzenmacher ()] 'Compressed Bloom Filters'. M Mitzenmacher . *IEEE/ACM Trans.Networking* 2002. 10 (5) p. .

[Yang and Deb ()] 'Cuckoo search via Lévy flights'. X S Yang , S Deb . *World Congress on Nature & Biologically Inspired Computing*, NaBIC 2009. 2009. IEEE Publications. p. .

[Kirsch and Mitzenmacher ()] 'Distance-Sensitive Bloom Filters'. A Kirsch , M Mitzenmacher . *Proc. Eighth Workshop Algorithm Eng. and Experiments (ALENEX '06)*, (Eighth Workshop Algorithm Eng. and Experiments (ALENEX '06)) 2006.

[Reynolds and Vahdat ()] 'Efficient Peer-to-Peer Keyword Searching'. P Reynolds , A Vahdat . *Proc. ACM Int'l Middleware Conf*, (ACM Int'l Middleware Conf) 2003. p. .

[Li and Zhong ()] 'Fast Statistical Spam Filter by Approximate Classifications'. K Li , Z Zhong . *Proc. Joint Int'l Conf. Measurement and Modeling of Computer Systems, SIGMETRICS/Performance*, (Joint Int'l Conf. Measurement and Modeling of Computer Systems, SIGMETRICS/Performance) 2006. p. .

[Broder and Mitzenmacher ()] 'Network Applications of Bloom Filters: A Survey'. A Broder , M Mitzenmacher . *Internet Math* 2005. 1 (4) p. .

[Kubiatowicz et al. ()] 'Oceanstore: An Architecture for Global-Scale Persistent Storage'. J Kubiatowicz , D Bindel , Chen Czerwinski , S Eaton , P Geels , D . *ACM SIGPLAN Notices* 2000. 35 (11) p. .

[Mullin ()] 'Optimal Semijoins for Distributed Database Systems'. J Mullin . *IEEE Trans. Software Eng* 1990. 16 p. .

[Mackert and Lohman ()] 'Optimizer Validation and Performance Evaluation for Distributed Queries'. L F Mackert , G M Lohman . *Proc. 12th Int'l Conf. Very Large Data Bases (VLDB)*, (12th Int'l Conf. Very Large Data Bases (VLDB)) 1986. p. .

[Kubiatowicz ()] 'Probabilistic Location and Routing'. Rhea S C Kubiatowicz , J . *Proc. IEEE INFOCOM*, (IEEE INFOCOM) 2004. p. .

[Probabilistic Verification Proc. Fifth Int'l Conf. Formal Methods in Computer-Aided Design ()] 'Probabilistic Verification'. *Proc. Fifth Int'l Conf. Formal Methods in Computer-Aided Design*, (Fifth Int'l Conf. Formal Methods in Computer-Aided Design) 2004. p. .

[Li et al. ()] 'Self-Organization in Peer-to-Peer System'. J Li , J Taylor , L Serban , M Seltzer . *Proc. ACM SIGOPS*, (ACM SIGOPS) 2002.

175 [Kumar et al. ()] 'Space-Code Bloom Filter for Efficient Per-Flow Traffic Measurement'. A Kumar , J Xu , J
176 Wang , O Spatschek , LiL . *Proc. 23rd IEEE INFOCOM*, (23rd IEEE INFOCOM) 2004. p. .

177 [Cohen and Matias ()] 'Spectral Bloom Filters'. S Cohen , Y Matias . *Proc. 22nd ACM SIGMOD*, (22nd ACM
178 SIGMOD) 2003. p. .

179 [Chazelle et al. ()] 'The Bloomier Filter: An Efficient Data Structure for Static Support Lookup Tables'. B
180 Chazelle , J Kilian , R Rubinfeld . *Proc. Fifth Ann. ACM-SIAM Symp. Discrete Algorithms (SODA)*, (Fifth
181 Ann. ACM-SIAM Symp. Discrete Algorithms (SODA)) 2004. p. .

182 [Feng et al. ()] 'The BLUE active queue management algorithms'. W Feng , K G Shin , D D. & D Kandlur ,
183 Saha . *IEEE/ACM Transactions on Networking* 2002. 10 p. .

184 [Cuena-Acuna and Peery ()] 'Using Gossiping to Build Content Addressable Peer-to-Peer Information Sharing
185 Communities'. F M Cuena-Acuna , C Peery , MartinR P , NguyenT D , Plantp . *Proc. 12th IEEE Int'lSymp.*
186 *High Performance Distributed Computing*, (12th IEEE Int'lSymp. High Performance Distributed Computing)
187 2003. p. .