ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/300038752

TRAVELLING SALESMAN PROBLEMS BY DYNAMIC PROGRAMMING ALGORITHM

Article · February 2016

CITATIONS
2

READS
14,249

2 authors, including:

Abha Singhal
Poornima University
3 PUBLICATIONS **2** CITATIONS

SEE PROFILE

All content following this page was uploaded by Abha Singhal on 09 April 2016.

The user has requested enhancement of the downloaded file.

International Journal of Scientific Engineering and Applied Science (IJSEAS) – Volume-2, Issue-1, January 2016
ISSN: 2395-3470
www.ijseas.com

TRAVELLING SALESMAN PROBLEMS BY DYNAMIC PROGRAMMING ALGORITHM

Dr. Abha Singhal[*] and Priyanka Pandey[**]
Department of Mathematics,
Poornima Group of Institution, Jaipur (Raj.)-, India

Abstract

In the present paper, I used Dynamic Programming Algorithm for solving Travelling Salesman Problems with Matrix. To illustrate the proposed Algorithm, a travelling salesman problem is solved. To make clear, algorithm of the proposed method is also given. The proposed method is easy to understand and apply to find optimal solution of travelling salesman problems occurring in real life situations.

Key Words: Travelling Salesman problem, Dynamic Programming Algorithm, Matrix.

1. Introduction

In the traveling salesman problem, a map of cities is given to the salesman and he has to visit all the cities only once and return to his starting point to complete the tour in such a way that the length of the tour is the shortest among all possible tours for this map. Clearly starting from a given city, the salesman will have a total of (n-1)! Different sequences. If n = 2, A and B, there is no choice. If n = 3, i.e. he wants to visit three cities inclusive of the starting point, he has 2! Possible routes and so on.

Dynamic programming (usually referred to as **DP**) is a very powerful technique to solve a particular class of problems. It demands very elegant formulation of the approach and simple thinking and the coding part is very easy. The idea is very simple, If you have solved a problem with the given input, then save the result for future reference, so as to avoid solving the same problem again. If the given problem can be broken up in to smaller sub-problems and these smaller subproblems are in turn divided in to still-smaller ones, and in this process, if you observe some over-lappping subproblems, then its a big hint for DP. Also, the optimal solutions to the subproblems contribute to the optimal solution of the given problem.

There are two ways of doing this.

1.) Top-Down : Start solving the given problem by breaking it down. If you see that the problem has been solved already, then just return the saved answer. If it has not been solved, solve it and save the answer. This is usually easy to think of and very intuitive. This is referred to as ***Memoization***.

2.) Bottom-Up : Analyze the problem and see the order in which the sub-problems are solved and start solving from the trivial subproblem, up towards the given problem. In this process, it is guaranteed that the subproblems are solved before solving the problem. This is referred to as ***Dynamic Programming***.

In this paper, we use the dynamic programming algorithm for finding a optimal solution for a travelling salesman problem where all parameters are in matrix form. The solution procedure is illustrated with the numerical example.

When we use the dynamic programming algorith for finding an optimal solution for a travelling salesman problem, we have the following advantages.

- We don't use linear programming techniques.
- We don't use goal and parametric programming techniques.
- The proposed method is very easy to understand and apply.

2. Travelling Salesman Problem

Consider the following fuzzy travelling salesman problem,

Min. $\tilde{z} = \sum_{j=1}^{n} \tilde{c}_{ij}\tilde{x}_{ij}$; [$i$= 1, 2, 3, .., n; j= 1, 2, 3, …, n]

Subject to the constraints:

(i) $\sum_{i=1}^{n} \tilde{x}_{ij} = 1$; j = 1, 2, …, n

(ii) $\sum_{j=1}^{n} \tilde{x}_{ij} = 1$; i= 1, 2, …, n.

Where $\tilde{x}_{ij} = \begin{cases} 1; \textit{if the salesman travel from city } i \textit{ to city } j \\ 0; \textit{otherwise} \end{cases}$

$\tilde{c}_{ij}$ = Distance (or cost or time) of going from city i to city j.

$\tilde{z}$ = Min. The total cost of the matrix.

3 Algorithm

The Dynamic Programming proceeds as follows:-

Step-1 Consider the given travelling salesman problem in which he wants to find that route which has shortest distance.

Step-2 Consider set of 0element, such that

$g(2, \Phi) = c_{21}$
$g(3, \Phi) = c_{31}$
$g(4, \Phi) = c_{41}$

Step-3 After completion of step-2, consider sets of 1 elements, such that

Set {2}: $g(3,\{2\}) = c_{32} + g(2, \Phi) = c_{32} + c_{21}$
$g(4,\{2\}) = c_{42} + g(2, \Phi) = c_{42} + c_{21}$

Set {3}: $g(2,\{3\}) = c_{23} + g(3, \Phi) = c_{23} + c_{31}$
$g(4,\{3\}) = c_{43} + g(3, \Phi) = c_{43} + c_{31}$

Set {4}: $g(2,\{4\}) = c_{24} + g(4, \Phi) = c_{24} + c_{41}$
$g(3,\{4\}) = c_{34} + g(4, \Phi) = c_{34} + c_{41}$

Step-4 After completion of step-3, consider sets of 2 elements, such that
Set {2,3}: $g(4,\{2,3\}) = \min\{c_{42} + g(2,\{3\}), c_{43} + g(3,\{2\})\}$

Set {2,4}: $g(3,\{2,4\}) = \min\{c_{32} + g(2,\{4\}), c_{34} + g(4,\{2\})\}$

Set {3,4}: $g(2,\{3,4\}) = \min\{c_{23} + g(3,\{4\}), c_{24} + g(4,\{3\})\}$

Step-5 After completion of step-4, Find the length of an optimal tour:

$$f = g(1,\{2,3,4\}) = \min\{ c_{12} + g(2,\{3,4\}), c_{13} + g(3,\{2,4\}), c_{14} + g(4,\{2,3\}) \}$$

Step-6 After completion of step-5, Find the Optimal TSP tour.

We can understand the praposed algorithm from the following numerical example.

4. Numerical example

The proposed method is illustrated by the following examples.

Example 1. A salesman must visit from city to city to maintain his accounts. He has to leave his home city A and visit other cities once and return home. The cost of going from city to city is shown in the table. Find the least cost route

Distance matrix

$$\begin{pmatrix} 0 & 2 & 9 & 10 \\ 1 & 0 & 6 & 4 \\ 15 & 7 & 0 & 8 \\ 6 & 3 & 12 & 0 \end{pmatrix}$$

Solution:

$g(2, \Phi) =$
$c_{21} = 1$
$g(3, \Phi) = c_{31}$
$= 15$
$g(4, \Phi) =$
$c_{41} = 6$

k = 1, consider sets of 1 element:

Set {2}: $g(3,\{2\}) = c_{32} + g(2, \Phi) = c_{32} + c_{21} = 7 + 1 = 8$
$g(4,\{2\}) = c_{42} + g(2, \Phi) = c_{42} + c_{21} = 3 + 1 = 4$

Set {3}: $g(2,\{3\}) = c_{23} + g(3, \Phi) = c_{23} + c_{31} = 6 + 15 = 21$
$g(4,\{3\}) = c_{43} + g(3, \Phi) = c_{43} + c_{31} = 12 + 15 = 27$

Set {4}: $g(2,\{4\}) = c_{24} + g(4, \Phi) = c_{24} + c_{41} = 4 + 6 = 10$
$g(3,\{4\}) = c_{34} + g(4, \Phi) = c_{34} + c_{41} = 8 + 6 = 14$

k = 2, consider sets of 2 elements:

Set {2,3}: $g(4,\{2,3\}) = \min \{c_{42} + g(2,\{3\}), c_{43} + g(3,\{2\})\} = \min \{3+21, 12+8\} = \min \{24, 20\} = 20$

Set {2,4}: $g(3,\{2,4\}) = \min \{c_{32} + g(2,\{4\}), c_{34} + g(4,\{2\})\} = \min \{7+10, 8+4\} = \min \{17, 12\} = 12$

Set {3,4}: $g(2,\{3,4\}) = \min \{c_{23} + g(3,\{4\}), c_{24} + g(4,\{3\})\} = \min \{6+14, 4+27\} = \min \{20, 31\} = 20$

Length of an optimal tour:

$$f = g(1,\{2,3,4\}) = \min \{ c_{12} + g(2,\{3,4\}), c_{13} + g(3,\{2,4\}), c_{14} + g(4,\{2,3\}) \}$$
$$= \min \{2 + 20, 9 + 12, 10 + 20\} = \min \{22, 21, 30\} = 21$$

Successor of node 1: $g(2, \Phi) = c_{21} = 1$

Successor of node 2: $g(4,\{2\}) = 4$

Successor of node 4: $g(3,\{2,4\}) = 12$

Successor of node 3: $g(1,\{2,3,4\}) = 21$

Optimal TSP tour: $1 \rightarrow 2 \rightarrow 4 \rightarrow 3 \rightarrow 1$

5. Conclusion

We have attempted to develop a new method to find a root to travelling salesman problem. The proposed method for an optimal tour is very simple, easy to understand and apply. From this method salesman can visit more cities at a time with minimum cost.

Reference

[1] Buckly, J. J., Possibilistic linear programming with triangular fuzzy numbers, fuzzy sets and system, 26, 135-138, 1998.

[2] Chen S. H., Operation on fuzzy numbers with function principle, Tamkang Journal of Management Sciences, 6, 13-25, 1985.

[3] Kaufmann A., Introduction to the theory of fuzzy sets, 1, Academic Press, New York, 1976.

[4] Pandian P. and Natarajan G., Anew algorithm for finding a fuzzy optimal solution for fuzzy transportation problems, Applied mathematical sciences, 4, 79-90, 2010.

[5] Rommelfanger H., Wolf J. and Hanuschek R., Linear programming with fuzzy objectives, Fuzzy Sets and System, 29, 195-206, 1989.

[6] Sharma J. K., Operation research theory and application, Third Edition, 2007.

[7] Zadeh L.A., Fuzzy sets Information and Control, 8, 3, 338-353, 1965.

[8] Use the link http://www.mafy.lut.fi/study/DiscreteOpt/tspdp.pdf

View publication stats