The Analysis of Forward and Backward Dynamic Programming for Multistage Graph

Anna Angela Sitinjak¹, Elvina Pasaribu², Justin E Simarmata³, Tedy Putra⁴, Herman Mawengkang⁵
Department of Mathematics, University of Sumatera Utara, Indonesia

Email:
¹annaangelasitinjak@yahoo.co.id
²elvina.pa1000@yahoo.com
³justinesimarmata@yahoo.com
⁴iqrastabat2015@gmail.com
⁵hmawengkang@yahoo.com

Abstract. Dynamic programming is an optimization approach that divides the complex problems into the simple sequences of problems in which they are interrelated leading to decisions. In the dynamic programming, there is no standard formula that can be used to make a certain formulation. In this paper we use forward and backward method to find path which have the minimum cost and to know whether they make the same final decision. Convert the problem into several successive sequential stages starting on from stages 1,2,3 and 4 for forward dynamic programming and the step back from stage 4,3,2,1 for backward dynamic programming and interconnected with a decision rule in each stage. Find the optimal solution with cost principle at next stage. Based on the characteristics of the dynamic programming, the case is divided into several stages and the decision is has to be made (xₖ) at each stage. The results obtained at a stage are used for the states in the next stage so that at the forward stage 1, f₁(s) is obtained and used as a consideration of the decision in the next stage. In the backward, used firstly stage 4, f₄(s) is obtained and used as a consideration of the decision in the next stage. Cost forward and backward always increase steadily, because the cost in the next stage depends on the cost in the previous stage and formed the decision of each stage by taking the smallest fₖ value. Therefore the forward and backward approaches have the same result.

1. Introduction
Mathematical problems cannot be separated from problems with complex or simple categories with various types of solutions. One of complex problem solving is with a dynamic programming. A dynamic programming is an optimization approach that turns complex problems into simple problems and form stages, each stage makes decisions so that form a set of interrelated and consecutive decisions that lead to the final decision. Therefore dynamic programming is known by its characteristic that is multistage on optimization procedure. In the dynamic programming, there is no standard formula that can be used so it needs to make a certain formulation. There are 2 types of problem solving procedures that are forward and backward dynamic programming.

2. Theoretical Framework
Dynamic program is an optimization approach used to solve problems that often occur in daily-life. The problem of dynamic program has characteristics is (Robert Andreani in www.ime.unicamp.br/~andreani/MS515/capitulo7.pdf access in May 30,2017):
1. The problem is divided into several stages by making decisions at each stage.
2. Each stage consists of a number of states and states that correspond to that stage.
3. The effect of the policy decision at each stage is to transform the current state to a state associated with the beginning of the next stage (possibly according to a probability distribution).
4. The solution procedure is designed to find an optimal policy for the overall problem.
5. Given the current state, an optimal policy for the remaining stages is independent of the policy decisions adopted in previous stages.
6. The solution procedure begins by finding the optimal policy for the last stage.
7. A recursive relationship that identifies the optimal policy for stage \( n \), given the optimal policy for stage \( n-1 \), is available.

Dynamic program has sequences of decisions so that to form an optimal decision used the optimal policy that is “in an optimal sequence of decisions or choices, each subsequence must also be optimal” (Puntambekar.2010:4-2). This means that we can use the optimal solution of a stage without having to go back to the initial stage for the process from one stage to the next stage.

One of the problems that can be solved with a dynamic programming is the multistage graph problem. Multistage Graph \( G = (V, E) \) is a graph with special properties (Puntambekar.2010:4-3):
1. Directed graph
2. All vertices are partitioned into the \( k \) stages where \( k \geq 2 \)
3. Each edge has weight
4. There is only 1 source (let S) and 1 sink (let T)
5. The path from source to sink consists of several stages with each stage has a set of vertices.
6. Its problem is to find the path from S to T (minimum or maximum cost).

Although there is no standard formula for solving a problem with a dynamic programming, it has fixed method namely forward and backward methods.

![Figure 1.](image)

To solve the multistage graph problem using several stages:
1. Convert the problem to some successive stages which for forward dynamic programming to move from stage 1, 2, 3, and so on until stage \( n \) and move back from stage \( n, n-1 \), to stage 1 for backward dynamic programming.
2. Complete the last stage by connecting all the previous stages of decisions.
3. Find the optimal solution with the cost principle at \( k + 1 = (\text{cost generated at k step}) + (\text{cost from stage k to stage k + 1}) \) with \( k = 1, 2, ..., n-1 \) for forward dynamic Programming and at stage \( k = (\text{cost generated at stage k + 1}) + (\text{cost from stage k + 1 to stage k}) \) with \( k = n, n-1, ..., 1 \) for backward dynamic programming (Munir in Ika Zulhidayati.2013:26).
3. Discussion.
Find the minimum cost for multistage graph problem by using dynamic programming (web.mit.edu/15.053/www/AMP-Chapter-11.pdf accessed in May 2, 2017).

![Diagram of multistage graph problem](image)

**Figure 2.**

**a. Forward Dynamic Programming**

\[
f_1(s) = c_{x_1 s} \text{ (initial state)}
\]

\[
f_k(s) = \min_{x_k} \left\{ c_{x_k s} + f_{k-1}(x_k) \right\} \quad \text{(recurrence)}
\]

with \( k = 2, 3, 4 \)

where :

- \( c_{x_k s} \) = weight
- \( f_{k-1}(x_k) \) = total weight
- \( f_k(s) \) = minimum value

final state: \( f_4(10) \) with looking for \( f_1(s), f_2(s), f_3(s) \) firstly

stage 1 :

\[ f_1(s) = c_{x_1 s} \]

**Table 1.**

| \( s \) | Optimal Solution |
|--------|------------------|
| \( f_1(s) \) | \( x_1 \) |
| 2      | 80               | Vertex 1 |
| 3      | 70               | Vertex 1 |
| 4      | 50               | Vertex 1 |
Stage 2:
\[ f_2(s) = \min_{x_2 \in X_2} \{ c_{x_2s} + f_1(x_2) \} \]

| s | \( f_2(x_2,s) = c_{x_2s} + f_1(x_2) \) | Optimal Solution |
|---|---|---|
| 2 | 3 | 4 | \( f_2(s) \) | \( x_2^* \) |
| 5 | 130 | 195 | 130 | 130 | \( 2 \) or \( 4 \) |
| 6 | 155 | 170 | 90 | 0 | 4 |
| 7 | 210 | 170 | 80 | 80 | 4 |

Stage 3:
\[ f_3(s) = \min_{x_3 \in X_3} \{ c_{x_3s} + f_2(x_3) \} \]

| s | \( f_3(x_3,s) = c_{x_3s} + f_2(x_3) \) | Optimal Solution |
|---|---|---|
| 5 | 6 | 7 | \( f_3(s) \) | \( x_3^* \) |
| 8 | 290 | 180 | 260 | 180 | 6 |
| 9 | 190 | 240 | 230 | 190 | 5 |

Stage 4:
\[ f_4(s) = \min_{x_4 \in X_4} \{ c_{x_4s} + f_3(x_4) \} \]

| s | \( f_4(x_4,s) = c_{x_4s} + f_3(x_4) \) | Optimal Solution |
|---|---|---|
| 8 | 9 | \( f_4(s) \) | \( x_4^* \) |
| 10 | 355 | 350 | 350 | 9 |

The paths that can be reached with minimum cost are:

a. \( 1-2-5-9-10 = 350 \)
b. \( 1-4-5-9-10 = 350 \)

b. **Backward Dynamic Programming**

\[ f_4(s) = c_{x_4s} \text{ (initial state)} \]

\[ f_k(s) = \min_{x_k \in X_k} \{ c_{x_k} + f_{k+1}(x_k) \} \quad \text{(recurrence)} \]

with \( k = 1,2,3 \),

where:

\( x_k \) = decision variable at the stage \( k (k=1,2,3) \)
$c_{sk}$ = cost dari $s$ ke $x_k$

$f_{s,k}(s,x_k)$ = total cost of path $s$ to $x_k$

$f_k(s) = minimum value of f_k(s,x_k)$

Stage 4:

$f_4(s) = c_{sx4}$

| $s$ | Optimal Solution |
|-----|------------------|
| 8   | 175              |
| 9   | 160              |

Stage 3:

$f_3(s) = \min_{x3} \{ c_{sx3} + f_4(x_3) \}$

| $x_3$ | $s$ | $f_3(s) = c_{sx3} + f_4(x_3)$ | Optimal Solution |
|-------|-----|--------------------------------|------------------|
| 5     | 8   | 335                            | 9                |
|       | 9   | 220                            |                  |
| 6     | 3   | 265                            | 310              |
|       | 9   | 310                            | 6                |
| 7     | 3   | 355                            | 310              |
|       | 9   | 310                            | 9                |

Stage 2:

$f_2(s) = \min_{x2} \{ c_{sx2} + f_3(x_2) \}$

| $x_2$ | $s$ | $f_2(s) = c_{sx2} + f_3(x_2)$ | Optimal Solution |
|-------|-----|--------------------------------|------------------|
| 2     | 5   | 270                            | 5                |
|       | 6   | 385                            |                  |
|       | 7   | 440                            |                  |
| 3     | 5   | 345                            | 410              |
|       | 6   | 410                            | 410              |
|       | 7   | 410                            |                  |
| 4     | 5   | 300                            | 340              |
|       | 6   | 350                            | 300              |
|       | 7   | 300                            | 5                |

Stage 1:

$f_1(s) = \min_{x1} \{ c_{sx1} + f_2(x_1) \}$

| $x_1$ | $s$ | $f_1(s) = c_{sx1} + f_2(x_1)$ | Optimal Solution |
|-------|-----|--------------------------------|------------------|
| 2     | 3   | 350                            | 4                |
| 4     | 3   | 350                            | 2 or 4           |

Paths with minimum cost are:

a. 1-2-5-9-10 dengan cost 350

b. 1-4-5-9-10 dengan cost 350
Figure 4.

Based on the optimal solution characteristics of the dynamic programming and the result of the case, it is found that the problem is divided into several stages and taken decision (x_k) at each stage. In this problem, it is made into 4 stages and there are decisions at each stage either forward or backward, where each stage of the problem has a number of states. These decisions are interconnected to form a final decision.

The result obtained at a stage is used for the states in the next stage so that in the forward stage 1, f_1(s) is obtained and used as a consideration of the decision modifiers in the next stage and at the backward in stage 4, f_4(s) is obtained and used as consideration of the decision modifier at next stage. This is done until the last stage. In this case only think about cost without thinking of the problem about time and others. Cost in forward and backward always increase steadily, because the cost in the next stage depends on the cost in the previous stage and formed the decision of each stage by taking the smallest value of f_k where optimal decision making follows the principle of optimality. So the recurrence relation is different both the forward and backward but it leads to the same path. Therefore the forward and backward have the same result.

4. Conclusion
Forward dynamic programming has the optimum principle of k + 1 = (cost generated at stage k) + (cost from stage k to stage k + 1) with k = 1,2, ..., N-1. While the principle of backward dynamic programming optimization is cost at the stage k = (cost generated at stage k + 1) + (cost from stage k + 1 to stage k) with k = n, n-1, ..., 1. Although the principle of optimizations both of them are different, but forward and backward dynamic programming produce the same path that is 1-2-5-9-10 and 1-4-5-9-10 with cost 350.

References
[1] Nagy G, Salhi S. 2007. Location-routing: Issues, models and methods. European Journal of Operational Research, 177, 649-672.
[2] Andreani, Robert in www.ime.unicamp.br/~andreani/MS515/capitulo7.pdf access in May 30, 2015
[3] Puntambekar,A.A., (2010), Design and Analysis of Algorithm, Technical Publication, India.
[4] Zulhidayati, Ika, (2013), Aplikasi Algoritma Greedy dan Program Dinamis (Dynamic Programming) Pada Permainan Greedy Spiders, Skripsi, FMIPA, UPI.
[5] web.mit.edu/15.053/www/AMP-Chapter-11.pdf accessed in May 2,2017.