Research on Optimization Algorithm Design Techniques

Parul Sood

Abstract—This paper discusses various optimization algorithm design techniques. So, optimization techniques which are discussed in this paper are greedy method, dynamic programming and branch and bound. Problem comes under optimization are used to find either maximum or minimum. All these techniques we have multiple inputs and some constraints and we have to find feasible solution using these inputs and constraints. In greedy method we follow some predefined method. Using that predefined method, we reach to the solution. On contrary to this in dynamic programming we take decision at every step and in the end we reach to the solution. In branch and bound we create state space tree and explore all possibilities of live node. Based on some constraint we start killing some alive nodes. Here, firstly I will discuss all the design techniques. Then types of problems that can be solved under each design techniques and their time complexities.

Keywords: greedy method, dynamic programming, branch and bound, optimization, algorithm design techniques, maximum and minimum problems.

I. INTRODUCTION

GREEDY METHOD is simplest design technique which is used to solve wide variety of applications. We have give number of inputs and some constraints. So based on input and constraints we have to find optimal solution. Greedy method follows one decision at every step and according to that decision it either going to include that particular input in solution or not [1].

General Greedy Method Algorithm[5]
Algorithm GreedyMethod(int array , int elements)
{
    OptimalSolution = {}
    For I = 0 to elements
    {
        Y= select (array)
        If(fesible(OptimalSolution,Y))
            OptimalSolution = OptimalSolution+Y
        endIf
    }
    return OptimalSolution
}

DYNAMIC PROGRAMMING is another algorithm design technique in which we take decision at every step. Problems under dynamic programming can be solved using either bottom up or top down approach. Bottom up approach is known as tabulation method and top down approach is also known as memorization [2].

General Dynamic Programming Algorithm[6]
Algorithm Bottom Up Dynamic Programming ()
{
    for each input define the structure of optimal solution recursively fill the value in that structure compute value of optimal solution using backtracking
}

BRANCH & BOUND is another algorithm design technique for solving optimization problems but this is somehow similar to backtracking like backtracking we have to create state space tree for every possible alive node and it follows BFS contrary to it backtracking algorithm use DFS
[3].

General Branch and Bound Algorithm[4]
Algorithm LeastCostsB&B(int n)
{
    For each node I:= n
    While(1) do
        If (I is a leaf) then
            I is optimal solution and print the path from I to leaf
            return
        endIf
        Expand(I)
        If(I is empty) then
            There is no solution
            Return
        endIf
    endwhile
}

II. ANALYSIS OF VARIOUS DESIGN TECHNIQUES

<table>
<thead>
<tr>
<th>Method</th>
<th>Dynamic Programming</th>
</tr>
</thead>
<tbody>
<tr>
<td>Based on predefined decisions. For example, in kruskal algorithm we always select node with minimum cost and we keep doing this until we traversed all nodes and reach the result.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Branch &amp;bound</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Based upon state space tree. For example, 0/1 knapsack we draw state space tree by adding first item and by not adding first item. At every step we check for bounding function. We keep making until we find the solution.</td>
<td></td>
</tr>
</tbody>
</table>

Revised Version Manuscript Received on September 13, 2019.
Parul Sood, Department of Computer Science and Engineering Chandigarh University Gharuan, India. E-mail: Parule7206@cumail.in
These algorithms have greedy local choice property. These algorithms achieve local optimization then only can achieve global optimization.\[8\]

| These algorithms are based on recursion which is used smartly here we store results of recursion in cache. Then we will solve all dependent problems using those results and lead to optimal solution. | These algorithms are based on BFS or DFS, these are used to create branches at every level or bounding function is applied on each branch to find the optimal solution. |
| Properties of greedy method are optimal substructure which is optimal at every step.\[9\] | Properties of branch and bound are generation on state space tree and apply bounding function. |
| Problem solved by greedy method are Minimum Spanning tree, Fractional knapsack and Huffman coding etc. | Problem solved by branch and bound are 0/1 knapsack and travelling salesman etc. |
| Problems can not be solved by greedy method are 0/1 knapsack. | Problems can’t be solved by dynamic programming are longest path |
| These are top down approach | This is bottom up approach |
| It makes maximum n choices for input size, So time complexity is \(O(n^2)\) (choice \(n\)) | It solves problem by considering it takes \(n\) states and time taken by each state. We can write it as states*time taken by each state. |
| It solves problems with not known polynomial time. In most cases complexity grows exponentially. |

### III. CONCLUSION

AS above I have discussed various algorithm design techniques for optimization algorithms. As far as complexity wise best algorithm technique is greedy method and on other hand maximum time required in case of branch and bound. If comparison made on based on type of problem solved, then best is branch and bound as where dynamic and greedy method fails their branch and bound works. On other hand greedy is used to solve very small range of problems when we try to find local optimal in end which results into global optimal. So in the end I would like to say selection of algorithm technique is depending of requirement of user.

### REFERENCES