

RESEARCH ARTICLE



Solving Shortest Route Using Dynamic Programming Problem

Tadios Kiros Kenea^{1*}

¹ Department Of Mathematics, Arbaminch University, Arbaminch, Ethiopia

 OPEN ACCESS

Received: 20-07-2021

Accepted: 26-05-2022

Published: 12.08.2022

Citation: Kiros Kenea T (2022) Solving Shortest Route Using Dynamic Programming Problem. Indian Journal of Science and Technology 15(31): 1527-1531. <https://doi.org/10.17485/IJST/v15i31.1342>

* **Corresponding author.**

kirostadios@gmail.com

Funding: Article Processing Charge is partially deferred by Indian Society for Education and Environment

Competing Interests: None

Copyright: © 2022 Kiros Kenea. This is an open access article distributed under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Published By Indian Society for Education and Environment ([iSee](https://www.indjst.org/))

ISSN

Print: 0974-6846

Electronic: 0974-5645

Abstract

Objective:- To determine the shortest distance in salesman of dynamic programming problem. **Method:-** Recursive equation and arrow drawing methods are used to determine the shortest distance in salesman of dynamic programming problem. An implementation of the traveling salesman problem using dynamic programming is presented in this study which generates optimal answer. **Findings:-** Forward and backward recursive methods take time and complex to find optimal solution (shortest distance) but arrow drawing method is useful and simple for solving several different types of salesman problems in dynamic programming problem compared to forward and backward recursive equation method to determine shortest route in dynamic programming problem. **Novelty:-** Dynamic programming allows to break up the given problem into a number of sub problems or stages. At each stage there are a number of decision alternatives to be called as states. So the dynamic programming uses the idea of recursive equation to solve traveling salesman problem. The Traveling salesman problem is easy to understand and difficult to solve using forward and backward recursive equation.

Keywords: Dynamic programming; Salesman problem; Operation research; Recursion equation

1 Introduction

Dynamic programming determines the optimum solution of a multivariable problem by decomposing it into stages, each stage is comprising of 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 than dealing with all the variables simultaneously. A dynamic programming 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. Analysis of biological networks is one of the central research topics in computational systems biology. A Bayesian network is a graph-based model of joint multivariate probability distributions that captures properties of conditional independence between variables. Such models are attractive for their ability to describe complex stochastic processes and because they provide a clear methodology for learning from (noisy) observations. We start by showing how Bayesian networks can describe interactions between genes. Suppose that the given set of gens g_1, g_2, \dots, g_m form a regulatory network, where each

node of the network is represented by gene. Associated each node, g_m is a random variable X_m with unknown steady-state distribution from which the expression of g_m are generated. We assume that for gene g_m , where $X_{m,n}$ is the gene expression of gene g_m under condition n . In particular, extensive studies have been done on inference of genetic networks using gene expression time series data, and a number of computational methods have been proposed, which include methods based on Boolean networks^(1,2), Bayesian networks^(3,4), time-delayed Bayesian networks⁽⁵⁾, mutual information^(6,7), and linear classification⁽⁸⁾. However, there is not yet an established or standard method for inference of genetic networks, and thus it still remains a challenging problem.

Network completion is, given an initial network and an observed dataset, to modify the network by the minimum amount of modifications so that the resulting network is (most) consistent with the observed data. Since we were interested in inference of signaling networks in our previous study⁽⁹⁾, we assumed that activity levels or quantities of one or a few kinds of proteins can only be observed. Furthermore, since measurement errors were considered to be large, and we were interested in theoretical analysis of computational complexity and sample complexity, we adopted the Boolean network⁽¹⁰⁾ as a model of signaling networks. In a paper by Halpern and Priess⁽¹¹⁾ an algorithm is presented for finding a shortest path problem with time constraints on movement and parking. It is assumed that some arcs are closed for travelling during specified periods of time and parking in the vertices of the network is permitted when it is necessary to wait for an arc to be opened. We proved that network completion is computationally intractable even for tree-structured networks. In order to cope with this computational difficulty, we developed an integer linear programming-based method for completion of signaling pathways⁽¹²⁾. However, this method could not handle addition of edges because of its high computational cost. To make survey the various methods/techniques available to solve traveling salesman problem⁽¹³⁾ and analyze arrow drawing method to make critical evaluation of their time complexities. Arrow drawing method is simple and short time taking to solve the dynamic programming problem compared to recursive equation. When we apply the forward or backward recursive equation method for the given dynamic programming problem of salesman route, we lose time and complicate to construct the table to find optimal solution.

2 Method

In this study, we survey the various methods/techniques available to solve traveling salesman problem and analyze arrow drawing method to make critical evaluation their time complexities. Recursive equation and arrow drawing methods are the methods used to determine the shortest distance in salesman of dynamic programming problem.

The purpose of network completion is to modify a given network with the minimum number of modifications so that the resulting network is most consistent with the observed data. In this paper, we consider additions and creation of sequence in states through stages as modification operations. To find the path of minimum distance between the point of source to terminal using back ward recursion or forward recursion of dynamic programming can be obtained using the following method.

Assign the initial value of the salesman as zero. i.e the value of the stage $f_0(x_0) = 0$

$f_i(x_i) = \min\{d(x_{i-1}, x_i) + f_i(x_i)\}$, $i = 1, 2, 3, \dots$ or

$f_i(x_i) = \min\{d(x_{i-1}, x_i) + f_{i-1}(x_{i-1})\}$, $i = 1, 2, 3, \dots$

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.

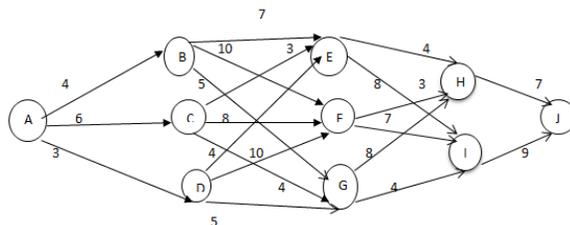
$f_i(x_i) = \min\{d(x_{i-1}, x_i) + f_{i-1}(x_{i-1})\}$, $i = 1, 2, 3, \dots$

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 dynamic programming 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 dynamic programming.

Example

A salesman located in a city A decides to travel city J. He knew the distances of alternative routes from city A to city J, then he drew a high way network map as shown in the figure below. The city of origin is A and the destination city is J. Other cities through which the salesman will have to pass through are named B to I. The arrow is representing routes between cities and distances in kilometers are indicated on each route. The salesman problem is to find shortest route from A to J. Use dynamic programming to solve it.



In the figure above the circular regions are cities and a numbers in the arrows are distance between the cities in kilometer.

The back ward recursive equation is

$$f_i(x_i) = \min\{d(x_i, x_{i+1}) + f_{i+1}(x_{i+1})\}, i=1,2,\dots,n \text{ and } d \text{ is distance between } x_i \text{ and } x_{i+1}$$

$$f_n(x_n) = 0$$

$$f_5(x_5) = 0$$

Stage :4

| x_4 | $d(x_4, x_5) + f_5(x_5)$ $x_5 = J = 0$ | Optimal solution $f_4(x_4)$ | x_5' |
|-------|---|--------------------------------|--------|
| H | 7+0 | 7 | J |
| I | 9+0 | 9 | J |

Stage :3

| x_3 | $d(x_3, x_4) + f_4(x_4)$ $x_4 = H \quad x_4 = I$ | Optimal solution $f_3(x_3)$ | x_4' |
|-------|---|--------------------------------|--------|
| E | 7+4=11 8+9=17 | 11 | H |
| F | 7+3=10 7+9=16 | 10 | H |
| G | 7+8=15 4+9=13 | 13 | I |

Stage:2

| x_2 | $d(x_2, x_3) + f_3(x_3)$ $x_3 = E \quad x_3 = F \quad x_3 = G$ | Optimal solution $f_2(x_2)$ | x_3' |
|-------|---|--------------------------------|--------|
| B | 11+7=18 10+10=20 13+5=18 | 18 | E,G |
| C | 11+3=14 10+8=18 13+4=17 | 14 | E |
| D | 11+6=17 10+10=20 13+5=18 | 17 | E |

Stage :1

| x_1 | $d(x_1, x_2) + f_2(x_2)$ $x_2 = B \quad x_2 = C \quad x_2 = D$ | Optimal solution $f_1(x_1)$ | x_2' |
|-------|---|--------------------------------|--------|
| A | 18+4=22 14+6=20 17+3=20 | 20 | C,D |

Therefore, the minimum distance is 20. i.e $f_1(x_1) = 20$

New method to solve dynamic programming problem using arrow drawing method

When using arrow drawing method we can determine the shortest and longest paths in each stages but our target is to get the shortest path. Arrow drawing method is simple and short time taking to solve the dynamic programming problem compared to recursive equation. When we apply the forward or backward recursive equation method for the given dynamic programming problem of salesman route, we lose time and complicate to construct the table to find optimal solution.

Hence A_{dk} represents Arrow drawing k sequence, S_{ij} represents each states.

$A_{dk} = S_{ij} + \dots + S_{mn}$, where $i=1,2,\dots, m$ and $j= 1,2,\dots,n$, $k=$ number of arrow drawing sequences, where i represents stage and j represents each state.

$f_i(x_i) = \min \{ \sum_{j=1}^m S_{ij} \text{ where } j=1, \dots, n \}$ is the way to find optimal solution.

$A_{d1}. S_{11} \rightarrow S_{12} \rightarrow S_{13} \dots \rightarrow \dots \dots S_{1n}$

$A_{d2}. S_{21} \rightarrow S_{22} \rightarrow S_{23} \dots \rightarrow \dots \dots S_{2n}$

$A_{d3}. S_{31} \rightarrow S_{32} \rightarrow S_{33} \dots \rightarrow \dots \dots S_{3n}$

⋮

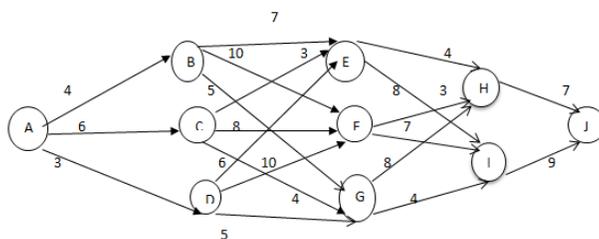
⋮

⋮

$A_{dk}. S_{m1} \rightarrow S_{m2} \rightarrow S_{m3} \dots \rightarrow \dots \dots S_{mn}$

$f_i(x_i) = \min \{ \sum_{j=1}^m S_{ij} \text{ where } j=1, \dots, n \}$ is optimal solution.

Example1. A salesman located in a city A decides to travel city J. He knew the distances of alternative routes from city A to city J, then he drew a high way network map as shown in the figure below. The city of origin is A and the destination city is J. Other cities through which the salesman will have to pass through are named B to I. The arrow is representing routes between cities and distances in kilometers are indicated on each route. The salesman problem is to find shortest route from A to J. Use dynamic programming to solve it.



Solution

Using Arrow Drawing Method

$A_{d1}. S_{11} \rightarrow S_{12} \rightarrow S_{13} \dots \rightarrow \dots \dots S_{1n}$

$A_{d2}. S_{21} \rightarrow S_{22} \rightarrow S_{23} \dots \rightarrow \dots \dots S_{2n}$

$A_{d3}. S_{31} \rightarrow S_{32} \rightarrow S_{33} \dots \rightarrow \dots \dots S_{3n}$

⋮

⋮

⋮

$A_{dm}. S_{m1} \rightarrow S_{m2} \rightarrow S_{m3} \dots \rightarrow \dots \dots S_{mn}$

$f_i(x_i) = \min \{ \sum_{j=1}^m S_{ij} \text{ where } j=1, \dots, n \}$ is optimal solution.

$A_{d1}. 4+7+4+7=22$

$A_{d2}. 4+7+8+9=28$

$A_{d3}. 4+10+3+7=24$

$A_{d4}. 4+10+7+9=30$

$A_{d5}. 4+5+8+7=24$

$A_{d6}. 4+5+4+9=22$

$A_{d7}. 6+3+4+7=20$

$A_{d8}. 6+3+8+9=26$

$A_{d9}. 6+8+3+7=24$

$A_{d10}. 6+8+7+9=30$

$A_{d11}. 6+4+8+7=25$

$A_{d12}. 6+4+4+9=23$

$A_{d13}. 3+6+4+7=20$

$$A_{d14}.3+6+8+9=26$$

$$A_{d15}.3+10+3+7=23$$

$$A_{d16}.3+10+7+9=29$$

$$A_{d17}.3+5+8+7=23$$

$$A_{d18}.3+5+4+9=21$$

$f_i(x_i) = \min \{ \sum_{j=1}^m S_{ij} \text{ where } j=1, \dots, n \}$ is 20 optimal solution.

There fore $f_i(x_i) = 20$ is optimal solution, the sequence

$A \xrightarrow{6} C \xrightarrow{3} E \xrightarrow{4} H \xrightarrow{7} J = 20$ and $A \xrightarrow{3} D \xrightarrow{6} E \xrightarrow{4} H \xrightarrow{7} J = 20$ is shortest route.

3 Discussion and results

The value of certain dynamic shortest route among the cities from A to J is 20. The recursive procedure described four different tables which is taken at least 23 lines, time taking and complex to achieve solution but as we have seen the modified method cover 19 lines, time saving and simpler even if we compared in line coverage, time and the way of solving so here was carried out from A to J. Those the same procedure is often utilized in the reverse direction from J to A which leads to an equivalent solution by backward recursion. Hence using arrow drawing method to computing the shortest and longest path between two given location in salesman problem is an important and simplest way that to find applications in various map services. An implementation of the traveling salesman problem using dynamic programming is presented in this paper which generates optimal answer. So these forward and backward recursive methods take time and complex to find optimal solution but **arrow drawing method** is useful and simple for solving several different types of salesman problems in dynamic programming problem compared to forward and backward recursive methods. 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 available to solve the dynamic shortest route problem.

4 Conclusion

To solve the traveling salesman route we apply the recursive equation (either forward or backward recursive). It is taking long time and very complicate if a number of states are many to get the shortest distance of salesman problem in dynamic programming problem. So that it is better to apply arrow draw method to determine the shortest distance and to take short time for solving the given problem compared to recursive equation. We can see not only the shortest distance of route but also able to identify the longest distance that helps us not to use this longest one to minimize the cost.

References

- 1) Liang S, Fuhrman S, Somogyi R. REVEAL, a general reverse engineering algorithm for inference of genetic network architectures. *Proceedings of the Pacific Symposium on Biocomputing*. 1998;3:18–29.
- 2) Akutsu T, Miyano S, Kuhara S. Inferring qualitative relations in genetic networks and metabolic pathways. *Bioinformatics*. 2000;16(8):727–734.
- 3) Friedman N, Linial M, Nachman I, Pe'er D. Using Bayesian networks to analyze expression data. *Proceedings of the fourth annual international conference on Computational molecular biology - RECOMB '00*. 2000;7:601–620.
- 4) Imoto S, Sunyong K, Goto T, Aburatani S, Tashiro K, Kuhara S, et al. Bayesian network and nonparametric heteroscedastic regression for nonlinear modeling of genetic network. *Proceedings IEEE Computer Society Bioinformatics Conference*. 2003;1:231–252.
- 5) Liu TFF, Sung WKK, Mittal A. Learning gene network using time-delayed Bayesian network. *International Journal on Artificial Intelligence Tools*. 2006;15(03):353–370.
- 6) Margolin AA, Wang K, Lim WK, Kustagi M, Nemenman I, Califano A. Reverse engineering cellular networks. *Nature Protocols*. 2006;1(2):662–671.
- 7) Margolin AA, Nemenman I, Basso K, Wiggins C, Stolovitzky G, Favera RD, et al. ARACNE: An Algorithm for the Reconstruction of Gene Regulatory Networks in a Mammalian Cellular Context. *BMC Bioinformatics*. 2006;7(S1).
- 8) Kimura S, Nakayama S, Hatakeyama M. Genetic network inference as a series of discrimination tasks. *Bioinformatics*. 2009;25(7):918–925.
- 9) Akutsu T, Tamura T, Horimoto K. Completing Networks Using Observed Data. *Lecture Notes in Computer Science*. 2009;5809:126–140.
- 10) Ffman SAK. *The Origins of Order: Self-Organization and Selection in Evolution*. New York, NY, USA. Oxford University Press. 1993.
- 11) Halpern IJ, Priess. *Shortest path with time constraints on movement and parking*. 1974;4:241–253.
- 12) Tamura T, Yamanishi Y, Tanabe M, Goto S, Kanehisa M, Horimoto K, et al. Integer programming-based method for completing signaling pathways and its application to analysis of colorectal cancer. *Genome Informatics 2010*. 2010;24:193–203.
- 13) Lewis R. Algorithms for Finding Shortest Paths in Networks with Vertex Transfer Penalties. *Algorithms*. 2020;13(11):269–269.