Improvised Master's Theorem

Shashi Prakash Tripathi[1], Vaishali Srivastava[2], Harshita Rai[3]

[1]MCA, CCE, University of Allahabad, Uttar Pradesh, India
[2]MCA, CCE, University of Allahabad, Uttar Pradesh, India
[3]MCA, CCE, University of Allahabad, Uttar Pradesh, India

---***---

Abstract - *This paper analyses the existing representation of master's theorem which is based on the divide and conquer technique and the constraints or limitations of the existing method. The improvisation of the existing theorem that this paper presents tackles the problems where the preexisting theorem fails. In the proposed method we provide a new approach to analyze the recurrence relation of the type divide and conquer which in turns enables us to compute the complexity of recurrence relation without any constraints. This approach analyses the divide and conquer recurrence in order to improve in several aspects.*

Key Words: **Divide and Conquer, Recurrence Relation, Master's Theorem, Iteration Method, Asymptotic Notation, Substitution Method.**

1. INTRODUCTION

The proposed method enables us to solve those divide and conquer problems which the original theorem failed to solve in asymptotic time. These algorithm are based on multiple branching recursion [1]. It divides the problem recursively into sub problems that can be of the same or related type such that each sub problem is of the size n/b. The main problem is recursively subdivided until it becomes simple enough to be solved directly. The visualization of the sub problems can be done in as a call tree such that each node represents an instance of the recursive call and the child node represents the sub calls. Each node is divides into a number of sub nodes which does some processing in order to solve the problem and this determines the size of the sub problem n being passed to the subsequent recursive call and is denoted by f (n). Suppose we consider an example of a comparison sort, at each node of the tree the amount of work done will be at least equivalent to O (n log n). The total work done during the execution of the algorithm will be the sum of work done at each recursive call.

The recurrence relation $T(n) = aT(n/b) + f(n)$ [2] can represent the algorithm for the comparison sort mentioned above. In order to compute the expression for the total work done we can simply expand and substitute the given recurrence relation into itself. The total work done can also be computed without expansion if we use the original Master's method, the generalized form of the master's theorem as proposed by Akra and Bazzi in 1998 is also applicable.

When we are talking about recurrence relation, they can be categorized into two basic categories namely homogeneous and heterogeneous recurrence relation. The latter can be solved using three techniques: Substitution method where in the recurrence relation is initially guessed and then mathematical induction is used to show the correctness of the guessed relation. The use of valid statements and elicitations are made for proving the solution. The amount of work done depends largely on how accurate our initial guess is. If the initial guess is wrong, then it can be readjusted later. The second method is the iterative method wherein we use the concept of induction. In this method we expand the recurrence k time and then we prove for a couple of values in the series and then assumed for the nth value. The third method is the master's method which is a sure shot technique to compute the exact complexity of a recurrence relation. In order to use the master's method to solve a given recurrence relation it is necessary that the relation is in a specific format or has a predefined structure otherwise master's method is not applicable. In the next section we discuss the original master's theorem in detail.

2. BACKGROUND AND RELATED WORK

The master's method can be employed to compute the complexity of a given divide and conquer problem in asymptotic time [3] [4]. A divide and conquer problem we have three steps viz., divide, conquer and combine. In the divide step we start with the given problem and recursively divide it into two or more sub problems until we have a sub problem that can be directly solved. In the conquer step, the sub problem of the desirable size that we have achieved in the divide step is solved. The combine step involves the amalgamation of all the solutions of the sub problems achieved in the previous step. The form of the recurrence relation of the master's theorem is:

$$T\,(n) \;=\; aT\,(n/b) \;+\; f\,(n),$$
$$Where\ a\ >=\ 1, b\ >=\ 1$$

The constants and function in the above equation represents the following:

- n is the problem size.
- a represents the number of sub problems in which the original problem is being divided.
- n/b is the size of each sub problem. (Here it is assumed that all sub problems are essentially the same size.)
- f (n) is the cost of the work done outside the recursive calls, which includes the cost of dividing the problem and the cost of merging the solutions to the sub problems[5][6].It is possible to determine an asymptotic tight bound in these three cases:

2.1 If f(n)=O $n^{\log_b a - \varepsilon}$ for some constant ε> 0, then
T (n) = Θ ($n^{\log_b a}$).

2.2 If f(n) = Θ ($n^{\log_b a}$) with k >= 0, then
T (n) = Θ ($n^{\log_b a}\ log\ n$).

2.3 If f(n) = $\Omega(n^{\log_b a+\varepsilon} + \varepsilon)$ with ε> 0, and f(n) satisfies the regularity condition, then
T (n) = Θ (f(n)).

Regularity condition: $a\,f(n/b)\ <=\ cf(n)$ for some constant c < 1 and all sufficiently large n.

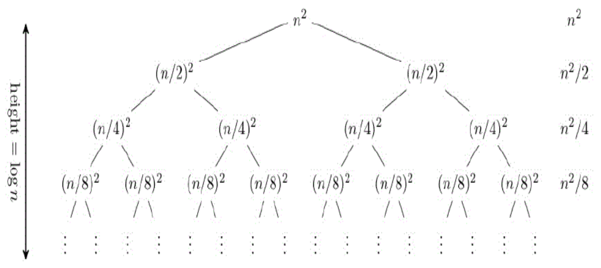

Figure 1: Divide and Conquer Problem of $T(n) = 2T(n/2) + n^2$

Fig -1: **Divide and Conquer Problem**

Consider the following problem:

$$T\,(n) \;=\; 2T\,(n/2) \;+\; n^2$$

The above recurrence ε > 0.

3. PROBLEM STATEMENT

The earlier used theorem has some limitations. The following equations violate the constraints of the original master's theorem and thus cannot be solved using the same: [7] [8]

3.1 $T(n) = 2^n\,T\left(\frac{n}{2}\right) + n^n$

a is not a constant; the number of sub-problems should be fixed.

3.2 $T(n) = 2T\,\left(\frac{n}{2}\right) + n/\log n$

Non-polynomial difference between f(n) and $n\log_b a$

3.3 $T(n) = 0.5\,T\left(\frac{n}{2}\right) + n$

a < 1 cannot have less than one sub problem.
Figure 1: Divide and Conquer Problem of $T(n)\ =\ 2T(n/2)\ +\ n^2$

3.4 $T(n) = 64T\,\left(\frac{n}{8}\right) + n^2\log n$

f(n) which is the combination time is not positive.

3.5 $T(n) = T\left(\frac{n}{2}\right) + n(2 - \cos n)$

The above equation lies in third case of Master's theorem but regularity of the equation is violation.
In the second equation above, the difference between f(n) and $n^{\log_b a}$ can be expressed with the ratio $\frac{f(n)}{n^{\log_b a}} = \frac{n}{nlogn} = \frac{1}{logn}$. It is clear that $1/\log n < n^{\varepsilon}$ for any constant $\varepsilon > 0$. Therefore, the difference is not polynomial and the Master Theorem does not apply.

4. SOLUTION

$$T(n)\ = aT\left(\frac{n}{b}\right) + \theta(n^k log^p n$$

$(a \geq 1, b \geq 1, k \geq 0\ and\ p\ is\ real\ number)$

4.1 $If\ a > b^k, then\ T(n)\ = \theta(n^{\log_b a}$

4.2 $If\ a = b^k$

4.2.1 $If\ p > -1, then\ T(n)\ = \theta(n^{\log_b a} log^{p+1} n)$

4.2.2 $If\ p = -1, then\ T(n)\ = \theta(n^{\log_b a} loglog\ n)$

4.2.3 $If\ p < -1, then\ T(n)\ = \theta(n^{\log_b a})$

4.3 $If\ a < b^k$

4.3.1 $If\ p \geq 0, then\ T(n)\ = \theta(n^k log^p\ n)$

4.3.2 $If\ p < 0, then\ T(n)\ = O\ (n^k)$

The above solution is able to solve the problem like:

The iterative approach to solve this problem is as follows:

$T(n) = 2T(n/2) + n/\log n$

$<= 2T(n/2) - n \log n$

$T(n) = 2T(n/2) - \Theta(n \log n)$

Using iteration method, we get

$$2T(n/2) = \sum_{i=0}^{k-1} 2^i 2^{k-1}(k-i)\log_2 2$$

$$= 2^k \sum_{i=0}^{k-1} k - i)$$

$= 2^{k-1}k(k+1)$

$As\ k = \log n$

$So\ T(n) = \Theta(n \log^2 n)$

Now solution by our proposed method:

Let's compare the following equation with proposed format you get:

a = 2, b = 2, k = 1, p = -1;

It satisfies the second case because

$a = b^k$

$2 = 2^1$ and $p = -1$

Therefore the solution is $\Theta(n \log\log n)$

$T(n) = 64\,T(n/8) + n^2 \log n$

Now compare the following equation with proposed format you get:

$a = 64, b = 8, k = 2, p = 1;$

It satisfies the second case because

$a = b^k$

$64 = 8^2$ and $p = 1$

Therefore the solution is $\Theta(n^2 \log^2 n)$.

Conclusion and Future work

The improvised master's theorem proposed in this paper can be used to solve a wider range of problems. It overcomes the constraints of the original method by giving us the relaxation of the floor and ceiling, as illustrated by the examples above the equations (A, B, C, D and E) could not be solved using the original master's method but after the relaxation of the constraints they can be solved using the new approach. It also helps us to reduce the time complexity in comparison if the relation is solved using iterative and substitution method.

The proposed method is able to relax some of the constraints of the original method, in the future work we would like to deal with the other constraints like the set structure of the equations etc. so that the wider range of problems can be dealt with.

References

[1] D.R. Smith, The structure of divide and conquer algorithms, Technical Report NPS 52-83-002, Naval Postgraduate School, Monterey, CA (1983)

[2] Koshy, Thomas. "Solving Recurrence Relations." *Fibonacci and Lucas Numbers with Applications* (2001): 142-146.

[3] M. Akra and L. Bazzi, "On the Solution of Linear Recurrence Equations" ,Computational Optimization and Applications, 10(2):195-210, 1998.

[4] Hyvärinen, Aapo. "Fast and robust fixed-point algorithms for independent component analysis." *Neural Networks, IEEE Transactions on* 10.3 (1999): 626-634.

[5] T. H. Cormen, C. E. Leiserson and R. L. Rivest, Introduction to algorithms, MIT Press, 1990.

[6] S. Dube, "Using Fractal Geometry for Solving Divide-and-Conquer Recurrences," Jour. of Australian Math. Soc. Series B37, 145-171,1995.

[7] T. Leighton, "Notes on Better Master Theorems for Divide-and-Conquer Recurrences," Manuscript. Massachusetts Institute of Technology, 1996.

[8] B. Mandelbrot, "The Fractal Geometry of Nature", W. H. Freeman and Co., 1982.

[9] S. Roura, "Improved Master Theorems for Divide-and-Conquer Recurrences ",Jour. of ACM 48, 170-205, 200