A Comparative Study of Wu Manber String Matching Algorithm and its Variations

Vasudha Bhardwaj
LNCTE, Bhopal

Vikram Garg
LNCTE, Bhopal

ABSTRACT
String matching algorithms is become one of the most important topic in the computer science world. These algorithms are used in many real world problems like as scanning the threat in intrusion detection system, finding the pattern in text mining, match the similarity of the document in the plagiarism detection system, recognition in bio informatics and so on. String Matching Algorithms are broadly categorized into single pattern string matching and multiple pattern string matching algorithms. To get the faster solution of the problem most of the cases multiple pattern matching is the best choice. Aho-Corasick, Shift-OR, Robin Karp but Wu Manber Algorithm is better choice because it search the pattern faster(take less time) as well as occupies less space. This paper discus the various pit falls of the wu manber algorithm with their detailed explanation. Also discuss the various improved wu manber algorithm and comparison between these algorithms.

Keywords
Multiple String Matching, Wu Manber, BLAST Algorithm, Quick Wu Manber, Improved Wu Manber.

1. INTRODUCTION
String matching algorithms[1,2] becomes necessary tool for most of the application like as Intrusion Detection System[4,5], Plagiarism Detection[6,7], Text Mining[8,9], Bio Informatics[10,11]. In all we have to find the pattern from the database. Due to the popularity of the string matching algorithm number of algorithms are invented which are broadly categorized into single pattern matching[2] like as Navies algorithm[1], MP[20], KMP[21], BM[12], BMH[22], BNDM[25], TNDM[26], BNDM with Q gram[27] and multiple pattern matching[3] like as Aho-Corasick[23], Commentz Walter, Shift OR[24], Shift OR with Q gram[24], Wu Manber[13].

Finding single pattern again and again from a same text is a slow process but most of the application wants result as fast as can so instead of single pattern multiple pattern is suitable option here we can find multiple pattern at once. Aho-Corasick is the benchmark algorithm of multiple pattern matching algorithms which is based automata. When number of pattern increase the size of automata also increase due to this complexity is also increase as well as it takes lots of space. After Aho Corasick algorithm Commentz Walter present a new algorithm which is based on the combination of boyer moore and aho-corasick algorithm. By the use of boyer moore concept in aho-corasick algorithm it is faster than the basic aho-corasick algorithm. Shift OR and Shift OR with Q gram both are multiple patterns algorithms which is based on the approximate string matching means it may give false matches. Wu Manber is finest algorithm which is hashing based algorithm occupies less space. Wu Manber algorithm used the concept of boyer-moore algorithm which describe in next section. After Wu Manber lots of improvement is done to improve the performance which describe one by one.

In 1994 Sun Wu and Udi Manber implement a multiple exact pattern matching algorithm named as Wu Manber[13] which is faster than the previous algorithm and occupies less space. It uses the idea of boyer moore and hash table for shifting by using this we can shift the window by M-B+1.

In 2006 Yang did some changes in basic Wu Manber to improve the efficiency of algorithm named as Quick Wu Manber[14]. In this algorithm yang introduced head table in preprocessing phase to get the better results.

In 2008 Chen Zhen implement algorithm named as Improved Wu Manber Algorithm[15] which is again a variation of basic Wu Manber algorithm. Here we construct two shift table instead of single shift table to get the better shift in case of like patterns.

In 2009 Quick Multiple Matching Algorithm[16] is implemented by the Liuling Dai. In this algorithm the maximum shift distance m is achieved where m is the smallest pattern size by eliminate overlapping between hash table and shift table.

X.Chen implements the High Concurrence Wu Manber Algorithm[17] in 2009. Here we resolve the limitation that pattern should have same size. It divides the patterns in to different sets of equal size and for each set different method is used to obtain high concurrency.

Addresing Filtering Wu Manber Algorithm[18] is again a improved variation of basic Wu Manber Algorithm[13] which is implemented by B. Zhang in 2009. It filter the pattern with the use of prefix table and arrange the pattern in ascending order which is hardly used in the basic wu manber algorithm.

In 2013 Yoon Ho implements the B-Layered bad-character SHIFT table which is named as BLAST[19]. It is another improved version of Wu Manber algorithm. This algorithm overcomes the limitation of multi character shift table here we use shift table based on single character.

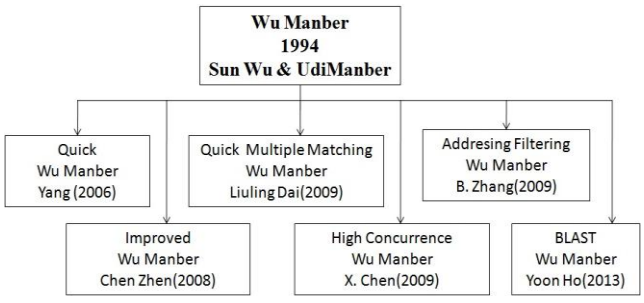

Figure 1: Evolution of Wu Manber String Matching Algorithms

Figure 1 Shows the evolution of the various Wu Manber algorithm with their authors name and publishing year of the algorithm.

2. WU MANBER ALGORITHM

Boyer Moore algorithm[12] is the single pattern algorithm which gives the faster result among all single pattern matching algorithms. Wu Manber[13] try the concept of the boyer moore algorithm to get the maximum shift like as boyer moore algorithm as well as work on multiple pattern instead of single pattern. Before discussing WM algorithm we have to discuss some pre requisites like Boyer-Moore which required to understand the basic Wu Manber algorithm.

- **Boyer Moore Algorithm**

In 1977 Boyer-Moore algorithm[12] is developed by R.S.Boyer and J.C.Moore. This algorithm is based on exact single pattern matching. The algorithm starts the searching of patterns from right to left beginning with rightmost character. Boyer-Moore algorithm is based on two pre-processed functions bad character shift and good suffix shift. In case of complete match of pattern or mismatch happened these two functions are used two shift the scanning window to the right based on which gives the maximum value either good suffix shift function or bad character shift function. The calculation of good suffix is based on how many characters were matched before the mismatch of character is happened by using the good suffix of mismatched character and matched character of pattern. The calculation of bad character of text character is done by mismatch character of pattern. This calculates how many positions ahead to start the next search based on the value of the character that caused the mismatch. The advantage of Boyer-Moore algorithm is with the uses of two functions good-suffix and bad-char; algorithm provides a better shift value because shift value is based on maximum values of these two functions. The limitation of algorithm is implementation and understanding of good suffix function in pre-processing is complex and after many matches if there is mismatch than bad character i.e. mismatch character gives small shift, depends on pattern length.

- **Wu Manber Algorithm**

Wu Manber algorithm[13] consists in to two phase named as preprocessing phase and scanning phase. In preprocessing phase the basic calculation is done which is further required in the searching of the algorithm. Scanning phase consist of the various step taken to find the patterns.

- **Preprocessing Stage**

Preprocessing of the patterns is required to speed up the scanning stage. First thing is to find the smallest length pattern among all the patterns(Smallest pattern length is m). Now preprocessing stage build the three tables named as SHIFT Table, HASH Table and PREFIX Table. SHIFT Table gives us the maximum value of the shift in case of mismatch. HASH Table values are used when the value of the SHIFT table is zero to search the matched pattern. When numbers of patterns are large most of the patterns consist common suffix which will cause collision in HASH table. To speed up the process another table is introduce named as PREFIX Table it is the mapping of first B character of the patterns. All these tables can be easily understand with the help of the example.

Let us consider the TEXT is “STRINGCAREMATCH” and we have patterns CARE, GOOGLE, YAHOO. So we have to find the patterns in the text. In WU Manber algorithm we have to construct SHIFT table, HASH table and PREFIX table. Suppose value of B is 2. CARE is the minimum length pattern among all so value of m is 4. So we have to consider first four character of the all pattern to construct shift table. Shift table of the example is shown in the Table 1.

Table 1: SHIFT Table of the Wu Manber Algorithm

SHIFT TABLE									
CA	AR	RE	GO	OO	OG	YA	AH	HO	*
2	1	0	2	1	0	2	1	0	3

After calculating the shift table hash table is constructed whose values are calculated on the basis of the last B character of the m size pattern. If two patterns have same last B character occupies same hash value. Table 2 shows the hash value of different patterns of example. In example no two pattern has same hash value so prefix table is not required.

Table 2: Hash Table of the Wu Manber Algorithm

HASH TABLE			
RE	OG	HO	*
1	2	3	NULL

- **Scanning Stage**

In scanning stage start with the first m character of the text called it window. After that compare the last B character of the window from the shift table and according to the value of the shift table shift the window. If value of the shift table is zero than hash table comes in the existence. Here we match the hash value of the various patterns. If the hash table consist of multiple entries than we consider the prefix table and compare the prefix of the pattern if value of prefix table is matched we compare with actual pattern against the text if matches mark as occurrence and move the window right by one. Continue the process till the end of the text. Table 3 shows the various steps involved in the scanning stage of the example and report the occurrence of the pattern if pattern found.

Table 3: Scanning phase of Wu Manber Algorithm

SCANNING PHASE			
STEP	TEXT	SHIFT	MATCHING
1	STRINGCAREMATCH	3	
2	STRINGCAREMATCH	3	
3	STRINGCAREMATCH	0	CARE
4	STRINGCAREMATCH	3	

- **Advantages**

Wu MAnber Algorithm uses the concept of boyer moore algorithm which gives the larger shift as compare to other algorithm so no need to traverse whole text. No need of additional space for data structure. Less number of comparisons required.

- **Disadvantages**

In Wu Manber Algorithm all the pattern should be of same length because it only consider first m character of all pattern where m is size of smaller pattern. If smaller pattern length is too small than shift is small this makes it less efficient. Cannot handle more than few hundreds patterns

3. VARIENT OF WU MANBER

After Wu Manber algorithm lots of its variation comes into existence which make the small changes in the basic Wu Manber algorithm to get the better results. These algorithms are discussed below one by one:

- **Quick Wu Manber Algorithm**

In Wu-Manber algorithm, the maximum shift distance in Shift table is m-B+1 which is improved in QWM algorithm[14]. The limitation of WM algorithm[13] is, during the searching process when value in the Shift table is equal to zero, the shift distance is always equal to 1 whether there is full match or not. Here in preprocessing phase HEAD table is also calculated. The use of Head table is to find whether the first two characters in the text which we are currently scanning is the prefix of any pattern. Initialize Head table with the value 0 that means the first two characters does not appear in any pattern. Now update Head table with value 1, for first two characters who appears in some patterns as prefix. First step is to check whether the current text is the prefix of any pattern. If it is, then follow the method of Wu-Manber algorithm. Whether there is a complete match or not, the shift distance always equal to the value in Shift table. Limitations of QWM algorithm is, in each scanning step two hash calculations(for Head and Shift table) are performed to get shift distance whereas in basic WM algorithm only one needed. And also the complex of Head table computation depends on the size of pattern set. It requires more memory as compare to the basic Wu-Manber algorithm because in pre-processing four tables are created instead of three.

- **Improved Wu Manber Algorithm**

Improved Wu Manber algorithm[15] uses two shift tables instead of single shift table. The improved Wu-Manber algorithm introduced with three differences as compare to the basic Wu-Manber algorithm: 1) to better the quality of the Boyer-Moore like Shift table, for each original pattern a pattern representative (rarest substring with fixed length) is chosen; 2) to increase the possibilities of shifting text sliding window, a second Boyer-Moore Shift table is computed; 3) to design a balanced Hash table for improving the scanning speed of possible matching patterns a simple hash function with good randomness property is crafted.

The advantages of improved algorithm over the previous explained algorithms is that the text scanning speed of the improved Wu-Manber algorithm is faster because with the use of second shift table exact string comparison is avoided most of the time. This algorithm have some limitations like calculation of two shift tables is not a good idea in pre-processing stage and when the number of patterns reaches 40,000 or more, the introduced extra shift table still cannot improve the performance.

Quick Multiple Matching Wu Manber Algorithm

The limitation of basic WM algorithm[13] is that when the value in Shift table is zero then we go to check for the Hash table, in other words when the Hash table is not null then there must be larger (greater than zero) value in the shift table. This function overlap between Hash table and Shift table is eliminated in this algorithm. QMM algorithm[16] modifies the basic WM algorithm by introducing character next to scan window in Shift table. It uses the idea of QS algorithm as Shift table is only used to get shift distance. Shift table have the entry by hashing the last char of scan window and next to last character of scan window. The Hash table is used for finding the possible matching patterns list whether it is null or not. So the shift table is not depended on the possible matching, the maximum shift distance (m instead of m-B+1) is achieved using this concept.

The advantages of QMM algorithm is, the maximum shift distance m is achieved in Shift table as it is independent on the possible match of patterns, its only use is to get the shift distance and also the functional overlap between Hash table and Shift table is eliminated. But the algorithm has the limitation that at each scanning step both shift and Hash table is checked to get shift distance and to find possible match, whereas in basic WM algorithm does not check Hash table until value in shift table gets zero.

- **High Concurrence Wu Manber Algorithm:**

The performance of Wu-Manber algorithm is greatly affected when short patterns are mixed with long patterns. In order to solve this problem, High Concurrence Wu-Manber Algorithm[17] is proposed. It uses the concept of dividing all the patterns into different sets according to their length and for each set, independent data structures are established and different process methods are used in parallel to obtain high concurrency when searching patterns in text, because there are few resources shared among these sets. The search operation is executed concurrently because each pattern set has independent data structures, and the common text file is read only. Multiple threads are used for the pattern matching. When short patterns and long patterns mix together, the performance of HCWM is better than WM[13] obviously. The reason is that HCWM processes long patterns and short patterns in different ways. But the limitations of this algorithm are pre-processing or organizing the data structure takes time, because it divides all the patterns into different sets.

- **Addressing Filtering Wu Manber Algorithm**

The basic Wu-Manber algorithm has the following limitations. 1) There are redundant information and operations. 2) The Prefix table for filter pattern is constructed, but is hardly used. 3) Need to traverse the whole link list. These limitations affect the performance of algorithm. Address filtering based search method is introduced to overcome these limitations. AFWM algorithm[18] has only difference with the basic Wu-Manber algorithm is in the prefix link list; all the same prefix patterns will be sorted in ascending order according to the address pointers of the patterns. So, we only need to match the patterns whose address pointers locate in the range of Hash table. In this way the search process of improved algorithm can be finished faster than the basic Wu-Manber algorithm.

The advantages of address filtering algorithm is, it avoids traversing the whole link list and with the help of address pointer is it uses the Prefix table sufficiently. There are some limitations in this algorithm such as improved effect of AFWM is sensitive to the number of patterns, when the number of patterns increases the effect of AFWM is more obvious. The reason is that as the number of patterns increasing, the length of link list will become longer

- **B-Layered bad-character SHIFT Algorithm**

BLAST algorithm[19] is based on multiple layered shift tables with single character search at a time. BLAST algorithm overcomes the limitations of multi character shift table (as it has average shift values in typical search) used in various WM algorithms. The shift tables in BLAST algorithm is based on single character, so it takes less memory for shift table as compare to other WM algorithms. This algorithm also resolves the performance degradation issue, when the frequency of last character occurrence is higher in single character table. The advantages of BLAST algorithm is maximum shift distance m is achieved with the use of single character shift tables and also the memory size of shift table is reduced as compare to multi character shift table used in the basic WM algorithm. But the performance of BLAST

algorithm decreases, if every character in the rightmost position of the patterns is present, because BLAST algorithm‟s SHIFT tables are based on a single character search technique, so the Hash table will be compared for every character in the text.

4. COMPARISION AND ANALYSIS

Wu Manber is the efficient multiple pattern matching. Table 4 shows the comparison among all the variants of the Wu Manber algorithms with the parameters**.**

Table 4: Comparison of Various Wu Manber String Matching Algorithms

Algorithm / Parameter	BASIC WU MANBER ALGORITHM	QUICK WU MANBER ALGORITHM	IMPROVED WU MANBER ALGORITHM	QUICK MULTIPLE MATCHING ALGORITHM	HIGH CONCURRENCE WU MANBER	ADDESS FILTERING WU MANBER	BLAST ALGORITHM
Number of SHIFT TABLE	1	1	2	1	1	1	2
Number of HASH TABLE	1	1	1	1	1	1	1
Number of PREFIX TABLE	1	1	1	1	1	1	1
Number of HEAD TABLE	-	1	-	-	-	-	-
MAXIMUM SHIFT	M-B+1	M	M-B+1	M	M-B+1	M-B+1	M
SIZE OF PATTERNS	EQUAL	EQUAL	EQUAL	EQUAL	UNEQUAL	EQUAL	EQUAL

5. CONCLUSION

String matching algorithms are used in various applications. Due to this lots of algorithms are invented day by day which tries to improve the performance of the previous algorithms. Wu Manber is one of them which is exact multiple pattern algorithm used in various application. It is very attractive algorithm because it is faster algorithm used the concept of the boyer moore algorithm to get maximum shift distance and occupies the lesser space as compare to other algorithm. Due to popularity lots of variants are comes to improve the basic wu manber algorithm whose comparative analysis is given above.

6. REFERENCES

[1] Christian Charras and Thierry Lecroq," Handbook of Exact String_Matching Algorithms", Published in King's college publication, Feb 2004.

[2] Alberto Apostolico and Zvi Galil," Pattern Matching Algorithms" Published in Oxford University Press, USA, 1st edition, May 29, 1997.

[3] Fang Xiangyan, Xiong Tinggang, Ding Yidong and Youguang, "The research and improving for multi-pattern string matching algorithm", In the proc. of 2010 IEEE International Conference on Intelligent Computing and Intelligent Systems (ICIS), Vol. 1, pp. 266-270, Oct. 2010.

[4] Byung-joo Kim, Kim, "Kernel based intrusion detection system", In the proc. of Fourth Annual ACIS International Conference on Computer and Information Science, pp. 13-18, 2005.

[5] Pei-fei Wu and Hai-juanShen, "The Research and Amelioration of Pattern-matching Algorithm in Intrusion Detection System", In the proc. of IEEE 14th International Conference on High Performance Computing and Communication & IEEE 9th International Conference on Embedded Software and Systems (HPCC-ICESS), pp. 1712-1715, 25-27 June 2012.

[6] Alzahrani S.M. Salim N. and Abraham A., "Understanding Plagiarism Linguistic Patterns, Textual Features, and Detection Methods", In the proc. of IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews", Vol. 42, Issue 2, pp. 133-149, march 2012.

[7] Ramazan S. Aygün "structural-to-syntactic matching similar documents", Journal Knowledge and Information Systems, ACM Digital Library, Volume 16 Issue 3, pages 303-329, Aug 2008.

[8] Sanchez D., Martin-Bautista M.J., Blanco I. and Torre C.," Text Knowledge Mining: An Alternative to Text Data Mining", In the proc. of IEEE International Conference on Data Mining Workshops, ICDMW '08, pp. 664-672, 15-19 Dec. 2008.

[9] Qiong Zhang, Roger D. Chamberlain, Ronald S. Indeck, Benjamin M. West and Jason White," Massively Parallel Data Mining Using Reconfigurable Hardware: Approximate String Matching", In Proceedings of the 18th International Parallel and Distributed Processing Symposium (IPDPS'04), 2004.

[10] Marturana, F, Gianluigi Me and Tacconi, S,"A Case Study on Digital Forensics in the Cloud", In the proc. of International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (Cyber C), pp. 111-116, 10-12 Oct. 2012.

[11] Jooyoung Lee, Sungkyung Un, and Dowon Hong," Improving Performance in Digital Forensics", In the Proc.

of International Conference on Availability, Reliability and Security, 2009.

[12] BOYER, R. S. AND MOORE, J. S,"A fast string searching algorithm", Communication of ACM 20, Vol. 10, pp. 762–772, 1977.

[13] Wu S. and U.Manber, "A Fast Algorithm for Multi-Pattern Searching," Technical Report TR-94-17 Department of Computer Science, University of Arizona, Tucson, AZ, May 1994.

[14] Yang Dong hong, XuKe and Cui Yong, "An improved Wu-Manber multiple patterns matching algorithm", In the proc. Of 25th IEEE International Performance, Computing, and Communications Conference, IPCCC, pp. 680, 10-12 April 2006.

[15] Chen Zhen and Wu Di, "Improving Wu-Manber: A Multi-pattern Matching Algorithm", In the proc. of 2008 IEEE International Conference on Networking, Sensing and control (ICNSC), pp. 812 – 817, 6-8 April 2008.

[16] LiulingDai,"An aggressive algorithm for multiple string matching"Information Processing Letters, Volume 109, pp. 553–559, May 2009.

[17] Baojun Zhang, Xiaoping Chen, Xuezeng Pan, and Zhaohui Wu "High concurrence Wu-Manber Multiple Patterns Matching Algorithm", Proceedings of the International Symposium on Information Proces, p.404,August 2009.

[18] Baojun Zhang , XiaoPing Chen , Lingdi Ping , Wu, Zhaohui, "Address Filtering Based Wu-Manber Multiple Patterns Matching Algorithm", In the proc. of 2009 Second International Workshop on Computer Science and Engineering[WCSE], Qingdao, Vol.1, pp. 408 – 412,28-30 Oct. 2009.

[19] Yoon-Ho, Seung-Woo,"BLAST: B-Layered bad-character SHIFT tables for high-speed pattern matching", Journal of Information Security, Institution of Engineering and Technology (IET), Volume 7, pp.195-202, sept.2013. MORRIS JR, J. H. AND PRAT, V. R.,"A linear pattern-matching algorithm", Technical Report 40, University of California, Berkeley, 1970.

[20] KNUTH, D. E, MORRIS JR J. H AND PRATT V. R,"Fast pattern matching in strings", In the procd. Of SIAM J.Comput. Vol. 6, 1, pp. 323–350, 1977.

[21] HORSPOOL, R. N,"Practical fast searching in strings", In proc. Of Software Practical Exp, Vol. 10, 6, pp. 501–506, 1980.

[22] Alfred v. aho and margaret j. corasick,"efficient string matching: an aid to bibliographic search" communication of acm, vol. 18, june 1975.

[23] R. Baeza-Yates and G. Gonnet,"A new approach to text searching", Communication of ACM, Vol. 35(10), pp. 74–82, 1992.

[24] G. Navarro,M. Raffinot, "Fast and flexible string matching by combining bit-parallelism and suffix automata",ACM Journal. Experimental Algorithmics 2000.

[25] HannuPeltola and JormaTarhio, Alternative Algorithms for Bit-Parallel String Matching, String Processing and Information Retrieval, Spire 2003 Springer, LNCS 2857,pp. 80-93,2003. Branislav Durian, Jan Holub,HannuPeltola andJormaTarhio," Tuning BNDM with q-Grams" In proc. of workshop on algorithm engineering and experiments SIAM USA, pp. 29-37, 2009