Proceedings of

The 1st University Journal on

Research and Applications 2019

Contents

Web Page Categorization Using Frequency Ratio Accumulation Method 1-4
Thin Nu Nu Win, Ngu War Lwin, Kaythi Oo

Task Ontology for Tourist Information System 5-9
Lai Lai Win, Zar Zar Hnin, Khin Myat Nwe Win

University Ontology Development and Query Retrieval Using SPARQL 10-15
Zarni May Myat Thu, Kaythi Oo, Ngu War Lwin

The Model for Turning into Triumphant University 16-20
Yuzana, Tin Htar Nwe, Khin Myo Aye, Nwet Yin Tun Thein,
Nang Win Phyu Phyu Naing

Finding Shortest Path Using Heuristic Search Technique 21-26
Win Cherry Ko, Htwe Htwe Linn

Travelling Salesman Problem by Using Genetic Algorithm 27-30
Htwe Htwe Lin, Kyi Kyi Lwin

Disambiguation of Postposition for Myanmar Word Segmentation 31-36
Kaythi Oo, Zar Ni May Myat Thu, Thin Nu Nu Win

A Review on Sentiment Analysis Research in Myanmar Language 37-41
Win Win Thant, Ei Ei Mon, Sandar Khaing

Comparative Study of LAD, Naïve Bayes and Random Forest Classification Techniques for Student Performance Prediction 42-46
Phyo Yadana Soe, Aye Myat Thu

Theoretical Observation on Ionizing Radiation 153-157

Tin Tin Pyone, Yoon Mone Phoo

Applications of Weighted Least-Squares Approximations 158-164

Khin Kyawt Kyawt San

Minimum Spanning Tree of Road Network in Myanmar by using Dijkstra's Algorithm 165-168

Mar Lar Aung, Phyu Sin Win, Than Shwe

Application of Shortest Path 169-173

Aye Aye Myint, Myat Moe Khaing, Wai Zin Phyo

Developing Students' Language Ability through Oral Presentation Task 174-177

Ei Ei Khaing, Ei Ei Htay

Investigation into EFL Teachers' Perceptions of Motivational Strategies at University of Computer Studies (Monywa) 178-183

Nyein Thu Thu Swe

A Study of EFL Students' Reading Comprehension Skills in Language Teaching: Problems and Strategies 184-188

Ngu War War Htun

ညီသစ်ဆင်းကဗျာများတွင် တွေ့ရသော ကဗျာစာဆိုတော်ဇော်ဂျီ ၏ လမ်းညွှန်ချက်များ 189-195

Ni Lar Tint

ခင်ခင်ထူး၏ လူတန်းစားဘဝသရုပ်ဖော် ဝတ္ထုတိုများမှ ဇာတ်ဆောင်စရိုက်အဖွဲ့များ 196-202

Phyo Thiri Khaing

အင်းဝခေတ် မိဘနှင့်သားသမီးကျင့်ဝတ် ဆုံးမစာများ 203-208

Cho Cho Aye

The Annual University Journal on Research and Applications, 2019, Vol-1, Issue 1

Application of Shortest Path

Aye Aye Myint
Myanmar Institute of Information Technology
aye_aye_myint@miit.edu.mm

Myat Moe Khaing
Myanmar Institute of Information Technology
myat_moe_khaing@miit.edu.mm

Wai Zin Phyo
Myanmar Institute of Information Technology
wai_zin_phyo@miit.edu.mm

Abstract_ In this paper, a shortest path algorithm was studied. Firstly, the basic definitions of graph theory such as graph, directed graph (or digraph), weighted graph, path and shortest path will be introduced. Then the shortest path algorithm of Dijkstra into the MATLAB computer program was coded and applied it to the real road network of some towns in the Sagaing division so that there are two places in the region. This program tests the road-network which includes 16 junctions and 28 road-segments. It is hoped that the program would be applied on road-networks of a city or of larger sizes.

Keywords: graph, digraph, weighted graph, path, shortest path.

I. INTRODUCTION

In the city with road-networks, the shortest route between every two locations is useful and important from economic and other points of view. Using a shortest path from one location to the other, a person or an organization may save his or its time, cost and other resources.

A weighted graph is a graph in which edge values are allocated and a path length in a weighted graph is the sum of edge weights in the route. A path with a minimum distance between two specified vertices is called a shortest path in the weighted graph. The shortest path problems are the most significant issues of actual world optimization because they can be formulated and solved as shortest path problems.

II. GRAPH THEORY DEFINITIONS

A. Graph

A **graph** G = (V, E) is a non-empty set of vertices (or nodes) and a set of edges (or arcs). An unordered pair of vertices is connected with each edge. If edge e is connected to vertices v and w, an edge e is connected (v, w) or (w, v). A graph like this is shown in Figure 1.

B. Digraph

A **digraph** D = (V, E) comprises of a non-empty set V of vertices (or nodes) and a set E of separated edges (or arcs). An ordered pair of vertices is connected with each edge. If it connects to the ordered pair (v, w) of vertices, the edge e is denoted (v, w). In the Figure 2, e_1=(v, u), e_2=(w, u), e_3=(z, w), e_4=(v, v), e_5=(w, v), e_6=(v, w). D is a simple digraph if all of D's arcs are separate and there are no loops.

Fig. 1. Graph with V = {u, v, w, z} and E = {e_1, e_2, e_3, e_4, e_5, e_6, e_7}

Fig. 2. Digraph with V = {u, v, w, z} and E = {e_1, e_2, e_3, e_4, e_5, e_6, e_7}

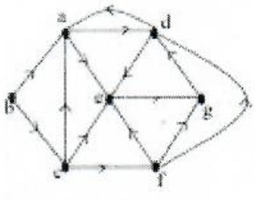


Fig. 3. A Simple digraph

C. Weighted Graph

A real number, called its **weight**, can be connected with each arc of a digraph D. Then D is called a **weighted digraph** along with these weights on its arcs. In most applications weights are positive values.

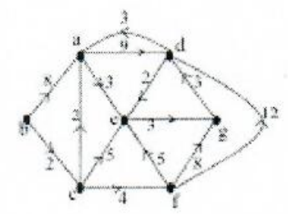


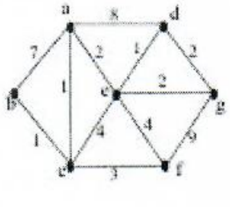
Fig. 4. A weighted graph with weights

D. Path and Shortest Path

If a person goes on a journey that starts in a certain town, runs through several towns and ends in a certain town, a person must know a path and a shortest path.

An alternating sequence of n+1 vertices and n edges starting with vertex v_0 and ending with vertex v_n is a **path** between two designated vertices v_0 and v_n with length n, denoted by $(v_0, e_1, v_1, e_2, v_2, ..., v_{n-1}, e_n, v_n)$ where e_i connects with v_{i-1} and v_i for $i = 1, 2, ..., n$.

A path with a minimum distance between two specified vertices is called a **shortest path** in the weighted graph.



Path	Length
(b,a,d,g)	17
(b,c,e,f,g)	18
(b,c,f,g)	13
(b,a,e,g)	11
(b,a,e,d,g)	12
(b,c,a,e,g)	6

Fig. 5. Weighted graph with path

III. Dijkstra's Algorithm

Dijkstra's algorithm finds the length of a shortest path and shortest distance from any vertex to other vertex in a connected and weighted graph.

E. *Procedure*

Suppose G is a connected weighted graph with vertex set $V = \{v_1, . . . , v_n\}$ in which all weights are positive and let $L(v_i)$ be the label of the vertex v_i and the weight of edge connects with v_i and v_j be greater than zero.

Dijkstra's algorithm finds the shortest distance and shortest path by adding vertices $v_1, . . . , v_n$ into the this path as follow:

Starts with the weighted n×n matrix of a simple weighted digraph or weighted graph, $M=(a_{ij})$.

where a_{ij} = the weight of the edge from vertex v_i to vertex v_j,

$a_{ii} = 0$,

$a_{ij} = \infty$, if there is no edge from vertex v_i to vertex v_j,

and n is the number of vertices.

Set $L(v_1) = 0$ and $L(v_k) = \infty$ for all vertices $v_k \neq v_1$, $k = 2, . . . , n$.

Then choose vertex v_2 adjacent to v_1 with minimum $L(v_2)$ by using the equation

$$L(v_2) = \min\{L(v_2), L(v_1) + \text{weight}(v_1, v_2)\}.$$

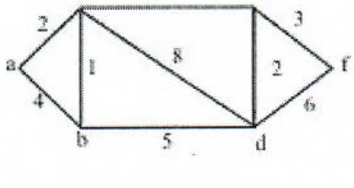
That is, in step 1, vertex v_1 is inserted in the path from vertex v_1 to vertex v_n.

In step 2, vertex v_2 is inserted in this path.

Continuing in this way, at termination, $L(v_n)$ is the length of a shortest path from vertex v_1 to vertex v_n.

F. *Example*

Consider G is a weighted graph:



Let $L(f)$ is the shortest distance from vertex a to vertex f.

TABLE I. Path Length Table

Path	L(b)	L(c)	L(d)	L(e)	L(f)
(a)	4	***2***	∞	∞	∞
(a, c)	***3***	-	10	12	∞
(a, c, b)	-	-	***8***	12	∞
(a, c, b, d)	-	-	-	***10***	14
(a, c, b, d, e)	-	-	-	-	***13***
(a, c, b, d, e, f)	-	-	-	-	***13***

From the above table, the shortest distance is 13 and shortest path is (a, c, b, d, e, f).

IV. Coding Dijkstra's Algorithm

There are several computer programs in the graph that find a shortest distance and a shortest path between two vertices.

The shortest path algorithm of Dijkstra into the MATLAB computer program was coded as follow:

```
function [d P] = dijkstralgorithm(G,start,destin)
if start==destin
  d =0;
  P=[start];
else
G = drawgraph(G,inf,1);
if destin==1
  destin=start;
end
G=RowColchange(G,1,start);
lengthA=size(G,1);
Width=zeros(lengthA);
for i=2 : lengthA
  Width(1,i)=i;
  Width(2,i)=G(1,i);
end
for i=1 : lengthA
  Dis(i,1)=G(1,i);
  Dis(i,2)=i;
end
D2=Dis(2:length(Dis),:);
P=2;
while P<=(size(Width,1)-1)
  P=P+1;
  D2=sortrows(D2,1);
  k=D2(1,2);
  Width(P,1)=k;
  D2(1,:)=[];
  for i=1 : size(D2,1)
    if Dis(D2(i,2),1)>(Dis(k,1)+G(k,D2(i,2)))
      Dis(D2(i,2),1) = Dis(k,1)+G(k,D2(i,2));
      D2(i,1) = Dis(D2(i,2),1);
    end
  end
  for i=2 : length(G)
    Width(P,i)=Dis(i,1);
  end
end
if destin==start
  P=[1];
else
  P=[destin];
end
e=Width(size(Width,1),destin);
```

```
P = listalgorithm(P,Width,start,destin);
end
end
function G = RowColchange(G,a,b)
%Exchange element at column;
buffer=G(:,a);
G(:,a)=G(:,b);
G(:,b)=buffer;
%Exchange element at row;
buffer=G(a,:);
G(a,:)=G(b,:);
G(b,:)=buffer;
end
function L = listalgorithm(L,W,s,d)
index=size(W,1);
while index>0
  if W(2,d)==W(size(W,1),d)
     L=[L s];
     index=0;
  else
     index2=size(W,1);
     while index2>0
        if W(index2,d)<W(index2-1,d)
           L=[L W(index2,1)];
           L=listalgorithm(L,W,s,W(index2,1));
           index2=0;
        else
           index2=index2-1;
        end
        index=0;
     end
  end
end
end
function G = drawgraph(G,b,s)
if s==1
  for i=1 : size(G,1)
     for j=1 :size(G,1)
        if G(i,j)==0
           G(i,j)=b;
        end
     end
  end
end
if s==2
  for i=1 : size(G,1)
     for j=1 : size(G,1)
        if G(i,j)==b
           G(i,j)=0;
        end
     end
  end
end
end
```

G. Example

Suppose G is a weighted graph in below.

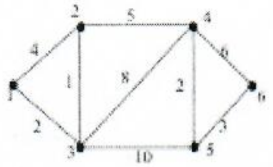


Fig. 6. Weighted graph with weights

Input the data:

```
G=[0 4 2 0 0 0;  4 0 1 5 0 0; 2 1 0 8 10 0; 0 5 8 0 2 6;
   0 0 10 2 0 3;  0 0 0 6 3 0; ];
[d P] = dijkstra (G,1,6)
The result is
d = 13
P = 1 3 2 4 5 6
```

That is, the shortest distance and shortest path from vertex 1 to vertex 6 are 13 and (1, 3, 2, 4, 5, 6) respectively.

V. Some Results And Findings

In this paper, the program with partial road-networks of some towns in Sagaing division was tested. Such a partial road-network with 16 towns and 28 road-segments is shown in Fig.7. Here the road-lengths are in kilometers. The list representation for this partial road-network is shown in Table II. (See in Page. 5)

Let Ayadaw = vertex 1
Budalin = vetex 2
Tabayin = vertex 3
Ye-U = vertex 4
Shwebo = vertex 5
Myinmu = vertex 6
Chaung-U= vertex 7
Monywa = vertex 8
Kani = vertex 9
Khin-U = vertex 10
Wetlet = vertex 11
Sagaing = vertex 12
Myaung = vertex 13
Salingyi = vertex 14
Pale = vertex 15
Yinmarpin = vertex 16.

Input data:

```
s = [1 1 1 1 1 1 1 2 2 2 3 4 4 5 5 11 11 12 6 6 7 7 8 8
     16 16 16 14];
t = [2 3 4 5 6 7 8 3 8 9 4 5 10 10 11 6 12 6 7 13 8 13
     14 16 9 14 15 15];
weights = [51 70 84 60 46 65 43 38 32 33 14 37 21 35
           37 88 74 60 37 27 22 33 19 47 32 36 19 35];
```

By using the technical tool in Matlab, the result plot is the partial road-network with 16 towns and 28 road-segments shown in Figure 7.

The result plot:

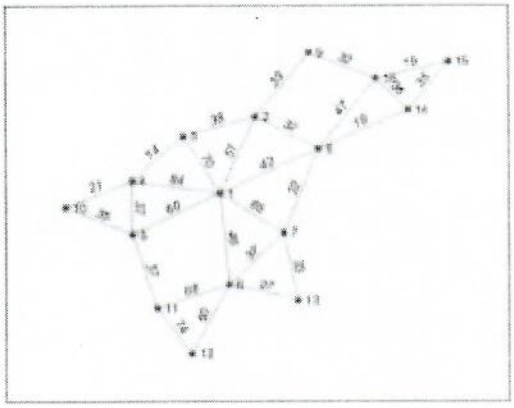


Fig. 7. A partial road-network of Sagaing division

The shortest distance and shortest path between two specified towns will be calculated by using Matlab computer program.

Input the data:

```
G = [0 51 70 84 60 46 65 43 0 0 0 0 0 0 0 0;
    51 0 38 0 0 0 0 32 33 0 0 0 0 0 0 0;
    70 38 0 14 0 0 0 0 0 0 0 0 0 0 0 0;
    84 0 14 0 37 0 0 0 0 21 0 0 0 0 0 0;
    60 0 0 37 0 0 0 0 0 35 37 0 0 0 0 0;
    46 0 0 0 0 0 37 0 0 0 88 60 27 0 0 0;
    65 0 0 0 0 37 0 22 0 0 0 0 33 0 0 0;
    43 32 0 0 0 0 22 0 0 0 0 0 0 19 0 47;
    0 33 0 0 0 0 0 0 0 0 0 0 0 0 0 32;
    0 0 0 21 35 0 0 0 0 0 0 0 0 0 0 0;
    0 0 0 0 37 88 0 0 0 0 0 74 0 0 0 0;
    0 0 0 0 0 60 0 0 0 0 74 0 0 0 0 0;
    0 0 0 0 0 27 33 0 0 0 0 0 0 0 0 0;
    0 0 0 0 0 0 0 19 0 0 0 0 0 0 35 36;
    0 0 0 0 0 0 0 0 0 0 0 0 0 35 0 19;
    0 0 0 0 0 0 0 47 32 0 0 0 0 36 19 0;];
[d P] = dijkstra (G,12,8)
The result is
d = 119
P = 8 7 6 12
```

That is, the results indicate that a shortest path from Monywa to Sagaing is (Monywa, Chaung-U, Myinmu, Sagaing) and a shortest distance is 119km.

VI. Conclusion

There are several government and private organisations in a town whose cars must operate one or more times a day between two designated town places. The organisations can considerably save their costs, time and other resources by using the shortest paths and can also enhance the organizations ' productivity and effectiveness. In addition, some graph theory applications are helpful in discrete mathematics, computer science, and other applications.

References

[1] J.A. Bondy and U.S.R. Murty, "Graph Theory with Applications", The Macmillan Press Ltd, London, 1976.

[2] H.P. Yap, "Some Topics in Graph Theory", Cambridge University Press, 1986.

[3] K.R. Parthasarathy, "Basic Graph Theory", Tata McGraw-Hill Publishing Company Limited, New Delhi, 1994.

[4] Kenneth H. Rosen, *"Discrete Mathematics and Its Applications"*, Seventh Edition McGraw-Hill Pub. Co. Ltd., New York, (2011).

[5] R. Johnsonbaught, *"Discrete Mathematics"*, Seventh Edition Pearson Education Limited UK, (2013).

[6] Yi Fan Hou, Zhi Wu Li, Mi Zhao, Ding Liu, "Extension of Elementary Sophons in a class of Generalized Petri Nets", Applied Mechanics abd Materials, Trans Tech Publication, Switzerland, 2013.

[7] Vemani, "Graphs", A Course in Discrete Mathematical Structures, 2012

[8] Sage, http://sagemath.org., Free, public notebook server at http://sagenb.org

[9] "Mathlab Numerical Computing", Tutorials Point (I) Pvt.Ltd, 2014.

[10] *"The Language of Technical Computing"*, MATLAB R2018b.

Table II. List Representation Of Partial Road-network Of Figure

Vertex	adjacent vertices	arc length	adjacent vertex	arc length	adjacent vertex	arc length	adjacent vertex	arc length	adjacent vertex	arc length	adjacent vertex	arc length	adjacent vertex	arc length	adjacent vertex
1	7	51	2	70	3	84	4	60	5	46	6	65	7	43	8
2	3	51	1	38	3	33	9								
3	3	70	1	38	2	14	4								
4	4	84	1	14	3	37	5	21	10						
5	4	60	1	37	4	35	10	37	11						
6	5	46	1	37	7	88	11	60	12	27	13				
7	4	65	1	37	6	22	8	33	13						
8	5	43	1	32	2	22	7	19	14	47	16				
9	2	33	2	32	16										
10	2	21	4	35	5										
11	3	37	5	88	6	74	12								
12	2	60	6	74	11										
13	2	27	6	33	7										
14	3	19	8	35	15	36	16								
15	2	35	14	19	16										
16	4	47	8	32	9	36	14	19	15						

Table II. List Representation Of Partial Road-network Of Figure