Impact Factor (SJIF): 2.125

International Journal of Advance Research in Engineering, Science & Technology

e-ISSN: 2393-9877, p-ISSN: 2394-2444

Volume 3, Issue 1, January-2016

DIJKSTRA GREEDY - DYNAMIC PROGRAMMING ALGORITHM

Nikita R. Patel[1], Bijal N. Dalwadi[2]

[1,2]Assistant Professor Information Technology Department

BVM Engineering College

V.V.Nagar, Anand Gujarat-India

Abstract

Dijkstra's Algorithm is one of the popular algorithm for finding shortest path in computer science. As well it is also popular in operations research. It is generally presented and viewed as a greedy algorithm. In this Article we attempt to change this perception by providing a dynamic programming perspective on the algorithm. This Article present Dijkstra's is a greedy as well dynamic programming algorithm.

Keywords: *Principle of Optimality; Dijkstra's; Greedy programming; Dynamic programming;*

I. INTRODUCTION

A greedy algorithm is one that at a given point in time, makes a local optimization. Dynamic programming can be thought of as 'smart' recursion. It often requires one to break down a problem into smaller components that can be cached. Greedy is indeed a special case of Dynamic Programming. In dynamic programming a set of solutions to smaller problems are derived and pick the best among them. While in greedy algorithm, one of the solutions to smaller problems to be the optimal one (depending on certain properties of the problem) and don't examine other solutions. Suppose Given an array of integers, to find the longest zigzag sub array (which may be non-contagious). A naive dynamic programming solution would be to store the longest zigzag sub array ending at the i^{th} position in $d_{[i]}$.Then use the recurrence relation, d[i+1] = 1 + max{d[j] : j <= i and zigzag property is maintained}. This runs in O (n^2) time. A greedy will start from the first index and move forward to include a point where the 'derivative' changes sign. This will run in O (n) time. The choice to pick and believe one sub-solution ('derivative' change point) to be optimal saves a lot of time here, but is essentially just dynamic programming with sub-solution sample space pruned.

II. RELATED RESEARCH

Let's take example of one graph.

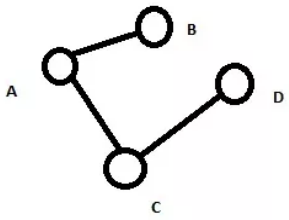

Fig: 1.1 Graph

Where, AB = 1, AC = 2, CD = 4.Suppose we need to compute the shortest distance between A and D. A dynamic programming algorithm solves sub-problems. So, a sub-problem would be finding the shortest distance to a node that lies in a path between A and D. But Dijkstra's doesn't do that.

Dynamic programming is focus on the **Principle of Optimality**, for any problem solution is optimal then solution of sub-problem is also optimal. Whereas, the *greedy technique* focuses on expanding partially constructed solutions until you arrive at a solution for a complete problem. It is then said, it must be "the best local choice among all feasible choices available on that step".

If we consider the problem " For make change with dollars, pound, and rupee." This is a greedy problem. It exhibits optimal substructure because you can solve for the number of dollars. Then, solve for the number of

pound. Then the number of rupee. If you combine the solutions to these sub problems is also optimal. Now consider the problem "Fibonacci numbers." It exhibits optimal substructure because you can solve F(11) from F(10) and F(9) efficiently (by addition). These sub problems overlap because they both share F(8). If the result of F(8) when solving F(9), you can solve F(10) more quickly, as about dynamic programming having to do by reconsidering decisions: This is not true for any linear dynamic programming algorithm like the maximizing sub array problem or the Fibonacci problem. Suppose a problem having optimal substructure as a directed acyclic graph whose nodes represent sub problems and whose directed edges represent dependencies between sub problems. Then, a greedy problem is a tree (all nodes except the root have unit in degree). A dynamic programming problem has some nodes with in-degree greater than one. This illustrates the overlapping sub problems.

III. COMPARISION WITH EXISTING WORK

Dynamic Programming is focus on the Principle of desirability, "An optimal solution to any instance of an optimization problem is composed of optimal solutions to its public abstract class". Whereas, the greedy technique focuses on expanding partially constructed solutions until you arrive at a solution for a complete problem. It is then said, it must be "the best local choice among all feasible choices available on that step.

In Dijkstra's Algorithm, we start from a single node and find the shortest path to each of the other nodes in every iteration. If we start from a node A and reach node C via node B, we are in a way using the already solved sub-problem to compute the shortest path to node C.

Dijkstra's is a DP, and it doesn't make sense to think of the algorithm as greedy. The sub problems here are the computed shortest paths to intermediate vertices. If you were to try and view this as a greedy, choosing a locally optimal solution in the context of a general graph doesn't make much sense. While it is crystal clear that DA is commonly regarded as a greedy algorithm: it deploys a greedy rule to determine what city should be processed next. I guess you can view the algorithm as greedy in that light, although that's a little weird. It's greedy because you always mark the closest vertex. It's dynamic because distances are updated using previously calculated values.

Dijkstra’s Algorithm is a greedy approach at each step, chose the node that the path to it is currently minimized - the choice is done greedily based on the local state of the algorithm. Where Dijkstra's is an example of dynamic programming even by definition: the sub problem being solved is the distance from the root function applied to each node. There are even three references on the Wikipedia page you linked that suggest the same thing.

Greedy method focuses on expanding partially constructed solutions, it provides many results such as feasible solution, more efficient while Dynamic programming: Focuses on principle of optimality, it provides specific answers, less efficient. So Dijkstra’s is Greedy method.

It is a dynamic programming algorithm. We are storing the values of distance from source and once a node is visited, we would not be considering it further. This is nothing but storing the distance to a particular node and not changing it. We can hence say that there is no duplication. Dijkstra’s algorithm can be considered as finding the distance to a destination in a bottom up fashion. Here the only difference is we are finding the distance from source and storing it in the matrix.

IV. CONCLUSION

Dijkstra's algorithm is a greedy algorithm because when we get a problem, from the initial point we choose the best possible way just looking at the initial point we choose the best path and follow it and then we do the same with that point. We are not concerned with the distance of other vertex which we might encounter while applying the Dijkstra’s algorithm. Using Greedy approach make decision about which edge to merge in every iteration. It uses memorization (DP) to actually compute the solution using a solution to a sub-problem. Where as in Dynamic programming algorithm might analyse the complete path first, that is it would take whole graph look at each combination of possible shortest path and then would have given a solution.

So, Dijkstra’s algorithm is combination of both - greedy and Dynamic Programming.

REFERENCES

[1] “A Review And Evaluations Of Shortest Path Algorithms” Kairanbay Magzhan, Hajar Mat Jani, International Journal Of Scientific & Technology Research , ISSN 2277-8616

[2] Benjamin Zhan F. Three Fastest Shortest Path Algorithms On Real Road Networks. Journal Of Geographic Information And Decision Analysis, 1997, 1 (1): 69-82.

[3] Dijkstra’s Algorithm Revisited: Dynamic Programming Connexion By Moshe Sniedovich, Control And Cybernetics, Vol. 35 (2006) No. 3

[4]Dijkstra's Algorithm, Available at Http://Informatics.Mccme.Ru/Moodle/Mod/Statements/ View.Php?Id=193#1. 2012.
[5] Greedy Algorithm Annu Malik, Anju Sharma, Mr. Vinod Saroha (Guide) International Journal of Scientific and Research Publications, ISSN 2250-3153
[6] Research On Improved Dijkstra Algorithm Using For Safety Management In The
Road Network, FAN J U N, GUO Zhong-Yin, Yuan Hong
[7] Algorithms Design and Analysis by Udit Agarwal