IJCSE International Journal of Computer Science and Engineering Open Access
Research Paper Volume-2, Issue-2 E-ISSN: 2347-2693

Shortest Route Algorithm Using Dynamic Programming by Forward Recursion

N. Karthikeyan

Dept. of Mathematics, Kongunadu College of Engineering and Technology, India
Email: karthi.lect@gmail.com

www.ijcaonline.org

Received: 22 Jan 2014 Revised: 08 Feb 2014 Accepted: 26 Feb 2014 Published: 28 Feb 2014

Abstract— The current widespread use of location-based services and Global Positioning System technologies has revived interest in very fast and scalable dynamic shortest path queries. We introduce a new shortest path query type in which dynamic constraints may be placed on the allowable set of edges that can appear on a valid forward recursion shortest path. Computing the shortest path between two given locations in a road network is an important problem that finds applications in various map services and commercial navigation products. Our experimental results reveal the characteristics of different techniques, based on which we provide guidelines on selecting appropriate methods for various scenarios. Although the raw data about geography and roads may be more readily available today, computing forward shortest paths is still not trivial. Dynamic algorithm allows us to compute point-to-point dynamic shortest path queries on any road network in essentially linear time. . In a preprocessing stage, these heuristics compute some auxiliary data, such as additional edges (shortcuts) and labels or values associated with vertices or edges.

Index Term— Dynamic Programming, Forward Recursion, Shortest Route, Stage i, State i, Minimum Paths, Backward Recursion

I. INTRODUCTION

Dynamic programming (DP) determines the optimum solution of a multivariable problem by decomposing it into stages, each stage comprising a single variable sub problem. The advantage of the decomposition is that the optimization process at each stage involves one variable only, a simpler task computationally then dealing with all the variables simultaneously. A DP model is basically a recursive equation linking the different stages of the problem in a manner that guarantees that each stage's optimal feasible solution is also optimal and feasible for entire problem.

Suppose that we want to select the shortest highway route between two cities. The following route network provides the possible routes between the starting city at node A and the destination city at node U. The routes pass through intermediate cities designated by different stages. All the stages its distance is measured and write in the route network (multiple of hundred).

II. DEFINITIONS

STAGE i
A stage signify a portion of total problems for which a decision can be taken at each stage there are no of alternatives and best out of those is called as stage decision which may not be optimal for the stage but the contribute to obtain the optimal decision policy.

STATE i
The condition of the decision process at a stage is called its state.

RECURSIVE EQUATION
let $f_i(x_i)$ be the shortest distance to node x_i at stage i, and define $d(x_{i-1}, x_i)$ as the distance from the node x_{i-1} to node x_i : then f_i is computed from f_{i-1} using the following forward recursive equation.

$$f_i(x_i) = \min_{\substack{\text{all feasible} \\ (x_{i-1}, x_i) \text{ routes}}} \{d(x_{i-1}, x_i) + f_{i-1}(x_{i-1})\},\ i = 1, 2, 3...$$

Starting at $i = 1$, the recursion sets $f_0(x_0) = 0$. The equation shows that the shortest distances $f_i(x_i)$ at stage i, must be expressed in terms of the next node x_i. In the DP terminology, x_i is referred to as state of the system at stage i. In effect, the state of the system at stage i, is the information that links the stages together, so that optimal decisions for the remaining stages can be made without reexamining how the decisions for the previous stages are reached. The proper definition of the state allows us to consider each stage separately and guarantee that the solution is feasible for all the stages. The definition of the state leads to the following unifying framework for DP.

SHORTEST ROUTE ALGORITHM

To find the path of minimum distance between the point s (source) to t (sink) using forward recursion of dynamic programming DP can be obtained using the following steps.

Step 1: identify the decision variables and specify objective function to be optimized for dynamic networks.

Step 2: start by assigning the notation x_i $(i=1,2,..n)$ to the decision variables connected with each of the cites.

Step 3: decompose the network into a number of smaller sub intervals. Identify the stage variable at each stage and write down the transformation function as a function of the state variable and decision variable at the next stage.

Step 4: represent each x_i as stage $(i+1)$, where $S(i+1)$ denotes the distance between x_i and x_{i+1}

Step 5: assign the initial value of the network as zero. (i.e) the value of the stage $f_0(x_0)=0$

Step 6: write down a general recursive relationship using the forward dynamic programming recursion as follows.

$$f_i(x_i)= \min_{\text{all feasible routes}(x_{i-1},x_i)} \{d(x_{i-1},x_i)+f_{i-1}(x_{i-1})\},\ i=1,2,3...$$

Step 7: determine the overall optimal decision or policy and its value at each stage of an networks.

IV. PROBLEM DEFINITION

We shall illustrate the technique with a simple example and provide the mathematical verification.

Transport options from Chennai to kanyakumari

For shortest route between the points Chennai and kanyakumari by bus with forward dynamic programming, consider the tamilnadu map and point some cities and consider them as vertices. Point A (Chennai), located at initial stage, leads towards 7 stages with the end point U (Kanyakumari). Here we consider each city as vertex and distance between any two cities is represented as edge. In order to develop the above problem in terms of forward recursion, the distances between any two cities are considered in hundreds of units.

In order to identify the city involved in this problem, let us adopt alphabet A - Chennai, B - Vellore, C - Chengalpattu, D - Villupuram, E - Thiruvanamalai, F - Dharmapuri, G - Perambalur, H - Namakal, I - Erode, J - Ooty, K - Thanjavur, L - Trichy, M - Coimbature, N - Pudukottai, O - Dinukkal, P - Kodaikanal, Q - Ramanathapurm, R - Madurai, S - Tuticorin, T - Thirunelveli, and U - Kanyakumari.

Fig: 4.1 Map of Tamil Nadu

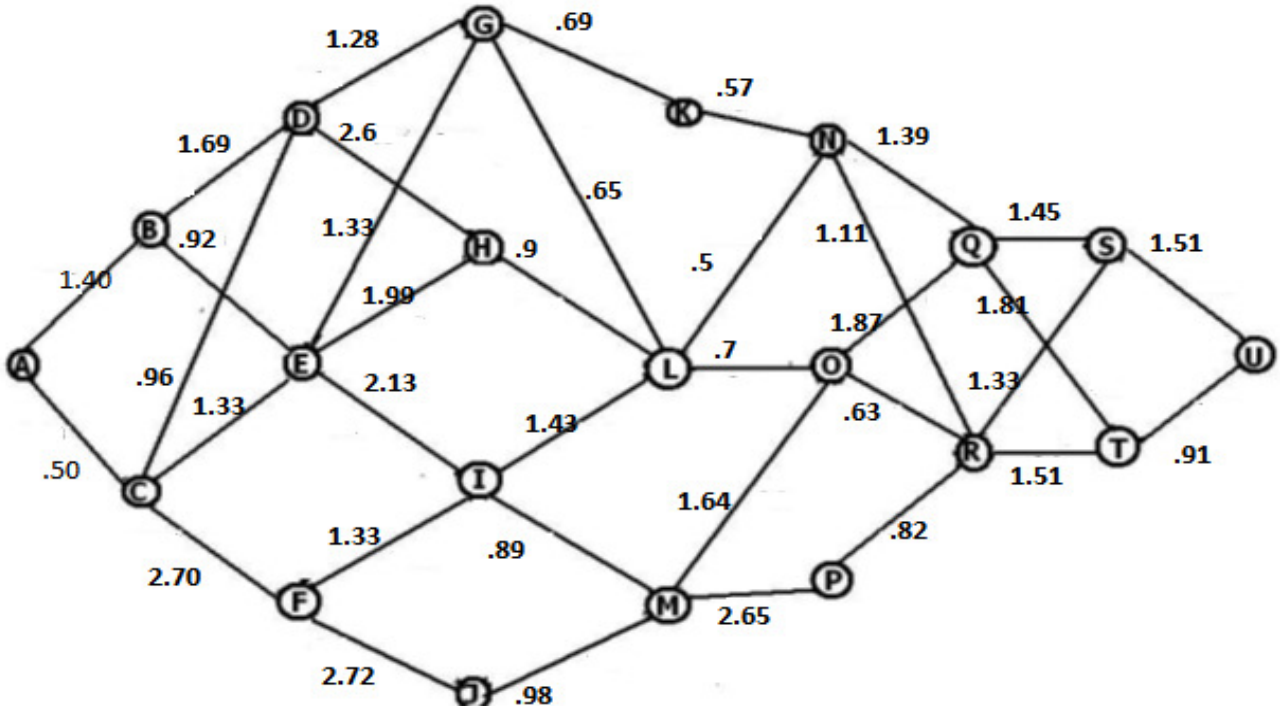

Fig: 4.2 Gives the distance of each city

The decision variables do not initially have a number assigned but would be defined at each stage by an appropriate vertex on the same vertical line as the decision variable. For example, x_3 can be represented by G, H, I, or J. Hence the decision variables can be the collection of vertices as follows:

X_0 : A
X_1 : B, C
X_2 : D, E, F
X_3 : G, H, I, J
X_4 : K, L, M
X_5 : N, O, P
X_6 : Q, R
X_7 : S, T
X_8 : U

Any route going from A to U is called forward recursion of dynamic programming. For example ACFJMPRTU is a optimal solution of forward recursion. Among the possible routes we have to find the forward shortest distance between the point A and U. We start by assigning the notation.

Stage i – Si – distance between and to the decision variables connected with each of the stages as shown in figure (4.3)

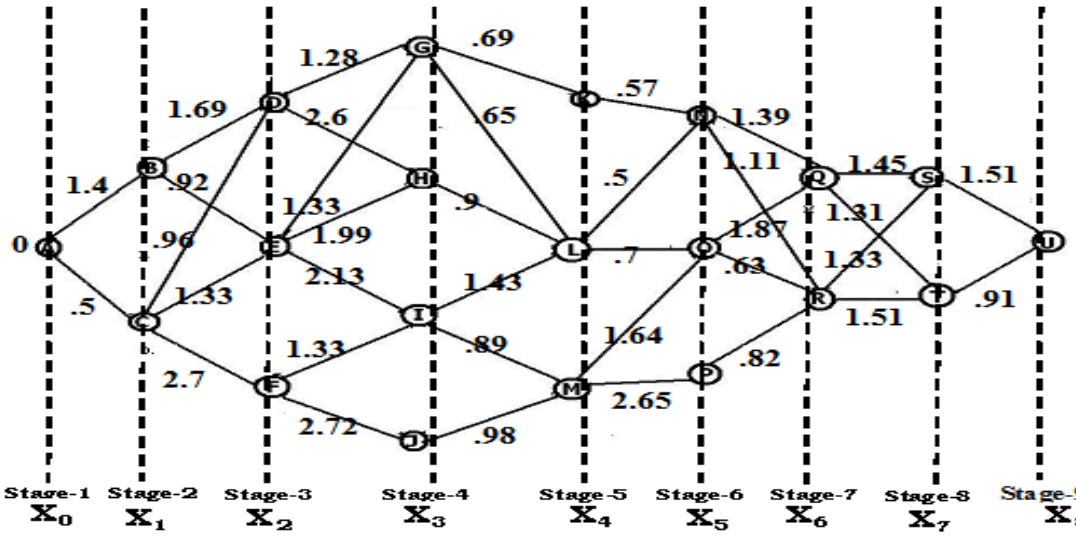

Figure 4.3 - Stage wise representation as per Algorithm

Let $S_1(X_0, X_1)$ be the distance between the point X_0 and X_1. It is apparent that the value assigned to decision variable X_1 is dependent upon the value of X_0. In this case is A for an initial value of zero. There is only one starting point in this network, namely A, but in more complex networks, there could be multiple starting points resulting in a variety of values of X_0. Similarly for $S_2(X_1, X_2)$ be the distance between the point X_1 and X_2. Hence $S_j(X_i, X_j)$ represents the distance between the point X_i and X_j respectively (i = 0, 1, 2, …,8 and j = 0, 1, 2, …,8).

The total shortest distance for this stage is $D(X_0, X_1, X_2, X_3, X_4, X_5, X_6, X_7, X_8) = S_1(X_0, X_1) + S_2(X_1, X_2) + S_3(X_2, X_3) + S_4(X_3, X_4) + S_5(X_4, X_5) + S_6(X_5, X_6) + S_7(X_6,X_7) + S_7(X_7, X_8)$.

The shortest distance for the stage 1 which terminates B or C. The choice is a simple one because the distance at X_0 or A is zero. Therefore

Stage 1 $S_1(X_0, X_1)$:

$f_1(x_1)$ / X_1	$d(X_0, X_1) + f_0(X_0)$	Optimal solution	
	$X_0 = A$	$f_1(X_1)$	X_0^*
B	1.40	1.40	A
C	0.50	0.50	A

From the stage 1 the shortest distance is 0.50 and shortest route is A → C

Stage 2 $S_2(X_1, X_2)$:

$$f_2(x_2) = \min \{d(x_1, x_2) + f_1(x_1)\},$$

X_2	$d(X_1, X_2) + f_1(X_1)$		Optimal solution	
	$X_1 = B$	$X_1 = C$	$f_2(X_2)$	X_1^*
D	1.40 + 1.69 = 3.09	0.50 + 0.96 = 1.46	1.46	C
E	1.40 + 0.92 = 2.32	0.50 + 1.33 = 1.83	1.83	C
F	-------	0.50 + 2.70 = 3.20	3.20	C

From the stage 2 the shortest distance is 1.46 and shortest route is A → C → D

Proceeding in this way we get

Stage 3: the shortest distance is 2.74 and shortest route is A→C→D→G

Stage 4: the shortest distance is 3.39 and shortest route is A→C→D→G→L

Stage 5: the shortest distance is 3.89 and shortest route is A→C→D→G→L→N

Stage 6: the shortest distance is 4.72 and shortest route is A→C→D→G→L→N→R

Stage 7: the shortest distance is 6.05 and shortest route is A→C→D→G→L→N→R→T

Stage 8: the shortest distance is 7.14 and shortest route is A→C→D→G→L→N→R→T→U

The shortest route network as shown in the figure 4.4

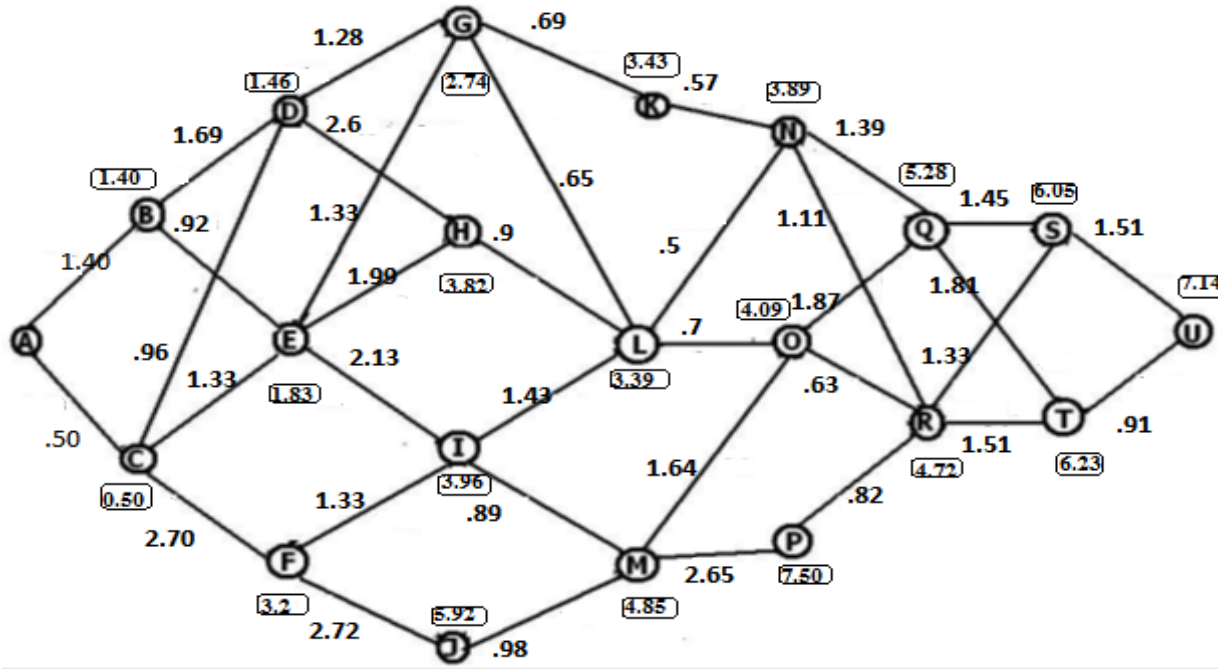

Figure 4.4 Shortest path from Chennai to Kanyakumari by Bus

Figure 4.2 - Graph from Chennai to Kanyakumari with different routes.

The shortest route from A to U are A-C-D-G-L-N-R-T-U (value = 714 km)

Hence the shortest routes among the cities are Chennai – Chengalpattu – Villupuram – Perambalur – Thanjavur – Pudukottai – Madurai – Thirunelveli – Kanyakumari.

CHENNAI TO KANYAKUMARI BY TRAIN:
Take a direct train from Chennai to Kanyakumari. Several direct express trains ply from Chennai covering the distance is approximately 14 hours. Shortest route from Chennai to Kanyakumari by train is A-C-D-G-L-N-R-T-U (Value = 708 km)

Hence the shortest routes among the cities are
Chennai – Chengalpattu – Villupuram – Ariyalur – Trichy – Dinukkal – Madurai – Thirunelveli – Kanyakumari.

V. CONCLUSION

The value of uncertain dynamic shortest route among the cities from Chennai to Kanyakumari by bus is 714 km and by train is 708 km. the recursive procedure described here was carried out from A towards U. This same procedure is often utilized in the reverse direction from U to A which leads to an equivalent solution by backward recursion. It is useful for solving several different types of network problems. Fundamentally, it consists of finding optimal solution and cumulating these through various cities and stages until optimal shortest route is found large number of algorithms is available to solve the dynamic shortest route problem.

REFERENCES

[1]. Hamdy A. Taha., Operation Research, Prentice Hall, Eighth Edition, **2011**.

[2]. Bertsekas. D., Dynamic Programming: Deterministic and Stochastic Models, Prentice Hall, Upper Saddle River, NJ, **1987.**

[3]. Denardo. E., Dynamic Programming theory and Applications, Prentice Hall, Upper Saddle River, NJ, **1982.**

[4]. Dreyfus. S and A. Law., The Art and Theory of Dynamic Programming, Academic Press, New York, **1977**.

[5]. Sntedovich. M., Dynamic Programming, Marcel Dekker, New York, **1991**.

[6]. Taha H. (**1997**) Operations Research; An introduction, Second Edition Macmillan Publishing Co. inc New York

ACKNOWLEDGMENT

Author is highly grateful to Professor Dr. Srinivasan, Department of Mathematics, SSM Engineering college , India for his valuable suggestions of this work.

AUTHOR PROFILE

First Author – karthikeyan at india. Hereceived Bacholars , Masters and Research Scholar of Scince in Mathematics from the university of Bharathidasan, India. Then he worked as a Assistant professor in Kongunadu college of engineering and technology, India from the year 2010 to till date. His research area of of intrest include Operation Research, Dynamic Programming, Arithmatic Coding Theory, Number theory, Random Processes and Queueing Theory.