Urban Network Analysis

A Toolbox for ArcGIS 10

HELP

April 2, 2012

Andres Sevtsuk; Michael Mekonnen
Contact:
cityform_rg@ mit.edu

Singapore University of Technology & Design in collaboration with MIT
Room L3R3
20 Dover Drive.
Singapore, 138682

Cover Image: Jason Hawkes

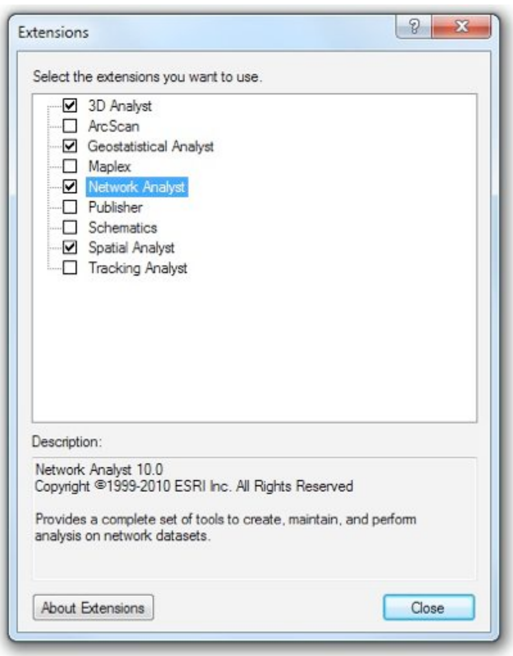
CONTENTS

INSTALLATION GUIDELINES

Prerequisites

Running the toolbox requires that your Windows operating system uses English as the operating language. If your Windows system is set to another language, you can (temporarily) change your Windows settings to English and then run the toolbox. You can find instructions on how to do that here.

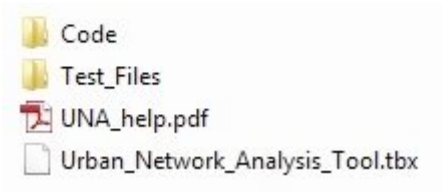
Using the Urban Network Analysis toolbox requires that you have ArcGIS 10 and the *Network Analyst* extension. If you have a working license for the Network Analyst extension, you can enable the extension by going to the *Customize* menu → *Extensions...* → and check the box next to Network Analyst.



In order for the tools to work correctly in ArcGIS 10, please make sure you have also downloaded and installed the Service Pack 2 for ArcGIS 10 Desktop. The original release of ArcGIS 10 has a known bug that prevents script results from adding to the display.

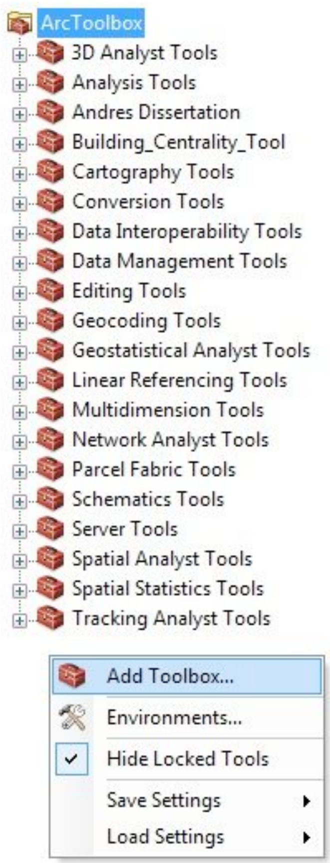
Download

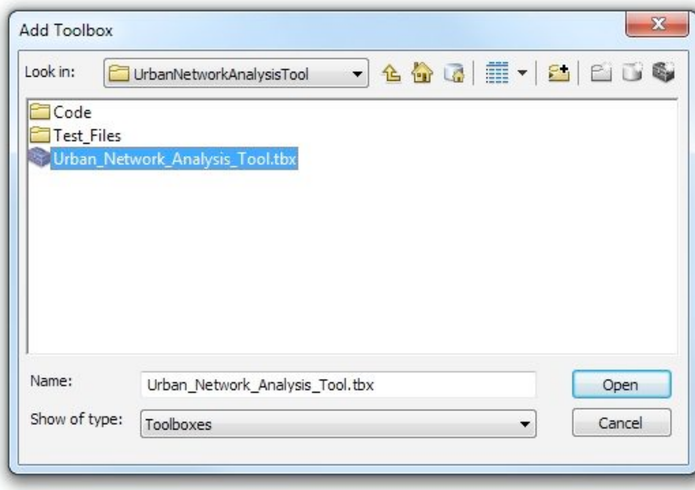
Download the zip folder (*UrbanNetworkAnalysis.zip*) to a location of your choice on your computer and unzip its contents. You will find a toolbox file (*UrbanNetworkAnalysis.tbx*), help documentation (UNA_help.pdf), and two subfolders - *Code* and *Test_files*. The *Code* folder stores all the scripts that are needed for the UNA tools to work, while the *Test_files* folder contains a few sample network datasets and point shapefiles that you can use to test the tools. You can delete this folder if you are not interested in using the test files.



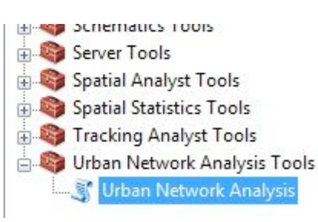
Installation

To install the toolbox, open ArcMap 10, make sure the toolbox tab is visible, and right-click inside the toolbox tab. Choose *Add Toolbox...* , then choose the downloaded *UrbanNetworkAnalysis.tbx* file from where you saved it, and click Open.

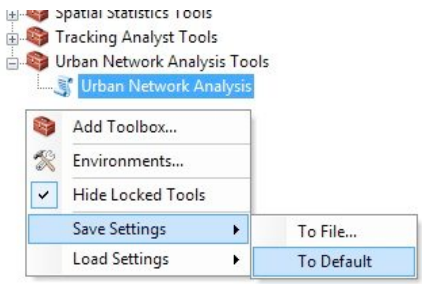




The urban Network Analysis toolbox should now appear in your toolbox list, as shown below.



If you would like to have ArcGIS load this toolbox by default along with other toolboxes every time you open the program, then right-click again inside the toolbox tab, choose *Save Settings → To Default*.



Test Files

The download zip folder (*Urban Network Analysis Tool.zip*) contains a street network file for Cambridge and Somerville MA (*streets3.ND*) and a building polygon and point shapefiles for the same area to getting started. We also include the Junctions shapefile for Cambridge/Somerville in case you want to test the tools with just nodes and edges (without buildings), as well as a network and building centroids for lower Manhattan in NYC. These files can be used as Input Buildings and Input Analysis Networks respectively in order to test the toolbox functions.

1. DESCRIPTION

This ArcGIS toolbox can be used to compute five types of network analysis measures on spatial networks: *Reach; Gravity; Betweenness*; *Closeness*; and *Straightness*. The toolbox builds upon previous efforts by the Martin Center for Built Form and Land Use Studies, the Space Group at the UCL, the Human Space Lab, and other research work in the area of spatial network analysis (Martin & March 1971; Hillier & Hanson 1984; Porta, Crucitti et al, 2005). It is aimed at urban designers, architects, planners, geographers, and spatial analysts who are interested in studying the spatial configurations of cities, and their related social, economic, and environmental processes.

Unlike topological network analysis tools, where geometric relationships between network elements have little meaning (e.g. relationships to a number of contacts in a phonebook can all be considered as equal network links, regardless of how far in space the contacts are located), spatial network analysis relies on an accurate consideration of distance and angularity between places. The UNA tools incorporate three important features that make them suitable for spatial network analysis. First, they can account for both geometry and topology in the input networks, using either metric distance (e.g. Meters) or topological distance (e.g. Turns) as impedance factors in the analysis. Second, unlike previous software tools that operate with two network elements (*nodes* and *edges*), the UNA tools include a third network element - *buildings* - which are used as the spatial units of analysis for all measures.[1] Two neighboring buildings on the same street segments can therefore obtain different accessibility results. And third, the UNA tools optionally allow buildings to be weighted according to their particular characteristics - more

[1] If building-level analysis is not desired, then the input points can also designate street intersections as nodes, producing the equivalent of the traditional node-edge type analysis.

voluminous, more populated, or otherwise more important buildings can be specified to have a proportionately stronger effect on the analysis outcomes, yielding more accurate and reliable results to any of the specified measures.

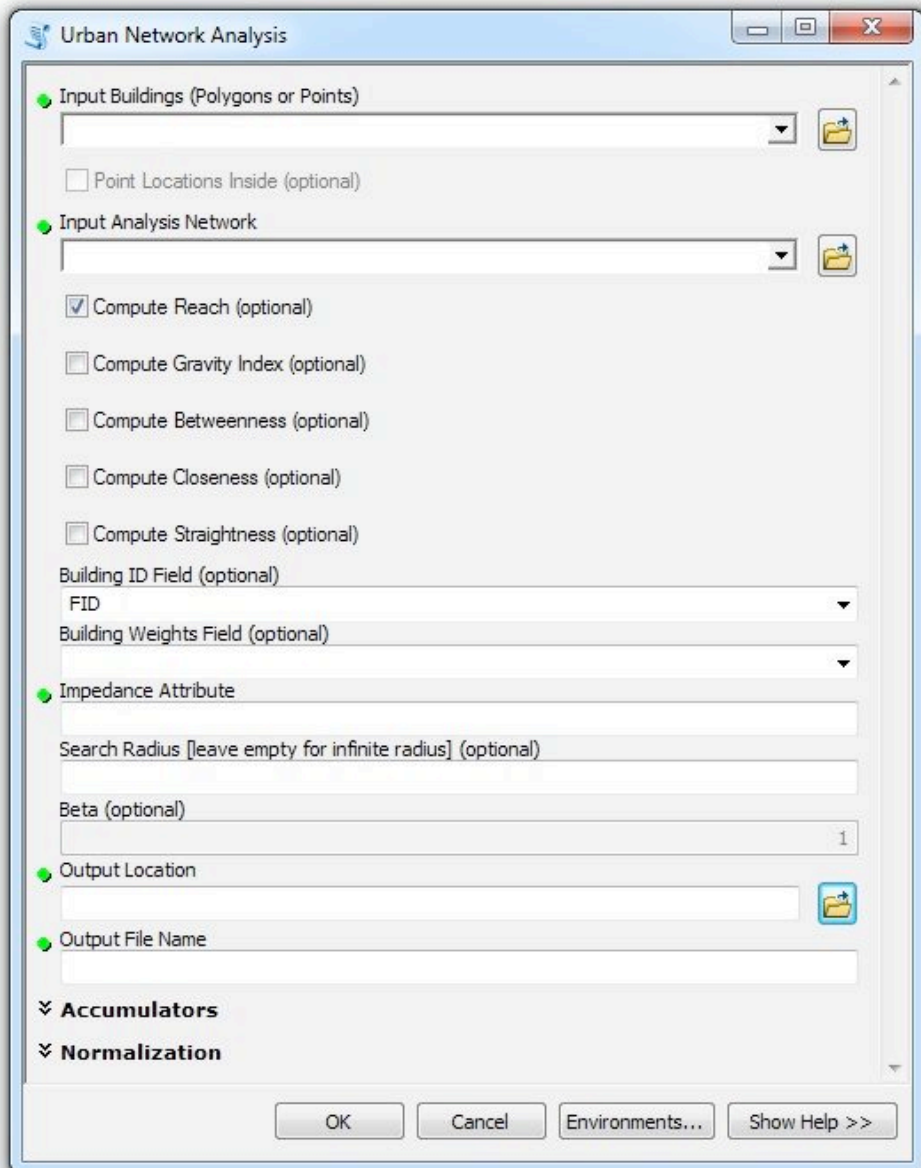


Figure 1: Graphic User Interface

2. TOOLBOX INPUTS

2.1. Input Buildings (Polygons or Points)

A polygon shape file[2] marking the locations of building footprints or a point shapefile (usually building street entrances or footprint centroids) for which the computation will be carried out. These buildings are used as the nodes of the graph on which we run the network analysis. From the Input Buildings shapefile we also obtain the weights for the buildings (if the user chooses to provide them). When using Polygons as inputs, pleae note that an ArcInfo license is required on your ArcGIS copy in order to execute the Feature to Point procedure. The ArcInfo license is not required if you use points as inputs.

In case the input is given as polygons, we assume that each building connects to a street (edge) that lies closest to it along the shortest perpendicular connection from the centroid of the polygon. This assumption can be erroneous in building polygons whose centroids lie closest to side streets, rather than the streets of their true entrances. To overcome this bias, a user can input points to represent building locations instead, making sure that the points reflect the true locations of the building or parcel entrances. The user can easily convert the polygons to point centroids by using ArcGIS' built-in *Feature To Point* (Data Management) tool.

By ticking the "Point Location Inside" option next to the Input Buildings, the centroid will always be placed inside the perimeter of the polygon.

In order to accurately model buildings that have multiple entrances on different streets, a user can provide a different input point for each entrance and divide the total attribute weights of the corresponding building by the number of entrances used. The end results of network centrality computations for multiple entrance points belonging to the same building should eventually be summed.

If you wish to compute the graph centrality measures for nodes of the network instead of buildings, then nodes illustrating street intersections can also be used as inputs.

2.1.1 Selections

Since the release of April 2, 2012, Input Buildings can also handle selections. If the user selects only a subset of Input Buildings, either manually or using a GIS query, and executes the UNA tool while the selection is active, then only the selected Input Buildings will be given Reach, Gravity, Closeness and Straightness results (Please see the implications for Betweenness below). All other Input Buildings that are not actively selected are still used as potential destinations in the analysis, but are do not assigned results themselves. This allows the user to

[2] The Input Points should be a point shapefile (.shp), not a Geodatabase (GDB) Featureclass.

focus the analysis on only desired input features and speeds up the calculation of centrality metrics. Note, however, that if the tool is used for the first time on a given dataset with the given parameter choices, then the usual adjacency matrix still needs to be calculated for the whole dataset, not only the selected buildings.

In case the Betweenness measure is used with a selection, then only the selected Input Buildings are used as origins i in the analysis. The Betweenness results are computed for all Input Buildings, not only the selected ones, but only the selected ones are used as starting points of shortest-path trips to all surrounding destinations. Since the set destinations to which trips are assigned can also be controlled using weights (see Section 5), then using a selection with Betweenness analysis allows the user to control both the origins and destinations of trips used in a Betweenness analysis, while computing the results for all buildings, not only the selected or weighted ones.

One scenario, where this feature may be used, is in estimating passersby of a particular type between given orgings and destinations. If a selection contains only subway stations, and the analysis is weighted by the number of office workers in each buidlings, then the results can help estimate how many times each building is passed by office workers on their way from subway stations to office buildings along shortest paths. Similar applications can be configured for other types of origins and destinations.

Note that if only the Betweeness measure is used with a selection and no other metrics are simultaneously chosen, then no normalization option is offered for the Betweenness metrics. This is because normalization requires the Reach measure, which in this case, would not be computed.

2.2. Network Dataset

The street network in which the buildings are located. This input requires a Network Dataset file of the street network with an **.nd* extension. If you do not have a Network Dataset file of your street network yet, then you can easily create a Network Dataset from your *.shp, *.dwg, or *.dxf files using *ArcMap* or *ArcCatalogue*. ArcGIS version 10 makes it particularly easy to convert a polyline layer to a Network Daset. Simply open the ArcCatalogue Tree in ArcMap10 (Figure 2), navigate to the polyline file that contains the appropriate network file, right-click, choose *New Network Dataset*, and follow the instructions (Figure 3). Before running the UNA tools, ensure visually that the Input Points are on the right locations on the Network Dataset.

Figure 2: Opening the Arc Catalogue Tree in Arc Map 10

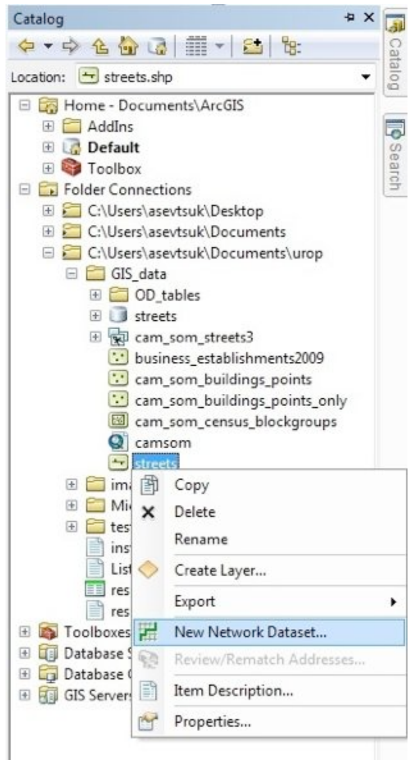


Figure 3: Creating a New Network Dataset from a polyline shapefile in ArcMap 10.

3. CHOICE OF ANALYSIS

The user can choose which of the five network analysis measurements are to be computed by checking the boxes next to the five measures: R*each, Gravity, Betwenness, Closeness*, and *Straightness*. These measures are described below.

3.1. Reach

The "Reach" measure (Sevtsuk, 2010) captures how many surrounding buildings each building reaches within a given *Search Radius* on the network.[3] The reach centrality, $R^r[i]$, of a building $i$in a graph G describes the number of other buildings in Gthat are reachable from i at a shortest path distance of at most r. It is defined as follows:

$$R^r[i] = \|\{j \in G - \{i\} : d[i,j] \le r\}\|$$

[3]Due to computational difficulties, we do not here model buildings with multiple entrances, which can play an important role for building accessibilities in a real-life urban settings.

where $d[i,j]$ is the shortest path distance between nodes i and j in G, and $\|S\|$ is the cardinality of the set S. If the nodes in G are weighted, then reach is defined as follows:

$$Reach[i]^r = \sum_{j \in G-\{i\}, d[i,j] \leq r} W[j]$$

where $W[j]$ is the weight of node j.[4] Figure 4 illustrates how the Reach index is calculated visually. A buffer is traced from each building i in every direction on the street network until the limiting radius r is reached. The Reach index corresponds to the number of destinations j (represented as smaller black points) that are found within the radius on the street network.

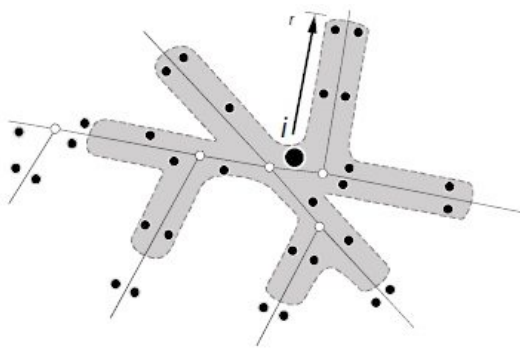


Figure 4: Illustration of the Reach measure.

Reach can be calibrated to measure access to any type of destination. In order to simply compute how many other buildings are reached within the given *Search Radius*, the *Node Weight Attribute Name* input box can be left blank, so that no weighting will be applied and only the count of destination buildings will be returned. To weight the measure by building size, for instance, the user can specify a building volume attribute column as the Node Weight Attribute Name.[5] The Reach measure would, in this case, compute how much built volume is reached within the Search Radius around each building in the network. To capture Reach to activities or land use destinations, one can use the number of jobs, the number of residents, or the number of business establishments in the surrounding buildings as *Node Weight Attributes*.

3.2. Gravity

Whereas the Reach measure simply counts the number of destinations around each building within a given Search Radius (optionally weighted by building attributes), the Gravity measure additionally factors in the spatial impedance required to travel to each of the destinations. First introduced by Hansen (1959), the Gravity index remains one of the most popular spatial accessibility measures in transportation research.

---

[4] The Reach measure we use is identical to a cumulative opportunities type accessibility index, described by Bhat et al. (2007), but applied on a network rather than Euclidian space.

[5] Assuming that such information is available in the Input Points attribute table.

The Gravity measure assumes that accessibility at building i is proportional to the attractiveness (weight) of destinations j surrounding i, and inversely proportional to the distances between i and j:

$$Gravity[i]^r = \sum_{j \in G-\{i\}, d[i,j] \le r} \frac{W[j]}{e^{\beta \cdot d[i,j]}}$$

where $Gravity[i]^r$ is the Gravity index at building i within graph G at Search Radius r (specified in the *Search Radius* box), $W[j]$ is the weight of destination j, $d[i,j]$ is the geodesic distance between buildings i and j, and β is the exponent for adjusting the effect of distance decay. The gravity type index thus captures both the attraction of the destinations ($W[j]$) as well as the spatial impedance of travel required to reach those destinations ($d[i,j]$) in a combined measure of accessibility.[6] If no *Node Weight Attributes* are given, then the weight of each destination building is considered to be "1".

The inverse effect of distance specified in the Gravity index decreases exponentially. The exact shape of the distance decay can be controlled with the exponent β, specified in another input field below. β and the corresponding shape of distance decay should be derived from the assumed mode of travel - for walking measured in "minutes", for instance, researchers have found β to fall around 0.1813 (Handy and Niemeier 1997)[7]. If no input for beta is specified by the user, then the default entry of "0" is used automatically, and no distance decay is applied.

3.3. Betweenness

The *Betweenness* of a building is defined as the fraction of shortest paths between pairs of other buildings in the network that pass by building i (Freeman 1977). If more than one shortest path is found between two nodes, as is frequently the case in a rectangular grid of streets, then each of the equidistant paths is given equal weight such that the weights sum to unity. The *Betweenness* measure is defined as follows:

$$Betweenness[i]^r = \sum_{j,k \in G-\{i\}, d[j,k] \le r} \frac{n_{jk}[i]}{n_{jk}} \cdot W[j]$$

---

[6] Consider two homes nearby a retail store. The first home is located a mile away, but the second home only half a mile away from the store. Even though both homes have the same number of store destinations available (one) and the destinations are identical in weight, the gravity index would consider the closer home to be more retail accessible than the further home.

[7] The equivalent value of Beta for impedance units in "meters" is 0.00217; in "feet" 0.000663; in "kilometers" 2.175, and in "miles" 3.501.

where $Betweenness[i]^r$ is the betweenness of building i within the Search Radius r (specified in the *Search Radius* box); $n_{jk}[i]$ is the number of shortest paths from node j to node k that pass by node i; and n_{jk} is the total number of shortest paths from j to k. The *Betweenness* measure is typically used to estimate the potential of passersby at different buildings on the network. If the analysis is weighted by Node Weight Attributes - demographics of a certain type in the surrounding buildings for instance - then the *Betweenness* can capture the potential of passersby of that particular demographic at building i.

The UNA tools use a very fast algorithm for computing Betweenness, originally developed by Brandes (2001).

3.4 Closeness

The *Closeness* of a *Input Building* is defined as the inverse of cumulative distance required to reach from that building to all other buildings in the system that fall within the S*earch Radius* along the shortest paths (Sabidussi 1966). Whereas the *Betweenness* measure indicates the potential traffic passing a building, the *Closeness* measure indicates how close a building is to all other surrounding buildings within a given distance threshold. If no *Search Radius* is provided, then the index is calculated to all other *Input Buildings* in system. The *Closeness* measure is defined as follows:

$$Closeness[i]^r = \frac{1}{\sum_{j \in G-\{i\}, d[i,j] \leq r} (d[i,j] \cdot W[j])}$$

where $Closeness[i]^r$ is the *Closeness* of building i within the *Search Radius* r (specified in the *Search Radius* box), $d[i,j]$ is the shortest path distance between nodes i and j, and $W[j]$ is the weight of the destination building j.

3.5 Straightness

The *Straightness* metric (Vragovic, Louis, et al. 2005) illustrates the extent to which the shortest paths from a node of interest to all other nodes in the system resemble straight Euclidian paths. Put alternatively, the *Straightness* metric captures the positive deviations in travel distances that result from the geometric constraints of the street network in comparison to straight-line distances in a featureless plane. The *Straightness* measure is formally defined by by (Porta, Crucitti et al. 2005) as:

$$Straightness[i]^r = \sum_{j \in G-\{i\}, d[i,j] \leq r} \frac{\delta[i,j]}{d[i,j]} \cdot W[j]$$

where $Straightness[i]^r$ is the Straightness of building i within the *Search Radius* r, $\delta[i,j]$ is the straight-line Euclidian distance between buildings i and j, and $d[i,j]$ is the shortest network distance between the same buildings. The *Straightness* index essentially illustrates how long the shortest path connections from each building to the surrounding points j are in comparison to the as-a-crow-flies distance. Naturally, as the distances between nodes get longer, the differences between the network distance and as-a-crow-flies distance start diminishing – a walk from Boston to Los Angeles is much closer to a straight line than a walk from MIT to downtown Boston.

Note, that the Straightness index should only be used if the impedance attribute is in the form of linear distance.

4. Building ID Attribute Name

This is the attribute name used to distinguish between buildings (Input Building names). *Building ID* names have to be an integer field. Users can choose any integer column in the *Input Points* shapefile as building IDs, and the analysis results are returned by the chosen Building ID column. If no specific IDs are chosen, then the FID field is used by default.

5. Building Weights Field

This input allows the user to choose an attribute column that exists in the Input Points shapefile to be used as building weights $W[j]$ in the analysis. *Input Building* weights can describe any meaningful numeric characteristics of buildings– building size, number of residence, the presence of businesses etc. When weights are chosen for the analysis, the results are returned in weighted form according the chosen attributes. If the number of employees in each building are chosen as weights for the *Reach* measure, for instance, then the results will return the total number of employees that can reached within the *Search Radius* around each building in the network.

6. Impedance Attribute

The *Impedance Attribute* selection designates which impedance characteristic associated with the Input Network will be used in all calculations to limit network radii (r) and shortest path computations. The default value *'Length'* sets linear distance as the impedance attribute, so that the *Search Radius*and shortest path computations use linear distance (e.g. meters) as the impedance factor. If *'Turns'* is used as the Impedance Attribute instead, then these radii will be limited by the number of turns required to reach the destinations, not distance. The Search

Radius value of "*2 Turns*", for instance, will tell the algorithm to search for neighboring buildings that are up to 2 turns away from each origin along the given street network.[8]

The choice of available impedance attributes are tied to the Input Analysis Network – they have to be [hard-coded to the network](). The following steps illustrate how to hard-code Turn-Count and Junction-Count impedance attributes to a Network Dataset:

A. Open (or create a new) network dataset (ND) in ArcCatalog
B. Riht-Click the ND fiel → Properties → Attributes → Add (when creating a new ND, the default settings will automatically generate a length attribute)
C. Make the newly added attribute a Cost Attribute / Unknown Units / Integer
D. Go to the Evaluators tab → Default Values
E. Go to Element "Turn" / Type "VB Script" / Default → Click Evaluator Properties
F. Enter the following VB script:

```
turnscount = 0
a = Turn.Angle
If a >= 20 And a<= 340 Then
   turnscount = 1
Else
   turnscount = 0 End If

Value= turnscount
```

This script will count every directional change that is larger than 20 degrees as a "Turn". For counting Junctions, add another attribute, and in Default Values, click "Junction", Default=1 This will count every junction as 1, allowing the network evaluators to keep track of how many junctions have been crossed on any of the calculated paths.

7. Search Radius

The *Search Radius* input defines the buffer radius used for computing the specified measures. For each Input Building, only other points whose shortest network distance from the given Input Point is less than the *Search Radius,* are considered in the analysis. If no Search Radius is specified by the user, then the default infinite radius is used to reach all parts of the graph. Note that the units for SearchRadius are the same as the Impedance Attribute units of your Network Dataset: if your network Impedance Attribute is in meters, then the Search Radius is in meters etc.

[8] In this case a *Space Syntax Integration* type analysis can be performed using the Reach and Closeness metrics.

8. Beta-value

β is used only in the gravity type index computation. It controls the shape of the distance decay curve in the denominator of the index. The inverse effect of distance specified in the Gravity index decreases exponentially. β and the corresponding shape of distance decay should be derived from the assumed mode of travel - for walking accessibility in “minutes”, for instance, researchers have found β to fall around 0.1813 (Handy and Niemeier 1997). The equivalent value of Beta for impedance units in “meters” is 0.00217; in “feet” 0.000663; in “kilometers” 2.175, and in “miles” 3.501. β must range between “0” and “1”. If no input for beta is specified by the user, then the default entry of “0” is used automatically.

9. Accumulators

Accumulators allow the different travel cost attributes available in the Network Dataset to illustrate the total sum of the attribute used as while reaching all the surrounding destination buildings that fall within the given Search Radius. Accumulating the „Length“ attribute, for instance, will sum up all indiviual path shortest path lengths required to reach the surrounding destinations. Accumulating a „Turns“ attribute, on the other hand, will instead sum up the total number of turns required to reach the same set of destinations. Such accumulated values may be useful as control factors in some statistical analyses.

10. Normalize Results

If normalizations are asserted to any of the specified measures, then the measures will be normalized by the total number of neighboring weights around each building within the given *Search Radius*. If no Search Radius is used, the results are normalized by the weights in all Input Buildings in the graph. For *Reach*, *Gravity*, *Betweenness* and *Straightness* measures, normalized results will always produce resulting values that fall between zero and one. Note that the nature of the *Closeness* measure, however, does not guarantee normalized values between zero and one.

Measure	Definition	Normalization
Reach	$Reach[i]^r = \sum_{j \in G-\{i\}, d[i,j] \leq r} W[j]$	$Reach[i]^r{}_{norm} = \frac{Reach[i]^r}{\sum_{j \in G-\{i\}} W[j]}$
Gravity	$Gravity[i]^r = \sum_{j \in G-\{i\}, d[i,j] \leq r} \frac{W[j]}{e^{\beta \cdot d[i,j]}}$	$Gravity[i]^r{}_{norm} = e^{\beta} \cdot \frac{Gravity[i]^r}{Reach[i]^r}$
Betweenness	$Betweenness[i]^r = \sum_{j,k \in G-\{i\}, d[j,k] \leq r} \frac{n_{jk}[i]}{n_{jk}} \cdot W[j]$	$Betweenness[i]^r{}_{norm} = \frac{Betweenness[i]^r}{(\|\{j\, G-\{i\} : d[i,j] r\}\| - 1) \cdot Reach[i]^r}$
Closeness	$Closeness[i]^r = \frac{1}{\sum_{j \in G-\{i\}, d[i,j] \leq r} (d[i,j] \cdot W[j])}$	$Closeness[i]^r{}_{norm} = Closeness[i]^r \cdot Reach[i]^r$
Straightness	$Straightness[i]^r = \sum_{j \in G-\{i\}, d[i,j] \leq r} \frac{\delta[i,j]}{d[i,j]} \cdot W[j]$	$Straightness[i]^r{}_{norm} = \frac{Straightness[i]^r}{Reach[i]^r}$

Normalization can be useful, for instance, when the number of neighbors around each building within a given *SearchRadius* varies widely. Whereas the usual *Betweenness* measure returns the total number of trips passing by each Input Building, the normalized *Betwenness* measure returns instead the *percentage* of total trips passing by each Input Building. Since the Reach measure is normalized by the maximum possible value of Reach at an infinite Search Radius in the graph, then the normalized Reach results tell us what percentage of all Input Buildings or weights (if the analysis is weighted) are reached from each Input Building at a given Search Radius.

11. TOOLBOX OUTPUTS

11.1 Output Location

This input allows the user to choose the folder in which all the result files will be written.

11.2 Output File Name

This input designates the name that will be given to the DBF file that contains the results. The DBF results file, and the results layer of the same name, will contain an attribute column showing the Building ID Attribute Name, and one or more columns showing the analysis results. The results column titled *R, G, B, C, S* refer to *Reach, Gravity, Betweenness, Closeness,* and *Straightness* accordingly.

12. Graph Adjacency Matrix

The UNA toolbox generates an intermediary Adjacency Matrix file in order to produce the analysis. This Adjacency Matrix is stored in the *Output Location* directory with a “Bldng_Adj...” file name beginning. The remaining part of the file name is obtained from the *Input Buildings* and *Input Analysis Network*. A new Adjacency Matrix is created each time new points or street network are used as inputs.

The Adjacency Graph represents immediate neighbor relationships between a building i and its immediately closest neighboring buildings on all available circulation routes that pass thorough *i*. The Adjacency Graph construction is demonstrated in Figure 5, where the small red buildings become neighbors to the larger red node i in the constructed Adjacency graph. The remaining buildings within the given Search Radius r are not considered to be i's adjacent neighbors, since there is at least one other building between them and i.

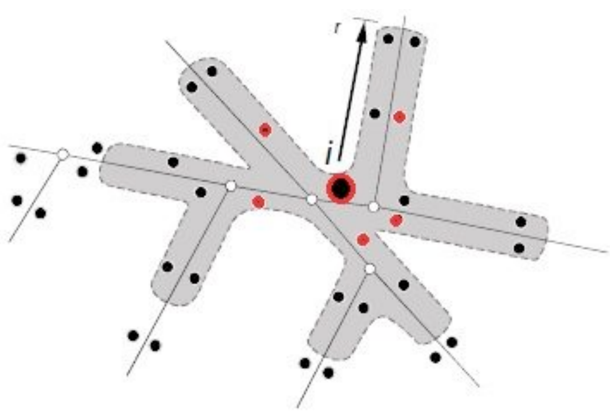


Figure 5: Illustration of adjacency matrix construction.

13. REFERENCES

Bhat, C., Handy, S., Kockelman, K., Mahmassani, H., Weston, L., Gopal, A., et al. (2002). Development of an Urban Accessibility Index. (NSF, Eds.). Austin, TX: Center for Transportation Research. The University of Texas at Austin.

Brandes, U. (2001). A faster Algorithm for Betweenness Bentrality. Journal of Mathematical Sociology, Volume: 25(Issue: 2), Pages: 163-177.

Handy, S., & Niemeier, A. D. (1997). Measuring Accessibility: an exploration of issues and alternatives. *Environment and Planning A*, *29*, 1175-1194.

Hansen, W. G. (1959). How Accessibility Shapes Land Use. *Journal of the American Planning Association*, *25*(2), 73-76.

Hillier, B., & Hanson, J. (1984). The Social Logic of Space. Cambridge: Cambridge University Press.

March, L., & Steadman, P. (1971). The geometry of environment: an introduction to spatial organization in design (p. 360 p.). London, RIBA Publications.

Porta, S., Crucitti, P., & Latora, V. (2005). The network analysis of urban streets: a primal approach. Environment and Planning B, 35(5), 705-725.

Sabidussi, G. (1966). "The centrality index of a graph." Psychmetrika 31: 581-603.

Sevtsuk, A. (2010). Path and Place: A Study of Urban Geometry and Retail Activity in Cambridge and Somerville, MA. PhD Dissertation. Cambridge: MIT.

Vragovic, I., Louis, E., & Diaz-Guilera, A. (2005). Efficiency of information transfer in regular and complex networks. Physics Review E., 71(026122).