

A Tutorial for Transforming a Line Shapefile to Pajek's Netfile, and then back to the shapefile

Yufan Miao

Department of Technology and Built Environment, Division of Geomatics
University of Gävle, 80176, Gävle, Sweden
Email: yufanmiao@gmail.com

Introduction

This tutorial is intended to provide a step by step guide to transform axial lines generated from line shapefile into a graph in Pajek's .net format so that Pajek can work as a complement to ArcGIS for the analysis of the relationships in an urban network. Axial lines are a key concept in space syntax theory; one can refer to another tutorial for details. Axwoman is an ArcGIS extension for the analysis of urban morphology based on the space syntax theory (<https://sites.google.com/site/axwoman60/>). Pajek is another free software tool for analyzing and visualizing large networks (<http://vlado.fmf.uni-lj.si/pub/networks/pajek/>). The net file is mainly based on node-node relationship which cannot be found directly from the information contained in a shapefile. Therefore, to realize the conversion, ArcGIS is adopted as a tool to search the relationship information and then write them into a net file.

In this tutorial, the relationship information is searched in ArcGIS through the call of built-in ArcObjects. More specifically, these ArcObjects are called and organized in VBA code to form a topology-search function first. The function is then activated by clicking a button which can be predefined through the customization option in ArcGIS. After the activation, the topological relationships are written into a file according to the .net file format so that it is recognizable by Pajek.

In summary, the idea behind the process can be divided into three steps: 1) build a button embedded in a toolbar; 2) write source code in VBA for the search function; 3) activate the search function by clicking the button predefined and get the generated .net file; and 4) add the attributes back to the shape file. To illustrate the details of the development and usage of such program, a fictive urban system containing 13 lines is employed as an example. The remaining texts will be organized in a step-by-step manner.

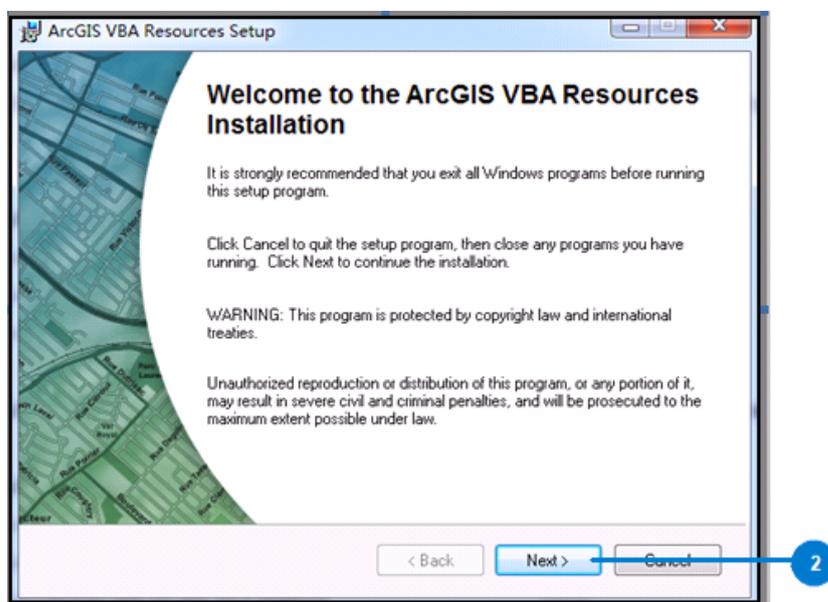
Setting the working environment

As has been mentioned in the introduction section, this program is written in VBA. The reason to adopt VBA as the programming language is that this language is employed by most versions (8, 9, and 10) of ArcGIS. Therefore, the program should be able to work in most cases. The VBA SDK is installed with ArcGIS for version 8 and 9. However, for ArcGIS 10 users, the SDK should be installed separately. The installation process will be introduced in the following paragraphs.

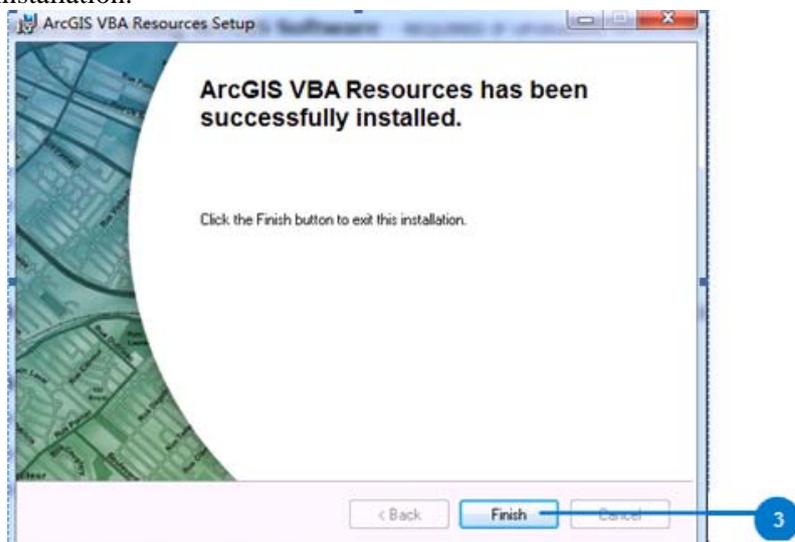
1. Insert your ArcGIS 10 installation disk and the setup program should be running automatically. If it is not, then go to your disk driver and double click to open it. Then an option window should pop out. In this window, choose "ArcGIS Desktop VBA Resources for Developers" to setup.



2. Then an installation window should pop up. Keep clicking “Next” until the installation process finishes.



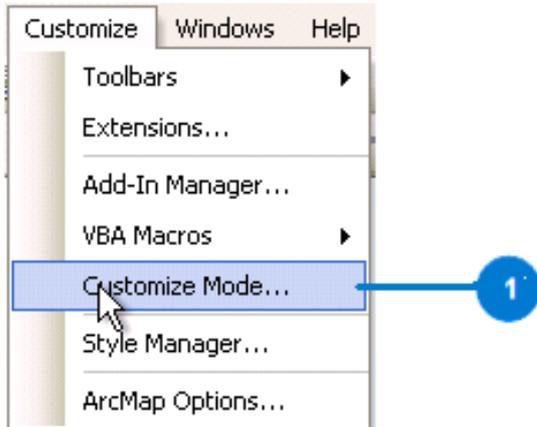
3. When the installation succeeds, a window will pop up to remind users. Just click finish and end the installation.



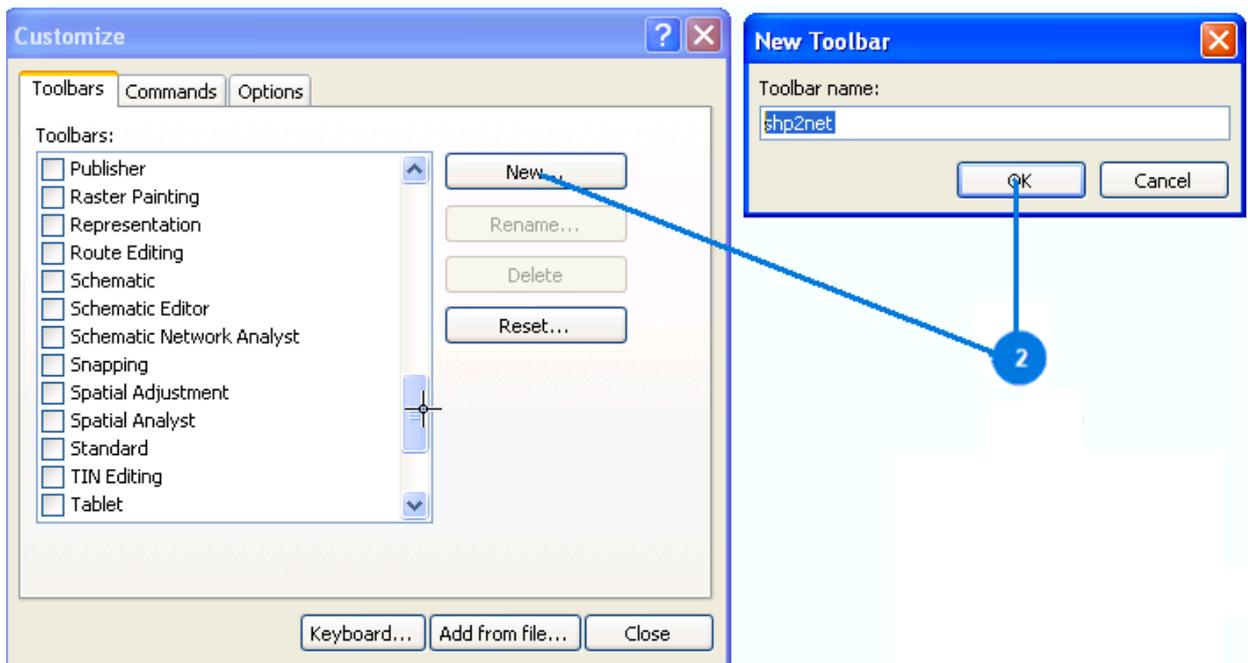
Define the button

The button is an activator to run the search function. Because it is a user interface control object, in ArcGIS, it is classified in the “UI Control” category which is specially designed to realize users’ customizations. In the following paragraphs, the process to build such a button will be explained step by step.

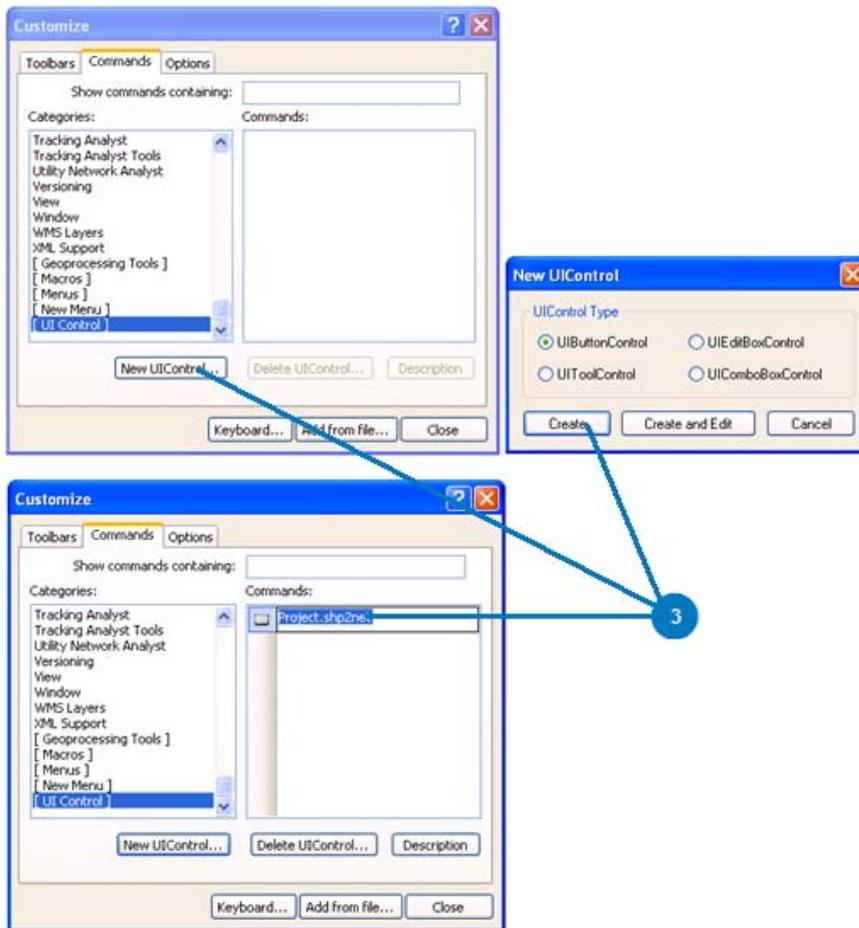
1. Enter the customization mode. Open ArcMap 10 and Click “Customize ⇒ Customize Mode” in the tool bar.



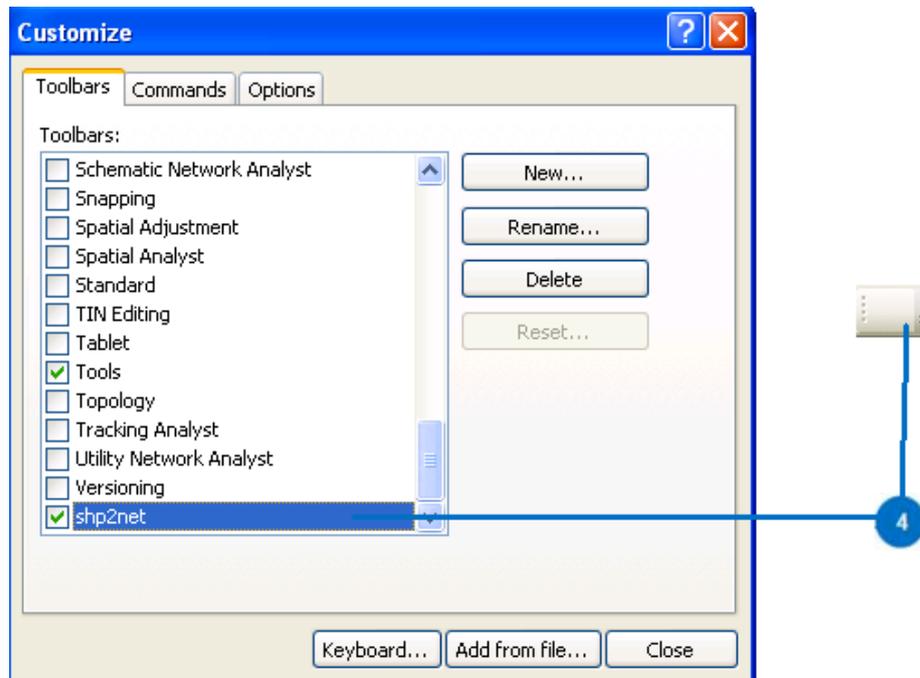
2. Build a toolbar. The toolbar is the container of the button, so it should be built up before the button. To build the toolbar, click new in the toolbar tab of the popped out window.



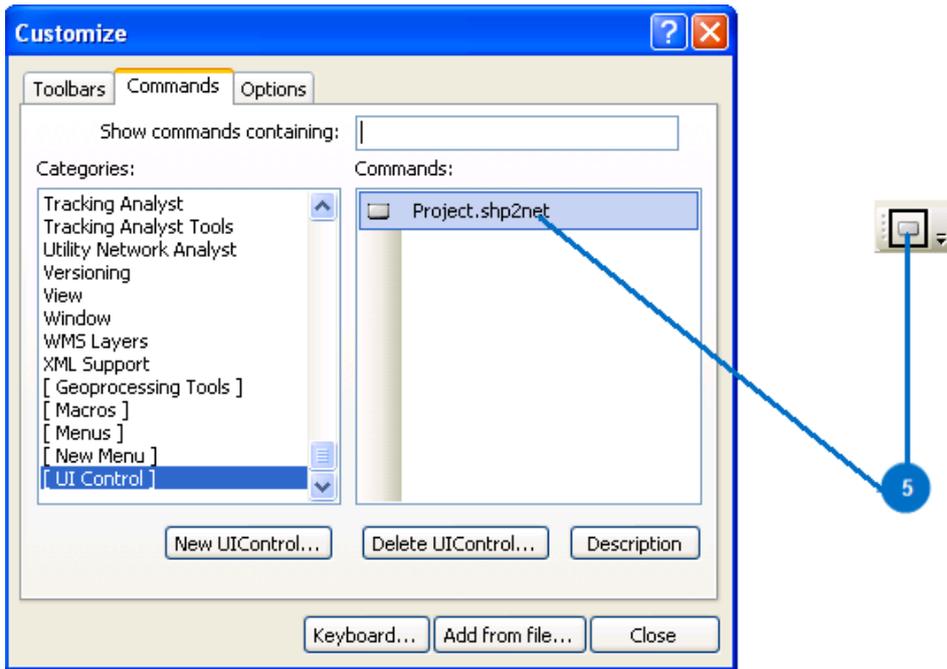
3. Build the button. To build the button, click the “Commands” tab, scroll down to the “UI Control” in the “Categories” box, and click “New UIControl”. In the popped up window choose “UIButtonControl” and then click “create”. Then a new UI control button shows up in the “Commands” box. Click it and change its name to “Project.shp2net”.



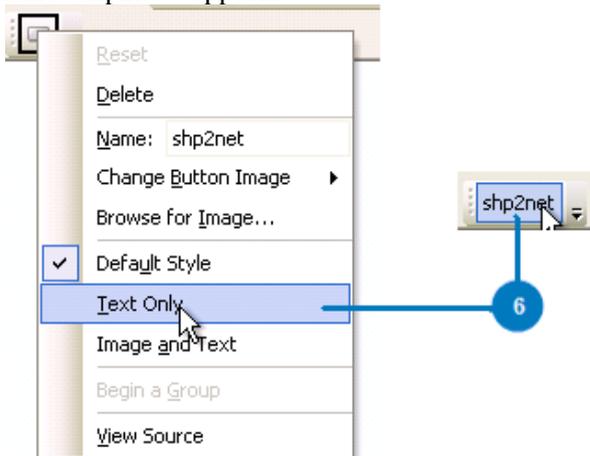
4. Drag the button to toolbar. Click the “Toolbar” Tab, scroll down the “Toolbar” box and check the toolbar “shp2net”. Then a blank tool bar can be seen in the ArcMap toolbar.



5. Switch back to the “Commands” tab, scroll down the “Categories” box and click “UI Control”. In the “Commands” box on the right, drag the button built just now to the blank tool bar.

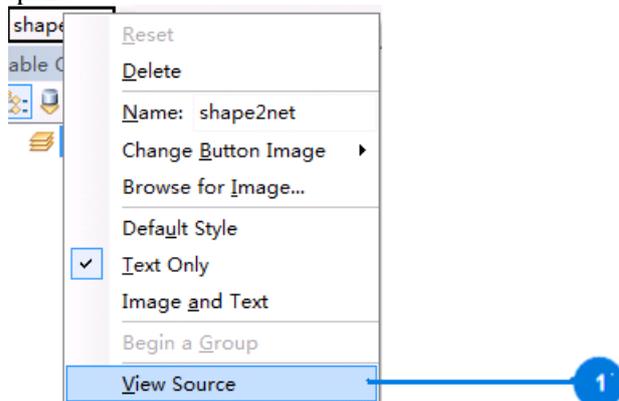


6. With the customize window open, right click the button and choose “Text Only” so that the button name “shp2net” appears.



Write source code for the function

1. Open the view source window. Right click the button and choose “View Source”. It should be noticed that this option, namely, “View Source” is only available when the customization window is open.



- Write a search function called “intersctme” with no parameters to pass in. All the following code except the button activating sub should be written within this function.

```
Sub intersectme()  
End Sub
```

- Define the target features. The defining process should go from general to specific. pMxDoc is defined as the document worked on. More specifically, pMap is defined as the focus map in this document. Further, pFLayer is defined as the vector layer in this map. Finally, pFClass is defined as the feature class containing all the features, in this case, 13 lines in this layer.

```
Dim pMxDoc As IMxDocument  
Set pMxDoc = ThisDocument  
  
Dim pMap As IMap  
Set pMap = pMxDoc.FocusMap  
  
Dim pFLayer As IFeatureLayer  
Set pFLayer = pMap.Layer(0)  
  
Dim pFClass As IFeatureClass  
Set pFClass = pFLayer.FeatureClass
```

- Define a cursor to point to the first feature. In the later loop, this helps to traverse all the features in the feature class until it points to no feature.

```
Dim pFCursor As IFeatureCursor  
Set pFCursor = pFClass.Update(Nothing, False)  
  
Dim pFeature As IFeature  
Set pFeature = pFCursor.NextFeature
```

- Define the spatial filter and set it as a filter to search for the intersection relationship.

```
Dim pSpatialFilter As ISpatialFilter  
Set pSpatialFilter = New SpatialFilter  
pSpatialFilter.SpatialRel = esriSpatialRelIntersects
```

- Define a file with a file number and a storing path and write the header for the file. In this case, the file number is a number free by the system. With this number, the contents can then be written into a file through the given path. In this case, the path where the file is stored is “C:\pajekdocument.net”. The header is written in the file with the vertices number as the count of the lines and the network type as an edge list.

```
Dim sFileName As String  
Dim iFileNum As Integer  
sFileName = "C:\pajekdocument.net"  
iFileNum = FreeFile()  
Open sFileName For Output As iFileNum  
Dim CountVertices As Long  
CountVertices = pFClass.FeatureCount(Nothing)  
Print #iFileNum, "*Vertices" & " " & CStr(CountVertices)  
Print #iFileNum, "*Edgeslist"
```

- Write the outer loop to traverse all the features in the feature class. In this loop, the geometry of the found intersection lines are copied and stored in the spatial filter. A string with the ID plus one of each feature in the feature class is written into the first column of the file under the edge list. The pFCursor makes sure the loop going on until no feature is found in the feature class. The inner loop to traverse all the lines intersected with each line in the feature class will be written after the statement “str = CStr(pFeature.OID+1)”.

```

Do Until pFeature Is Nothing
    Set pSpatialFilter.Geometry = pFeature.ShapeCopy
    pSpatialFilter.GeometryField = pFClass.ShapeFieldName
    Dim str As String
    str = CStr(pFeature.OID + 1)

    Print #iFileNum, str
    Set pFeature = pFCursor.NextFeature
Loop

```

7

8. Write the inner loop to traverse the lines intersected with each line in the feature class. Before looping through the lines intersected with the target lines, another feature cursor should be defined to ensure the running of the loop. Then the cursor is pointed to the line groups intersected with each target line and is prepared to traverse through all the intersected lines. Then the loop starts and writes every ID plus one of the lines intersected with each target line in a row behind the target line ID plus one.

```

Dim pFCursorIntersects As IFeatureCursor
Dim pFIntersects As IFeature
Set pFCursorIntersects = pFClass.Search(pSpatialFilter, False)
Set pFIntersects = pFCursorIntersects.NextFeature

Do Until pFIntersects Is Nothing
    If pFIntersects.OID > pFeature.OID Then
        str = str & " " & CStr(pFIntersects.OID + 1)
    End If
    Set pFIntersects = pFCursorIntersects.NextFeature
Loop

```

8

9. Close the file and activate the button. The file should be closed and the document should be refreshed to ensure that a new round of search or other functions would not be influenced by the results generated for this time. Moreover, the event click button should be used to call the function "intersectme" so that the button predefined can work.

```
Close iFileNum
```

```
pMxDoc.ActivatedView.Refresh
```

9

```

Private Sub shp2net_Click()
    Call intersectme
End Sub

```

10. An overview of the whole source code.

```

Sub intersectme()
    Dim pMxDoc As IMXDocument
    Set pMxDoc = ThisDocument
    Dim pMap As IMap
    Set pMap = pMxDoc.FocusMap
    Dim pFLayer As IFeatureLayer
    Set pFLayer = pMap.Layer(0)
    Dim pFClass As IFeatureClass
    Set pFClass = pFLayer.FeatureClass
    Dim pFCursor As IFeatureCursor
    Set pFCursor = pFClass.Update(Nothing, False)
    Dim pFeature As IFeature
    Set pFeature = pFCursor.NextFeature
    Dim pSpatialFilter As ISpatialFilter
    Set pSpatialFilter = New SpatialFilter
    pSpatialFilter.SpatialRel = esriSpatialRelIntersects
    Dim sFileName As String
    Dim iFileNum As Integer
    sFileName = "C:\pajekdocument.net"
    iFileNum = FreeFile()
    Open sFileName For Output As iFileNum
    Dim CountVertices As Long
    CountVertices = pFClass.FeatureCount(Nothing)
    Print #iFileNum, "*"Vertices" & " " & CStr(CountVertices)
    Print #iFileNum, "*"Edgeslist"

```

```

Do Until pFeature Is Nothing
    Set pSpatialFilter.Geometry = pFeature.ShapeCopy
    pSpatialFilter.GeometryField = pFClass.ShapeFieldName
    Dim str As String
    str = CStr(pFeature.OID + 1)
    Dim pFCursorIntersects As IFeatureCursor
    Dim pFIntersects As IFeature
    Set pFCursorIntersects = pFClass.Search(pSpatialFilter, False)
    Set pFIntersects = pFCursorIntersects.NextFeature
    Do Until pFIntersects Is Nothing
        If pFIntersects.OID > pFeature.OID Then
            str = str & " " & CStr(pFIntersects.OID + 1)
        End If
        Set pFIntersects = pFCursorIntersects.NextFeature
    Loop
    Print #iFileNum, str
    Set pFeature = pFCursor.NextFeature
Loop
Close iFileNum
pMxDoc.ActivatedView.Refresh
End Sub

Private Sub shp2net_Click()
    Call intersectme
End Sub

```

10

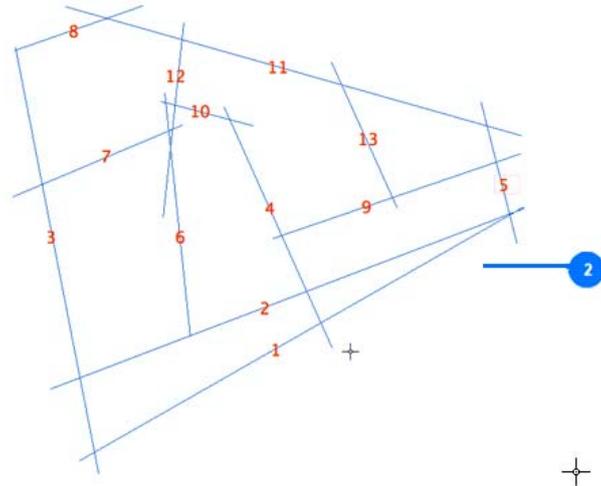
Activate the button and read the file in Pajek

1. An overview and explanation of the structure of the .net file. In the first line, the vertices number is described: 13 vertices representing 13 lines. In the second line the type of the network is defined: it is an edges list which means the network has no directions. In the following matrix, the first column is the IDs of the target lines with the lines's IDs intersected with each target line following behind in each row.

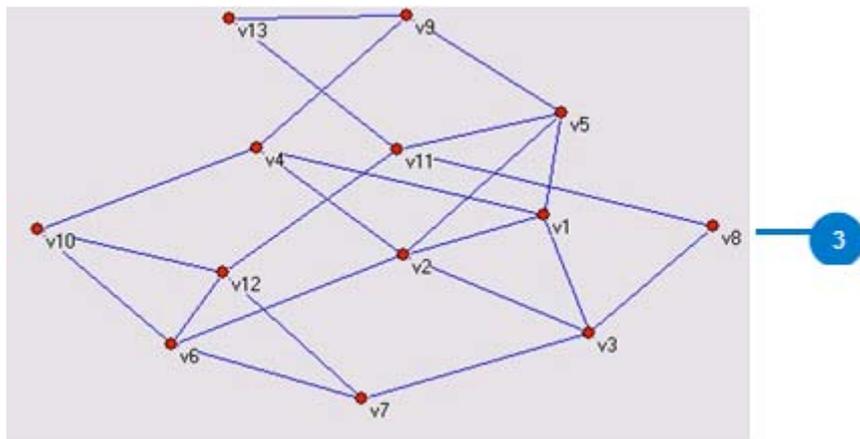
```
*Vertices 13
*Edgeslist
1 2 3 4 5
2 3 4 5 6
3 7 8
4 9 10
5 9 11
6 7 10 12
7 12
8 11
9 13
10 12
11 12 13
12
13
```



2. Import the shapefile into ArcGIS desktop and run the program. After activating the button, add the shapefile to ArcGIS Desktop, then click the button and the generated file can be found in drive C with a name “pajekdocument.net”.



3. Load the generated file into Pajek and draw the network. The file can be read by Pajek directly and drawn into networks. In this case, a KK-free style is employed.



- Then execute excel 2003 and open the dbf file.

ID	FROMOBJID	TOOBJID	CONNECT	CONTROL	MEANDEPTH	GINTEG	LINTEG	TOTALDEPTH	LOCALDEPTH
0	8	8	4	0.9500	1.8333	1.8185	2.2117	22	16
0	8	8	5	1.2500	1.6667	2.2731	2.6115	20	17
0	8	8	4	1.2833	1.9167	1.6532	2.2000	23	14
0	3	8	4	1.1167	1.9167	1.6532	2.2000	23	14
0	8	8	4	1.0333	1.8333	1.8185	2.2117	22	16
0	0	6	4	1.1167	1.9167	1.6532	2.2000	23	14
0	2	8	3	0.7500	2.1667	1.2989	1.7740	26	13
0	8	8	2	0.5000	2.1667	1.2989	1.4784	26	14
0	8	2	3	1.0000	2.2500	1.2123	1.7239	27	11
0	0	3	3	0.7500	2.0833	1.3989	1.7740	25	13
0	8	8	4	1.5000	1.7500	2.0206	2.2384	21	18
0	8	1	4	1.1667	1.8333	1.8185	2.2117	22	16
0	8	5	2	0.5833	2.3333	1.1366	1.2737	28	10

3

- Copy the column of values for each field from the attribute files and paste them in excel. Then click save and choose yes when the warning box appears when saving the file. Choose No when the warning box appears again when closing excel.

ID	FROMOBJID	TOOBJID	CONNECT	CONTROL	MEANDEPTH	GINTEG	LINTEG	TOTALDEPTH	LOCALDEPTH	DEGREE	BETWEEN	CLOSE
0	8	8	4	0.9500	1.8333	1.8185	2.2117	22	16	4	0.0543	0.5455
0	8	8	5	1.2500	1.6667	2.2731	2.6115	20	17	5	0.1346	0.6000
0	8	8	4	1.2833	1.9167	1.6532	2.2000	23	14	4	0.1169	0.5217
0	3	8	4	1.1167	1.9167	1.6532	2.2000	23	14	4	0.1199	0.5217
0	8	8	4	1.0333	1.8333	1.8185	2.2117	22	16	4	0.1174	0.5455
0	0	6	4	1.1167	1.9167	1.6532	2.2000	23	14	4	0.0652	0.5217
0	2	8	3	0.7500	2.1667	1.2989	1.7740	26	13	3	0.0376	0.4615
0	8	8	2	0.5000	2.1667	1.2989	1.4784	26	14	2	0.0303	0.4615
0	8	2	3	1.0000	2.2500	1.2123	1.7239	27	11	3	0.0646	0.4444
0	0	3	3	0.7500	2.0833	1.3989	1.7740	25	13	3	0.0487	0.4800
0	8	8	4	1.5000	1.7500	2.0206	2.2384	21	18	4	0.2056	0.5714
0	8	1	4	1.1667	1.8333	1.8185	2.2117	22	16	4	0.1316	0.5455
0	8	5	2	0.5833	2.3333	1.1366	1.2737	28	10	2	0.0247	0.4286

4

- The attribute table of the shapefile before adding new columns can be seen below.

FID	Shape *	ID	FROMOBJID	TOOBJID	CONNECT	CONTROL	MEANDEPTH	GINTEG	LINTEG	TOTALDEPTH	LOCALDEPTH	DEGREE	BETWEEN	CLOSE
0	Polyline	0	8	8	4	0.95	1.8333	1.8185	2.2117	22	16	0	0	0
1	Polyline	0	8	8	5	1.25	1.6667	2.2731	2.6115	20	17	0	0	0
2	Polyline	0	8	8	4	1.2833	1.9167	1.6532	2.2	23	14	0	0	0
3	Polyline	0	3	8	4	1.1167	1.9167	1.6532	2.2	23	14	0	0	0
4	Polyline	0	8	8	4	1.0333	1.8333	1.8185	2.2117	22	16	0	0	0
5	Polyline	0	0	6	4	1.1167	1.9167	1.6532	2.2	23	14	0	0	0
6	Polyline	0	2	8	3	0.75	2.1667	1.2989	1.774	26	13	0	0	0
7	Polyline	0	8	8	2	0.5	2.1667	1.2989	1.4784	26	14	0	0	0
8	Polyline	0	8	2	3	1	2.25	1.2123	1.7239	27	11	0	0	0
9	Polyline	0	0	3	3	0.75	2.0833	1.3989	1.774	25	13	0	0	0
10	Polyline	0	8	8	4	1.5	1.75	2.0206	2.2384	21	18	0	0	0
11	Polyline	0	8	1	4	1.1667	1.8333	1.8185	2.2117	22	16	0	0	0
12	Polyline	0	8	5	2	0.5833	2.3333	1.1366	1.2737	28	10	0	0	0

5

- Add the shapefile in ArcGIS. Right click its layer and choose "Attribute Table".

FID	Shape *	ID	FROMOBJID	TOOBJID	CONNECT	CONTROL	MEANDEPTH	GINTEG	LINTEG	TOTALDEPTH	LOCALDEPTH	DEGREE	BETWEEN	CLOSE
0	Polyline	0	8	8	4	0.95	1.8333	1.8185	2.2117	22	16	4	0.0543	0.5455
1	Polyline	0	8	8	5	1.25	1.6667	2.2731	2.6115	20	17	5	0.1346	0.6
2	Polyline	0	8	8	4	1.2833	1.9167	1.6532	2.2	23	14	4	0.1169	0.5217
3	Polyline	0	3	8	4	1.1167	1.9167	1.6532	2.2	23	14	4	0.1199	0.5217
4	Polyline	0	8	8	4	1.0333	1.8333	1.8185	2.2117	22	16	4	0.1174	0.5455
5	Polyline	0	0	6	4	1.1167	1.9167	1.6532	2.2	23	14	4	0.0652	0.5217
6	Polyline	0	2	8	3	0.75	2.1667	1.2989	1.774	26	13	3	0.0376	0.4615
7	Polyline	0	8	8	2	0.5	2.1667	1.2989	1.4784	26	14	2	0.0303	0.4615
8	Polyline	0	8	2	3	1	2.25	1.2123	1.7239	27	11	3	0.0646	0.4444
9	Polyline	0	0	3	3	0.75	2.0833	1.3989	1.774	25	13	3	0.0487	0.48
10	Polyline	0	8	8	4	1.5	1.75	2.0206	2.2384	21	18	4	0.2056	0.5714
11	Polyline	0	8	1	4	1.1667	1.8333	1.8185	2.2117	22	16	4	0.1316	0.5455
12	Polyline	0	8	5	2	0.5833	2.3333	1.1366	1.2737	28	10	2	0.0247	0.4286

6