International Journal of Mathematics and Statistics Invention (IJMSI)
E-ISSN: 2321 – 4767 P-ISSN: 2321 - 4759
www.ijmsi.org Volume 1 Issue 2 ǁ December. 2013ǁ PP-19-24

Theoretical Analysis Of Novel Variants Of MTF Algorithm

Debashish Rout[1]
Department Of Computer Science and Engineering,Vssut,Burla,India1

ABSTRACT : *In this paper, we have proposed a novel variants of MTF algorithm, which we popularly call as MPITF and MSITF.We have also performed theoritical study and comparative performance analysis of MPITF and MSITF with MTF*

I. INTRODUCTION

Data structure provides a way to store data in structure way efficiently, in the primary memory of computer.Various operations such as search, insert and delete can be performed on a data structure. If the data items are unsorted and stored in a linear list, each items can be searched by scanning the the list of items one by one linearly. In *linear search* if multiple elements are searched, the total search time can be reduced by making the data structure self-organizing. In self-organization datastructure the items can be re-organized after each operation to reduce the time of future operations Thereby enhancing the performance.

1.1 List Accessing Problem

List Accessing problem or List Update problem is the method used in the self-organizing linear search. In List Update problem a list (l) of records and a request sequence (σ) are taken as inputs. When a record is accessed from the list then the list is reconfigured to reduce the future search cost. When a record is accessed from the list, some cost is provided for that. List accessing problem is mainly implemented by single linked list. But it may be implemented through doubly linked list and tree also.

1.2 Cost Model

In the list accessing problem, we have two different models based on operations and list type. They are Static list accessing model and Dynamic list accessing model. The Static list accessing model is the one in which the number of items in the list is fixed and only the access operation can be performed. The Dynamic list accessing model is the one in which the size of the list varies dynamically and all the three operations i.e. insert, delete and access can be performed. In our work, we have considered only the static model of list accessing problem and hence we consider only the access operation. As one of the key issues is to find out the optimal access cost of elements on the list, we need a cost model which is an efficient tool to measure the access cost incurred by the various list accessing algorithms. A number of cost models have been developed and used so far but here we have considered only Full Cost Model (FCM) and Partial Cost Model (PCM). In Full Cost Model, the cost of accessing an item in the i^{th} position from the front of the list is i. In the Partial Cost Model the cost of accessing an item in the i^{th} position from the front of the list is (i-1) because we have to make (i-1) comparisons before accessing the i^{th} element in the list. So, the cost of accessing the first element in the list would be 1 in FCM and 0 in PCM. We are illustrating both the models as follows. Suppose the list is 1, 2, 3 and the request sequence is 1, 2, and 3. The costs of elements according to the various models are presented in below.

Elements	Access cost in PCM	Access Cost in FCM
1	0	1
2	1	2
3	2	3
Total cost	3	6

1.3 Application

List accessing algorithms are widely used in Data Compression. Other important applications of list update algorithms are computing point maxima in computational geometry, resolving collisions in hash table and dictionary maintenance. The List Accessing Problem is also of significant interest in the contest of self organizing data structures.

1.4 List Accessing Algorithms

The algorithm which efficiently recognizes the list and reduces the cost for the access is called a list accessing algorithm. List accessing algorithms can be of two types such as online algorithm and offline algorithm. In online algorithm, the request sequence is partially known. In offline algorithm, the request sequence is fully known; online algorithms can be further classified into two types such as deterministic and randomized. Deterministic algorithm is one which produces the same output always for a given request sequence or the algorithm passes through same states for a given request sequence. Some of the popular deterministic algorithms for the list accessing problem are Move-To-Front (MTF), Transpose (TRANS) and Frequency Count (FC).

MTF: After accessing an item, it is moved towards the front of the list without changing the order of other items in the list.

TRANS: After accessing an element, it is exchanged with its proceeding element.

FC: There is a counter for each item which counts the frequency of each item of the list according based on the requests from the request sequence. The list is arranged in the non-increasing order of frequency count of items in the list.In randomized online algorithm, while processing the request sequence, the algorithm makes some random decision at some step. Some well known randomized algorithms are SPLIT, BIT, COMB and TIME-STAMP.

1.5 Literature Review

List update problem was first studied by McCabe in 1965 with the concept of relocatable records in serial files. He also introduced two list accessing algorithms Move-To-Front (MTF) and Transpose (TRANS). Rivest has examined a class of heuristics for maintaining the sequential list in optimal order with respect to the average time required to search for a specified element with an assumption of fixed probability of each search in his experimental study.He has shown that MTF and Transpose heuristic are optimal within a constant factor. Hester and Hirschberg have done a comprehensive survey of all permutation algorithms that modified the order of linear search lists with an emphasis on average case analysis. Sleator and Tarjan in their seminar paper have formally introduced the concept of competitive analysis for online deterministic list update algorithms such as MTF, TRANS and FC using amortized analysis and potential function method. MTF is proved to be 2-competitive where as FC and TRANS are not competitive. Irani proposed first randomized online list update algorithm known as SPLIT which is 1.932-competitiv. Albers, Von-Stengel, and Werchner proposed a simple randomized online algorithm-COMB that achieves a 1.6-competitive, the best randomized algorithm in literature till date. Albers introduced the concept of look ahead in the list update problem and obtained improved competitive ratio for deterministic online algorithms. Reingold and Westbrook have proposed an optimal offline algorithm which runs in time $O(2^l l!n)$ where l is the length of the list and n is the length of request sequence. Bachrach et al. have provided an extensive theoretical and experimental study of online list update algorithm in 2002. The study of locality of reference in list accessing problem was initiated by Angelopoulos in 2006, where he proved MTF is superior to all algorithms. Relatively less work has been done on the offline algorithms for the list accessing problem. For analyzing the performance of online algorithm by competitive analysis an optimal offline algorithm is essential. Ambühl in 2000 proved that off-line list update is NP-hard by showing a reduction from the Minimum Feedback Arc Set Problem. In 2004, Kevin Andrew and David Gleich showed that the randomized BIT algorithm is 7/4-competitive using a potential function argument. They introduced the pair-wise property and the TIMESTAMP algorithm to show that the COMB algorithm, a Combination of the BIT and TIMESTAMP algorithms, is 8/5-competitive. In 2009, in one of the paper a survey has been done on online algorithms for self organizing sequential search .

1.6 Our Contribution

In this paper, we have proposed a novel variants of MTF algorithm, which we popularly call as MPITF and MSITF.We have also performed theoritical study and comparative performance analysis of MPITF and MSITF with MTF.

1.7 Organization of paper

The paper has been introduced in section 1.MTF algorithm is discussed in section 2. Proposed algorithms are discussed in section 3. Section 4 Theoritical analysis of the algorithms. The paper is concluded in section 5 followed by a set of references

II. MTF ALGORITHM

In deterministic online algorithm, we mainly concentrate on MTF and FC list accessing algorithms. Here we develop new variants of both MTF and FC algorithms and also the concept of hybrid algorithms.Here we have made access cost comparisons between MTF algorithm and my proposed algorithms and try to find some formulas for my proposed algorithms.

2.1Algorithm MTF:On *accessing an element x in the list, x is moved to the front of the list*

2.1 Variants of MTF
2.2.1 MPITF (Move Preceding Item to Front):-*On accessing an item x in the list, just preceding item of x in the list is moved to the front of the list.*
2.2.2 MSITF (Move Succeeding Item to Front of the List):- *On accessing an item x in the list, just succeeding item of x in the list is moved to the front of the list.*

III. THEORITICAL ANALYSIS

Special Type of Request Sequences:-
For characterization of request sequences we have considered the following parameters.
1. Size of the list:-l
2. Size of the request sequence:–n
Type 1($T_{1)}$: Request sequence is exactly the same as that of the list.
Type 2($T_{2)}$: Request sequence is the reverse order as that of the list.
Type 3($T_{3)}$: Request sequence is a permutation of arbitrary order as that of the list (except Type 1 and Type 2).
Type 4($T_{4)}$: Request sequence consist of any single element of the list at position p repeated n times where $1 \leq p \leq n$.
Type 5($T_{5)}$: Request sequences consist of more than one element each repeated at least once.

[1] For MPITF Algorithm:-(Move Preceding Item)

- No of Best case sequences =No of worst case sequences for **n>=2**.Where the n is the length of request sequence.
- No of best case sequences and worst sequences=**n**, for **n>2**.
  The worst case value=**nl-1**.where n is the size of request sequence and l is the size of the list.

[2] For MSITF Algorithm(Move Succeeding Item to the Front):-

- No. of worst case sequences=**n-1**
- The worst case value is **nl-1**.(Where **n** is the size of the request sequence and **l** is the size of the list)

Proposition 1:-For $\mathbf{T_1}$ request sequence of size n, the cost of MPITF is given by

$$C_{MPITF}(T_1) = \sum_{i=1}^{n} P_i$$

P_i --→Position of the element

Proposition 2:-For $\mathbf{T_2}$ request sequence of size n, the cost of MPITF is given by

$$C_{MPITF}(T_2) = C_n = \begin{cases} \sum_{i=\frac{n}{2}}^{n} P_i & \boxed{\text{n,even}} \\ \sum_{i=\frac{n-1}{2}}^{n} P_i & \boxed{\text{n, odd}} \end{cases}$$

Proposition 3:-For T_1 request sequence of size n, the cost of MSITF is given by

$$C_{MSITF}(T_1) = \begin{cases} \sum_{e=1}^{\frac{n}{2}} P_{2i} & \boxed{\text{n,even}} \\ \sum_{o=0}^{\frac{n-1}{2}} P_{2o+1} + \frac{n-1}{2} & \boxed{\text{n, odd}} \end{cases}$$

Proposition 4:-For T_2 request sequence of size n, the cost of MPITF is given by

$$C_{MSITF}(T_2) = n.(n-1)+1$$

Proposition 5:-For Type 4 request sequence of size n, the cost of MSITF is given by

$$C_{MSITF}(T_4) = \begin{cases} C_n = n^2 \\ C_{i=} C_{i+1} - 1 \end{cases}, Where, i = n-1.........1$$

Proposed Theorems:-
Theorem 1:-For Type 2 request sequence of size n, the cost of MPITF is given by

$$C_{MSITF}(T_2) = n.(n-1)+1$$

Proof-

l=Size of the list
n=Size of the request sequence
$|\sigma| = n \quad |L| = n$

Assume that $C_{MSITF}(Type2) = P_n$

$$P_n = n.(n-1)+1$$

Basic Case

P_1 =1(1-1) +1
R. H.S=n (n-1) +1
=1(1-1) +1
=1

L. H. S-

As list contains a single element, the position of that element is always 1. The reverse order of that single element is same as that of the single element of that list. There is no succeeding element present in the list for that single element. From that list only one request sequence is generated containing single element. By help of MSITF, the total accessed cost is 1.

L.H. S= R. H. S

Hence P_1 is true.

Inductive Step

Let P_n is true for n=k i.e.

$$P_k = k.(k-1)+1$$
$$= k^2 - k + 1$$

Now we have to prove that $P_{k+1} = k+1.(k+1-1)+1$

$$= k+1(k)+1$$
$$= k^2 + k + 1$$

Let the elements of the list of size k be $l_1, l_2 .... l_k$ and the elements of request sequence be $r_{1,} r_2 .... r_k$ such that all the elements of the list are present in the request sequence. Let (k+1) th element l_{k+1} occurs after l_k in the list and r_{k+1} occur after r_k in the request sequence. The access cost of k element is $k^2 - k + 1$. Whereas the total access cost including (k+1) th element in request sequence in the list is

$= P_k + 2k$

As every time the next cost is increased with a factor of 2n i.e. 2k.

$= k^2 - k + 1 + 2k$

$= k^2 + k + 1$

So L.H.S= R.H.S

Hence P_{k+1} is true.

So the expression is true for all n.

Theorem 2:-For Type 1 request sequence of size n, the cost of MPITF is given by

$$C_{MPITF}(T_1) = \sum_{i=1}^{n} P_i$$

P_i -->Position of the element

Proof-

Let $C_{MPITF}(n)$ cost is denoted by $C_n = \sum_{i=1}^{n} P_i$

Where C_n is a proposition about natural number such that C_n is true?

Basic Case

$C_1 = \sum_{i=1}^{1} P_1 = 1$

Let there is a single element in the list l i.e. l_1 and single element in the request sequence r i.e. r_1.

As a single element present in the list, so the position of that element is 1 always. The cost of the request sequence by using MPITF is 1 always.

So C_1 is true.

Inductive Step

Let C_n is true for n=k i.e. $C_k = \sum_{i=1}^{k} P_i$

Now we have to prove that $C_{k+1} = \sum_{i=1}^{k+1} P_i$

R.H.S= $\sum_{i=1}^{k+1} P_i$ *=0+1+2+3+.... +k+ (k+1)*

= (0+1+2+3+.... +k) + (k+1)

= ($\sum_{i=1}^{k} P_i$ *) + (k+1)*

Let the elements of the list of size k be $l_1, l_2 .... l_k$ and the elements of request sequence be $r_{1,} r_2 \ldots r_k$ such that all the elements of the list are present in the request sequence. Let (k+1) th element l_{k+1} occurs after l_k in the list and r_{k+1} occur after r_k in the request sequence. The access cost of k element is

$\sum_{i=1}^{k} P_i$ where as the access cost of (k+1) th element of request sequence in the list is (k+1) because position of that (k+1) th element is (k+1) .So total access cost for the (k+1) element is $\sum_{i=1}^{k} P_i$ + (k+1).

So L.H.S=R.H.S

Hence C_{k+1} is true.

So the expression is true for all n.

IV. CONCLUSION

Here we have derived some formula for special request sequence these formula are proved by Induction principle.

V. ACKNOWLEGDEMENT:-

This work is during my M.TECH career under supervision of Rakesh Mohanty ,Lecturer ,Department of Computer Science and Engineering ,VSSUT ,Burla.

REFERENCES:-

[1] J. McCabe, "*On serial files with relocatable records*", Operational Research, vol. 12, pp.609-618, 1965.
[2] G. Jr. Schay, F.W. Dauer-"*A Probabilistic Model for a Self Organizing File System*", SIAM J.of Applied Mathematics, (15) 874-888, 1967.
[3] P.J Burvile and J.F.C. Kingman "*On a model for storage and search*" –J. of Applied Probability, 10:697-701, 1973.
[4] R. Rivest-"*On Self Organizing Sequential Search Heuristics*", Communications of the ACM, 19, 2:63-67, 1976.
[5] J.R. Bitner" *Heuristics that dynamically organize data structures*"- SIAM J. of Computing, 8(1):82-110, 1979.
[6] G.H. Gonet, J. I, Munro and H. Suwanda, "*Towards self organizing linear search*"-FOCS, 169-174, 1979.
[7] G.H. Gonet, J. I, Munro and H. Suwanda, "*Exegenesis of self organizing linear search*", SIAM Journal of Computing, Vol. 10, no.3, pp.613-637, 1981.
[8] D. D. Sleator and R. E. Tarjan, "*Amortized efficiency of list update paging rules*", Commun. ACM, vol. 28 no. 2, pp.202-208, 1985
[9] J. H. Hester, D. S. Hirschberg" *Self organizing linear search*" – ACM Computing Surveys, 17(3): 295-312, 1985.
[10] J.L. Bently and C. C. McGeoch, "*Amortized analysis of self-organizing sequential search heuristics*", CACM, vol. 28, pp. 404-411, 1985.
[11] Nick Reingold and Jeffery Westbrook, "*Optimum Off-line Algorithms for the List Updates Problems*"-Technical Report YALEU/DCS/TR-805, Yale University, 1-24, 1990.
[12] S. Ben David, A. Borodin and R. Karp," *On the Power of Randomization in Online Algorithms*"- In Algorithmic a, pages 379-386, 1990.
[13] Sandy Irani, **"***Two Results on the List Update Problem*"-IPL, 38(6):301-306, 1990.
[14] Boris Teia, "*A lower bound for randomized list updates algorithms*"- IPL 47:5–9, August 1993.
[15] Nick Reingold, Jeffery Westbrook, and Daniel D. Sleator,"*Randomized Competitive Algorithms for the List Update Problem*"-algorithmic a, 11:15-32, 1994.
[16] Susanne Albers,"*Improved Randomized On-Line Algorithms for the List Update Problem*"- SODA, 412-419, 1995.
[17] Susanne Albers, Bernhard von Stengel and Ralph Werchner.**"***A Combined BIT and TIMESTAMP Algorithm for the List Update Problem*" - IPL, 56:135–139, 1995.
[18] Rajeev Motwani, Prabhakar Raghavan, "Randomized Algoritms" -ACM Computing Surveys, 1996.
[19] Allan Borodin and Ran EI-Yaniv, "*On Randomization in Online Computation*"-STOC, 1997.
[20] S. Albers and M. Mitzenmacher," *Revisiting the COUNTER algorithms List Update*"-IPL, 64(2):155-160, 1997.
[21] Rakesh Mohanty,Burle Sharma and Sasmita Tripathy," Characterization of Request Sequences for List Accessing Problem and New Theoretical Results for MTF Algorithm" -*International Journal of Computer Applications (0975 – 8887)Volume 22– No.8, May 2011.*