The Determination of a Graph Center

Cătălin Angelo Ioan[1] Gina Ioan[2]

Abstract: The article presents a method of determining the center of a graph based on Bellman-Kalaba's algorithm. There is also a software to determine it in Mathematica 9.0.

Keywords: graph; center; algorithm

JEL Classification: E17; E27

1. Introduction

The problem of determining the center of a graph (the node for which the sum of the distancies to the other nodes in minimal) comes as a complement to localization theory.

The problem lies in determining the point (points) for which the sum of distances at fixed points is minimal. If for arbitrary points in plan, the problem is very difficult (for three points being solved in the mid-century XVII by Pierre de Fermat in a letter to Evangelista Torricelli), for arbitrary number of points it is very difficult. In a previous paper (Ioan & Ioan, 2014, pp. 141-148), we tried to give an algorithm for this question.

In the case of a graph the problem is more simple. Let the fixed nodes A_k, k=$\overline{1,n}$, n≥2 and A_s – the target node. The Bellman-Kalaba algorithm ([1]) states that we build for each node A_k, the effective distances matrix $D=(d_{ij})$, $d_{ij}=d(A_i,A_j)$, $i,j=\overline{1,n}$ where $d(A_i,A_j)$ is the length arc connecting A_i to A_j if exists, $d(A_i,A_j)=\infty$ if between A_i and A_j is no arc and $d(A_i,A_i)=0$. Note now $\min_{1,1}$ - the minimum length of roads from A_i to A_s consists of a single arc. Obviously, they are in the column "s" of the matrix D. If we note now at the step p: $\min_{i,p}$ - the minimum length of roads from M_i to M_s consisting of at most p arcs, we have: $\min_{i,p}=\min_{k=\overline{1,n}}\left(d_{ik}+\min_{k,p-1}\right)$. It is clear that unless there is a path between M_i and M_s with at most p arcs we get $\min_{i,p}=\infty$. To do this, we will construct the matrix D_p obtained from the addition of each line of the matrix D of the vector $\min_{i,p-1}$. The vector $\min_{i,p}$ will be obtained from the matrix D_p by finding the minimum of the elements in the i-th line. The process is continued till we will obtain $\min_{i,p}=\min_{i,p-1}$, $i=\overline{1,n}$. Finally, the vector $\min_p=\left(\min_{1,p},...,\min_{n,p}\right)$ will have like components the minimal distances from A_s to each of the points A_k, $k=\overline{1,n}$. The sum of the components of $\min_p$ namely S_s will give total distancies from A_s to the other nodes. If we repeat

[1] Associate Professor, PhD, Danubius University of Galati, Corresponding author: catalin_angelo_ioan@univ-danubius.ro.
[2] Senior Lecturer, PhD, Danubius University of Galati, Department of Economics, E-mail: ginaioan@univ-danubius.ro.

this algorithm for each A_s, $s=\overline{1,n}$ and taking finally $\min_{s=\overline{1,n}} S_s$ we shall find the desired node – the center of the graph.

2. The Software

The authors wrote a software in Mathematica 9.0 in order to determine the center of the graph. For exemple, we consider the graph:

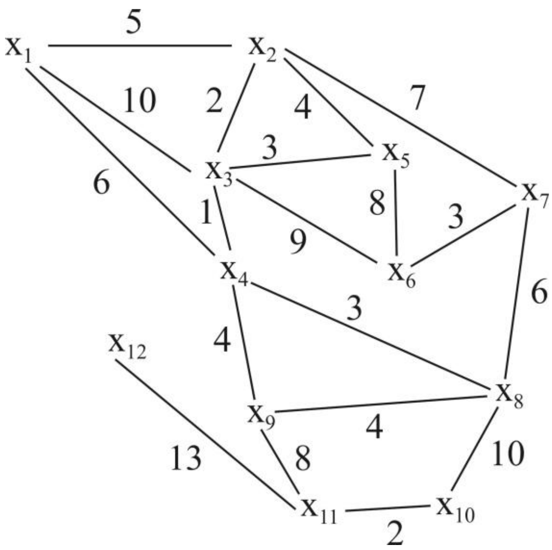

Figure 1

Clear[“Global`*”];

numberofnodes=12;

infinite=1000;

temporaryarray1=Table[0,{i,1,numberofnodes},{j,1,numberofnodes}];

temporaryarray2=Table[0,{i,1,numberofnodes},{j,1,numberofnodes}];

temporaryarray3=Table[0,{i,1,numberofnodes}];

(*The initialization of edge lengths*)

For[i=1,i<=numberofnodes,i++,For[j=1,j<=numberofnodes,j++,a[i,j]=infinite;vector[i,j]=0];

a[i,i]=0];

(*The introduction of the actual edges lengths*)

a[1,2]=5;a[1,3]=10;a[1,4]=6;a[2,3]=2;a[2,5]=4;a[2,7]=7;a[3,4]=1;a[3,5]=3;a[3,6]=9;a[4,8]=3;

a[4,9]=4;a[5,6]=8;a[6,7]=3;a[7,8]=6;a[8,9]=4;a[8,10]=10;a[9,11]=8;a[10,11]=2;a[11,12]=13;

For[i=1,i<=numberofnodes,i++,For[j=1,j<=i,j++,a[i,j]=a[j,i]]];

(*The determination of the amount of distances from one node to the others*)

determination[node_,step_]:=Module[{x=node, stepmodule=step},For[i=1,i<=numberofnodes,i++,

```
For[j=1,j<=numberofnodes,j++,temporaryarray1[[i]][[j]]=vector[stepmodule,j]+a[i,j]];
vector[stepmodule+1,i]=Min[temporaryarray1[[i]]];
temporaryarray2[[stepmodule+1]][[i]]=vector[stepmodule+1,i]]];
Print["Temporary computations:"]
For[nod=1,nod<=numberofnodes,nod++,
  For[i=1,i<=numberofnodes,i++,vector[1,i]=a[i,nod];temporaryarray2[[1]][[i]]=vector[1,i]];
  determination[nod,1];
  determination[nod,2];
  step=2;
While[temporaryarray2[[step]]□temporaryarray2[[step-1] ],determination[nod,step+1];step++];
sumdistances=0;For[i=1,i<=numberofnodes,i++,
sumdistances=sumdistances+temporaryarray2[[step+1]][[i]]];
  Print[temporaryarray2[[step+1]],", Total distancies=",sumdistances];
If[nod==1,minimal=sumdistances;minimalnode=1;
temporaryarray3[[step+1]]=temporaryarray2[[step+1]],
If[sumdistances<minimal,minimal=sumdistances;minimalnode=nod;
temporaryarray3=temporaryarray2[[step+1]]]]]];
Print["****************************************************************************
****"];
Print["Center of graph is the node",minimalnode, ", with total distancies=", minimal," and separated distancies:"];
Print[temporaryarray3]
```

The results of execution are:

```
Temporary computations:
{0, 5, 7, 6, 9, 15, 12, 9, 10, 19, 18, 31}, Total distancies=141
{5, 0, 2, 3, 4, 10, 7, 6, 7, 16, 15, 28}, Total distancies=103
{7, 2, 0, 1, 3, 9, 9, 4, 5, 14, 13, 26}, Total distancies=93
{6, 3, 1, 0, 4, 10, 9, 3, 4, 13, 12, 25}, Total distancies=90
{9, 4, 3, 4, 0, 8, 11, 7, 8, 17, 16, 29}, Total distancies=116
{15, 10, 9, 10, 8, 0, 3, 9, 13, 19, 21, 34}, Total distancies=151
{12, 7, 9, 9, 11, 3, 0, 6, 10, 16, 18, 31}, Total distancies=132
{9, 6, 4, 3, 7, 9, 6, 0, 4, 10, 12, 25}, Total distancies=95
{10, 7, 5, 4, 8, 13, 10, 4, 0, 10, 8, 21}, Total distancies=100
{19, 16, 14, 13, 17, 19, 16, 10, 10, 0, 2, 15}, Total distancies=151
{18, 15, 13, 12, 16, 21, 18, 12, 8, 2, 0, 13}, Total distancies=148
{31, 28, 26, 25, 29, 34, 31, 25, 21, 15, 13, 0}, Total distancies=278
********************************************************************************
Center of graph is the node 4, with total distancies=90 and separated distancies:
{6, 3, 1, 0, 4, 10, 9, 3, 4, 13, 12, 25}
```

Figure 2

3. Conclusions

The software presented above allows us to determine the graph center with applications in the location of control centers or distribution points of goods so that the sum of actual distances (on roads) being minimal.

4. References

Ioan, C.A. & Ioan, G. (2012). *Methods of mathematical modeling in economics*. Galați: Zigotto Publishers.

Ioan, C.A. & Ioan, G. (2014). The Localization Problem. *Euroeconomica*, no. 2(33), pp. 141-148.