Urban Network Analysis

Toolbox for ArcGIS 10 / 10.1 / 10.2

HELP

Version 1.01
September 10, 2013

City Form Lab

Better design trhough research | cityform.mit.edu

Andres Sevtsuk; Michael Mekonnen, Raul Kalvo
Contact:
cityformlab@ mit.edu

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

Cover Image: Jason Hawkes

CONTENTS

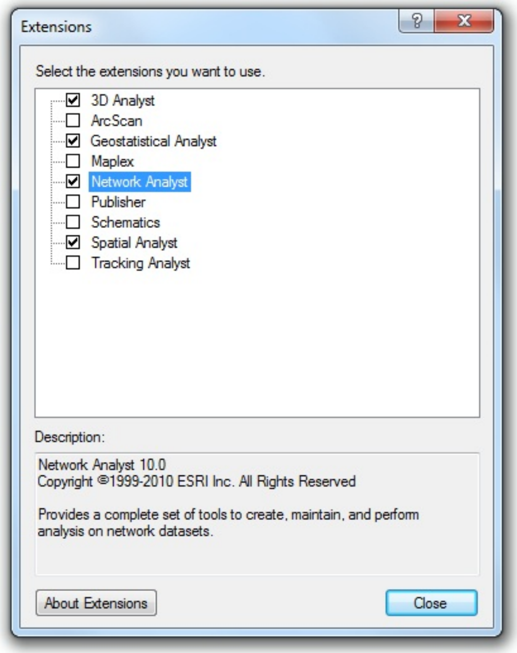
INSTALLATION GUIDELINES 3
1. CENTRALITY TOOL 6
1.1. Description 6
1.2. Toolbox Inputs 7
1.3. Choice Of Analysis 10
1.4. Building Id Attribute Name 14
1.5. Building Weights Field 14
1.6. Impedance Attribute 15
1.7. Search Radius 15
1.8. Radius Type 16
1.9. Beta-Value 18
1.10. Accumulators 18
1.11. Normalize Results 18
1.12. Toolbox Outputs 19
1.13. Graph Adjacency Matrix 20
2. REDUNDANCY TOOLS 22
2.1. Description 22
2.2. Redundancy Index and Redundant Paths 22
2.3. Inputs 24
14. TROUBLESHOOTING 27
15. REFERENCES 28

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 or 10.1 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.



Select the extensions you want to use.

☑ 3D Analyst
☐ ArcScan
☑ Geostatistical Analyst
☐ Maplex
☑ Network Analyst
☐ Publisher
☐ Schematics
☑ Spatial Analyst
☐ Tracking Analyst

Description:

Network Analyst 10.0
Copyright ©1999-2010 ESRI Inc. All Rights Reserved

Provides a complete set of tools to create, maintain, and perform analysis on network datasets.

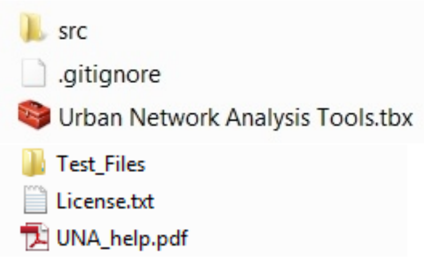
Figure 1. The UNA toolbox requires the Network Analyst Extension to be enabled.

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 or later. The original release of ArcGIS 10 has a known bug that prevents script results from adding to the display. We have updated the toolbox (September, 2012) to also work in version 10.1.

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 (*Urban Network Analysis Tools.tbx*), a license file (*License.txt*) help documentation (UNA_help.pdf), Test_Files, and one source code subfolder (src). The Urban Network Analysis is distributed and can be used under the Creative Commons Attribution – Noncommercial – Share Alike 3.0 Unported License. For more information, see: http://creativecommons.org/licenses/by-nc-sa/3.0/

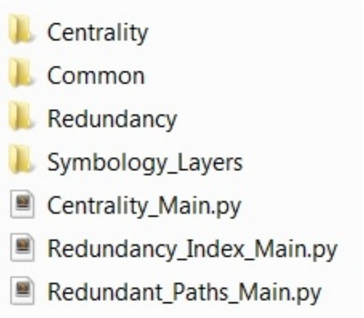
The Test_Files folder contains a few sample network datasets and point shapefiles that you can use to test the tools. You can delete this folder if you are not interested in using the test files.



src
.gitignore
Urban Network Analysis Tools.tbx
Test_Files
License.txt
UNA_help.pdf

Figure 2. Contents of the *UrbanNetworkAnalysis.zip* folder.

The source folder (src) contains several subfolders – Centrality, Common, Redundancy, and Symbology_Layers and python code files (with .py extensions). This source folder stores all the scripts that are needed for the UNA tools to work, while the Symbology Layers contain symbology templates that ArcGIS needs for automatically returning your graphic results on the screen.

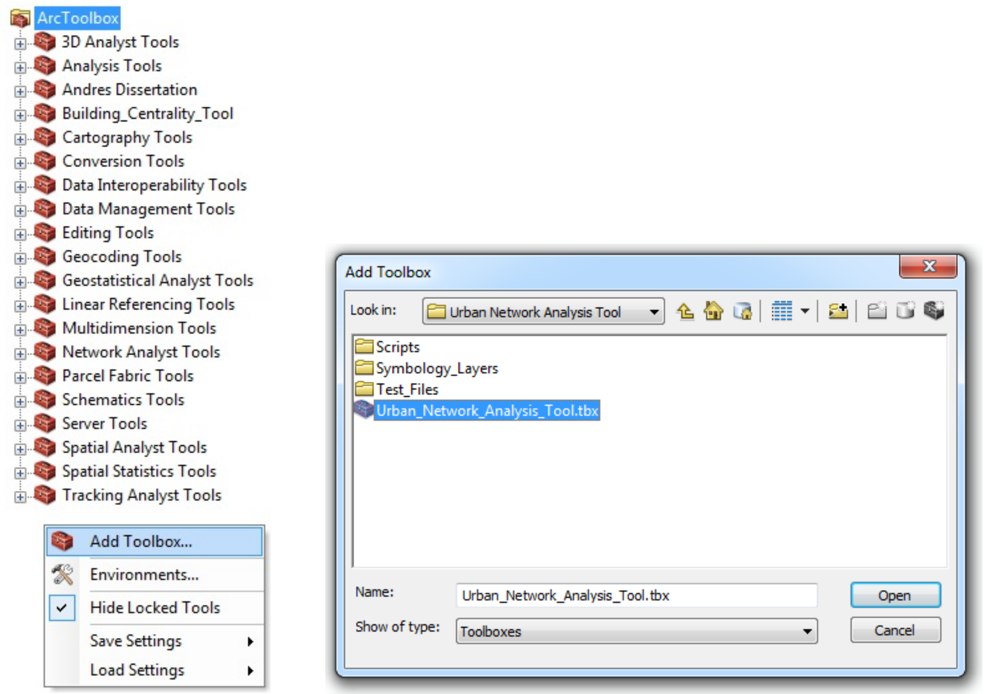


Centrality
Common
Redundancy
Symbology_Layers
Centrality_Main.py
Redundancy_Index_Main.py
Redundant_Paths_Main.py

Figure 3. Contents of the src folder.

Installation

To install the toolbox, open ArcMap 10 or 10.1, 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.



Look in: Urban Network Analysis Tool

Name: Urban_Network_Analysis_Tool.tbx

Show of type: Toolboxes

Figure 4. Adding the UNA toolbox to the ArcGIS tools tab.

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

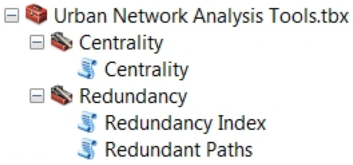


Figure 5. Urban Network Analysis Tools.

You can have ArcGIS load this toolbox by default along with other toolboxes every time you open the program by right-clicking on a white area inside the toolbox tab and choosing *Save Settings → To Default*.

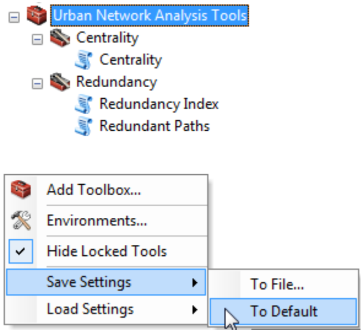


Figure 6. Saving the UNA tools to default settings to be loaded by ArcGIS every time the software is started.

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). These files can be used as Input Buildings and Input Analysis Networks respectively in order to test the toolbox functions.

License

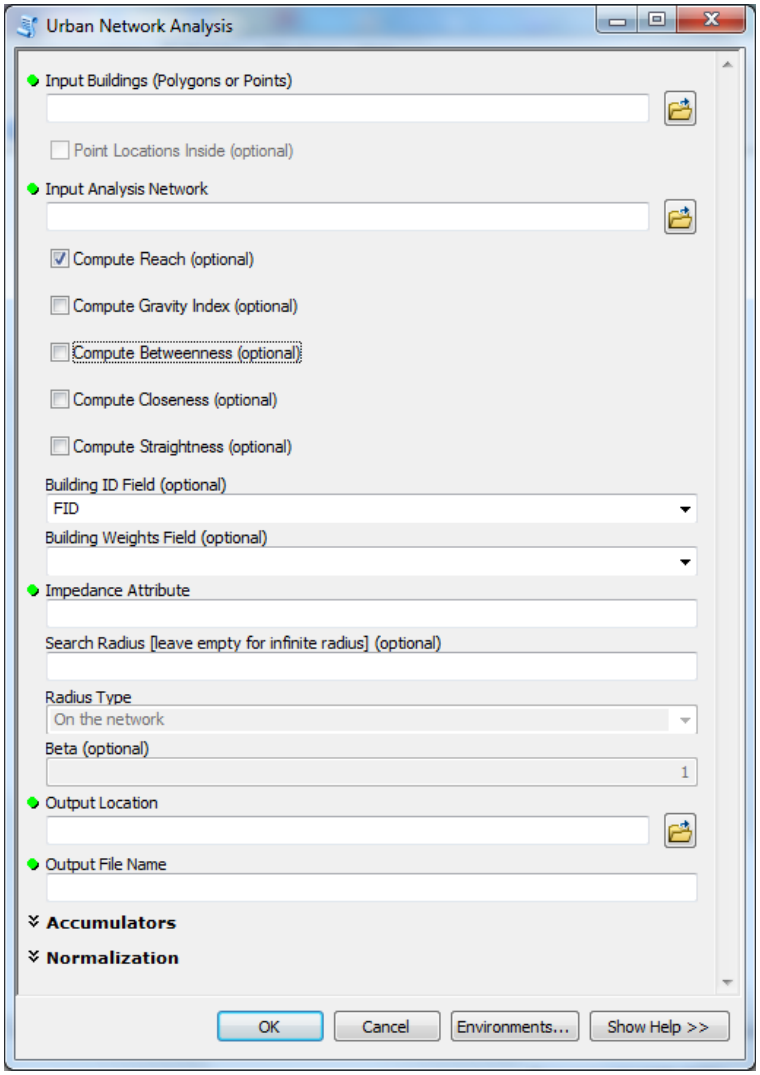
1. CENTRALITY TOOL

1.1. Description

This Centrality toolbox (Sevtsuk, Mekonnen 2012) 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; Harary 1969; 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 type of network elements - *buildings* - that 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 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.

[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.



Urban Network Analysis

● Input Buildings (Polygons or Points)

□ Point Locations Inside (optional)

● Input Analysis Network

☑ Compute Reach (optional)

□ Compute Gravity Index (optional)

□ Compute Betweenness (optional)

□ Compute Closeness (optional)

□ Compute Straightness (optional)

Building ID Field (optional)
FID

Building Weights Field (optional)

● Impedance Attribute

Search Radius [leave empty for infinite radius] (optional)

Radius Type
On the network

Beta (optional)
1

● Output Location

● Output File Name

Figure 7: Graphic User Interface

1.2. Toolbox Inputs

1.2.1. 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, please note that an ArcInfo license is required

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

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.

Note that the Centrality tool can only take Input Points from featureclasses (usual shapefiles), but not from geodatabases (gdb). The Redundancy tools don't have this restriction.

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.

1.2.1.1. 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 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 origins and destinations. If a selection contains only subway stations, and the analysis is weighted by the number of office workers in each buildings, 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 Betweenness 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.

1.2.2. Network Dataset

Network Dataset is 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 versions 10 and later make it particularly easy to convert a polyline layer to a Network Dataset. Simply open the ArcCatalogue Tree in ArcMap (Figure 8), navigate to the polyline file that contains the appropriate network file, right-click, choose *New Network Dataset*, and follow the instructions (Figure 9). Before running the UNA tools, ensure visually that the Input Points are on the right locations on the Network Dataset.

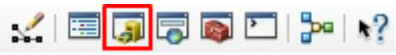


Figure 8: Opening the Arc Catalogue Tree in Arc Map 10

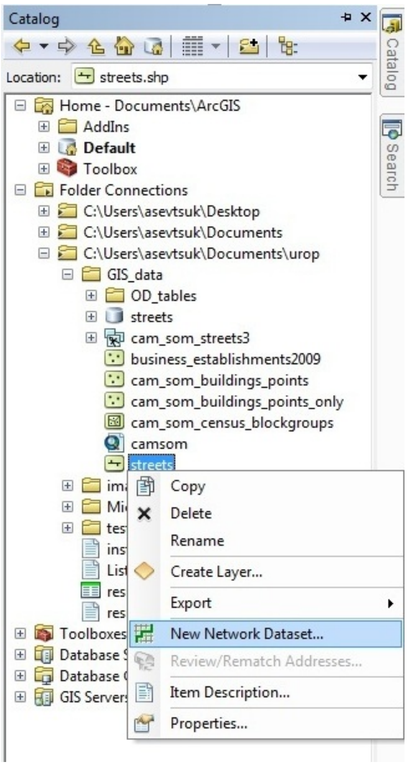


Figure 9: Creating a New Network Dataset from a polyline shapefile in ArcMap 10.

1.3. 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, Betweenness, Closeness*, and *Straightness*. These measures are described below.

1.3.1. 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 10 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.

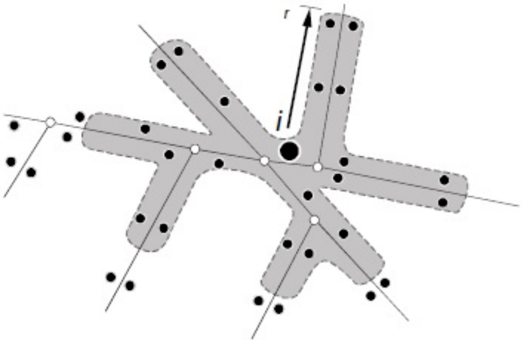


Figure 10: Illustration of the Reach measure.

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*.

1.3.2. 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 (*s*pecified 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, then the default entry of "0" is used automatically, and no distance decay is applied.

1.3.3. 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*.* *Betweenness* for building i is computed by considering all pairs of buildings j, k that are within a distance r from each other. It is not computed by considering all pairs of buildings j, k that are within a distance r from i. This is because we do not consider any trips between two buildings that are more than r apart. If we know that a pair of buildings j, k are within r from each other, and that a shortest path from j to k (or k to j) passes by building i, then both j and k are also certainly within a distance r from i.

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).

1.3.4. Closeness

The *Closeness* of an *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:

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

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.

1.3.5. 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 (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.

1.4. 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.

1.5. 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.

1.6. 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. Right-Click the ND file → 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.

1.7. 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 the user specifies no Search Radius, then the default infinite radius is used to reach all parts of the graph. Note that

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

the units for Search Radius 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.

1.8. Radius Type

The Radius Type variable enables you to choose whether the neighboring buildings or points that are chosen for the analysis are within a specified network distance or Euclidian distance from each unit of analysis. Note that this option only affects which neighbors are selected for the analysis; it has no effect on the actual route computations in the analysis. All trips between observations are still computed using network routes, but the Radius Type option additionally allows you to consider neighbors that may fall beyond the network radius. Since the geometry and topology of the network often differs around each building, this option can be used to equalize the spatial neighbor selections around each building.

The option is only available if you choose to use a radius. As with the default Network Radius, the Euclidian Radius assumes the center of the radius to be the network snap location of the origin building. A circular radius *r* is drawn from that location and all other buildings, whose network snap locations fall within the circle are chosen as neighbors for analysis (Figure 11).

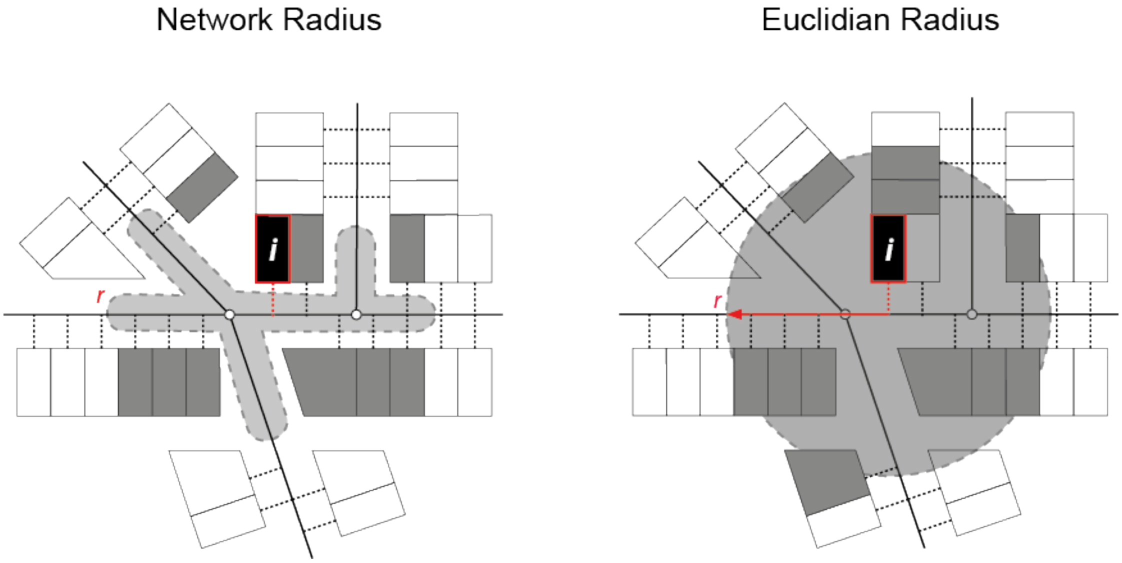


Figure 11: Network Radius Type (left) and Euclidian Radius Type (right).

Where is this useful? Consider a hypothetical situation for measuring *Straightness,* pointed out to us by Patrick, one of our users, in Figure 6. Building *i* in Figure 6 has four neighboring buildings around it, but one of them – *j* – does not have a street connection to *i* within the specified Network Radius. If we estimate *Straightness* using the Network Radius, then building *j* will not affect *i*'s *Straightness* result as it cannot be reached in the analysis (Figure 12, left). But imagine that a new street segment is built, which establishes a shorter connection from *i* to *j* (Figure 6, right). If we re-estimate Straightness for building *i* in this new configuration using the same Network Radius, we shall observe that the *Straightness* value for *i* decreases. This is because the new network path from *i* to *j* is not as straight as the paths to other neighbors. Paradoxically, increasing connectivity in the environment can decrease a building's *Straightness* value.

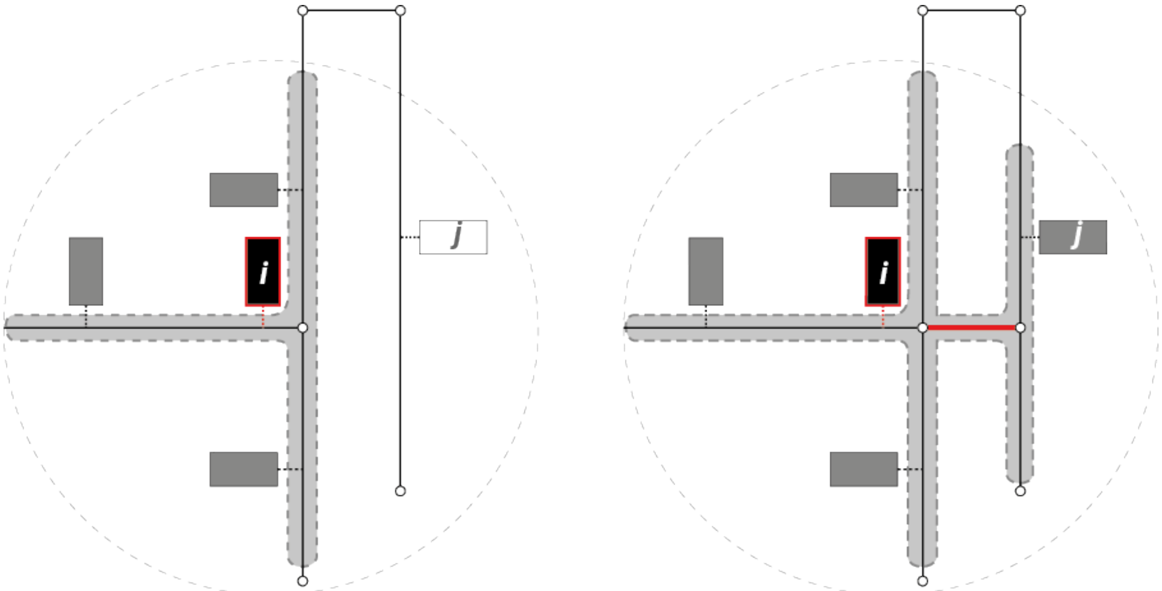


Figure 12: Estimating Straightness using Network Radius (left) and Euclidian Radius (right).

Even though technically correct, this paradox is overcome if we use an Euclidian Radius instead. Drawing an Euclidian Radius around *i* will include j in the Straightness estimation regardless of whether *j* has a network path to *i* that is shorter than the Search Radius, or not. In the denominator of the *Straightness* index we then use the shortest network distance between *i* and *j* that is available in the entire network, regardless of the specified Search Radius. [9] This Network Radius can largely exceed the Euclidian radius, there is no limit to it, as long as a connection between *i* and *j* is found in the whole network. But of course, the longer the route, the less *j* contributes to *i*'s *Straightness*. Adding a shorter network connection between *i* and *j* in Figure 6 (right) therefore increases *i*'s *Straightness* value, as one would intuitively expect. Using

[9] As a result, the index satisfies three conditions: 1) The existence of any building in the Euclidian buffer increases *i*'s Straightness value even if there is no network route between them within the same radius; 2) any neighboring building around i that has a network connection to i contributes more to *i*'s Straightness value than a building that does not have a network path to *i* within the given radius; 3) shorter network connections between *i* and j yield a greater Straightness value for *i* than longer network connections between *i* and *j*.

the *Euclidian Radius* thus allows us to evaluate how changes in street layouts affect accessibility values, while keeping the number and location of buildings constant.

1.9. 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 the user specifies no input for beta, then the default entry of “0” is used automatically.

1.10. 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 individual 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.

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

Table 1. Normalization of UNA output values.

Normalization can be useful, for instance, when the number of neighbors around each building within a given *Search Radius* varies widely. Whereas the usual *Betweenness* measure returns the total number of trips passing by each Input Building, the normalized *Betweenness* 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.

1.12. Toolbox Outputs

1.12.1. Output Location

This input allows the user to choose the folder in which all the result files will be written. Please note that the maximum length for feature classes and tables in ArcGIS is 160 characters, including the path name. Please be aware of this limit when choosing your path and Output File Name and shorten them if necessary.

1.12.2. 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.

Please note that the maximum length for feature classes and tables in ArcGIS is 160 characters, including the path name. Please be aware of this limit when choosing your path and Output File Name and shorten them if necessary. Event though this limitation is unlikely to affect most input file names, please keep in mind that it also affects the Adjacency Matrix name that the toolbox automatically generates. The name of the Adjacency Matrix is composed as follows:

„Adj _ [name of input buildings] _ [name of input network] _ [name of ID attribute] _ [name of impedance attribute] _ [names of accumulator attributes].dbf

As this dbf filename is composed of a number of components, it can indeed reach the 160-character limit if your input filenames are long. The UNA toolbox needs to keep the Adjacency Matrix name so long in order for it to be re-usable – the different components in the name each affect the matrix and we only re-use the matrix if all of its ingredients are identical.

1.13. Graph Adjacency Matrix

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 13, 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.

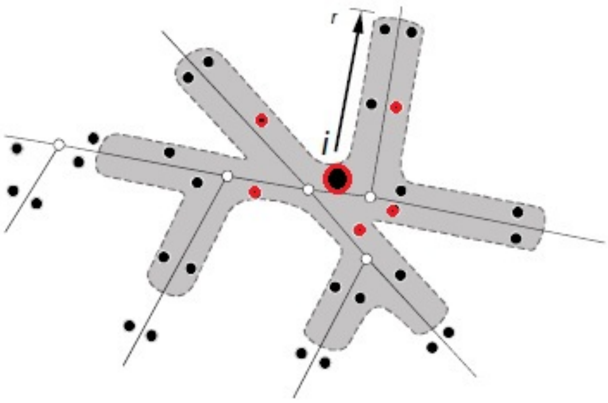


Figure 13: Illustration of adjacency matrix construction.

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 "Adj_..." file name beginning. The remaining part of the file name is obtained from the Input Buildings, the Input Analysis Network, the Name of the ID attribute, the name of the Impedance Attribute and the names of Accumulator Attributes. The name of the Adjacency Matrix is composed as follows:

„Adj _ [name of input buildings] _ [name of input network] _ [name of ID attribute] _ [name of impedance attribute] _ [names of accumulator attributes].dbf

A new Adjacency Matrix is created each time new Input Buildings / Points or a new Street Network are used as inputs. If the same inputs are used in two or more consecutive runs of the toolbox, then the Adjacency Matrix is automatically re-used. Since the computation of the Adjacency Matrix is quite costly, this can substantially shorten calculation time. Whether or note the matrix is re-used depends on the user's specification of toolbox inputs, which in turn determine the above name of the matrix. If all parts of the name remain identical, the matrix is re-used. If any part of the name changes due to changed inputs, the matrix is recalculated.

Note that the above convention requires that the user be attentive to any geometric or topological changes in the network itself. If the user alters the Network Dataset, but keeps the name identical, then the toolbox will knot know of the changes and will attempt to re-use the pre-existing matrix from a previous run. We therefore recommend that users keep close track of any geometric or physical alterations to the Network Dataset and save the ND file with a different name each time its shape or connectivity are changed.

2. REDUNDANCY TOOLS

2.1. Description

Much of spatial network analysis assumes that travel occurs along the shortest available routes. This is not necessarily the case in reality – people often use paths that are slightly longer than the shortest path in order to pass by other locations on the way, to experience a more scenic or interesting route or to navigate a simpler path. Pedestrian route choice surveys have found that people commonly deviate up to 20% longer than the shortest path. The Redundancy tools allow us to detect these alternative paths. The analysis finds alternative paths that are available between an origin and destination within a given distance quota beyond the shortest path and optionally displays the segments or routes that participate in any of the redundant paths.

2.2. Redundancy Index and Redundant Paths

The UNA toolbox offers two separate Redundancy tools – the *Redundancy Index* tool and the *Redundant Paths* tool. *Redundancy Index* computes the ratio of the sum of the lengths of redundant segments to the sum of the lengths of shortest path segments per each O-D pair. If there are multiple destinations for the same origin, we average these multiple values. Given a pair of nodes O and D in a positively weighted, undirected graph G, the redundancy index for the pair with a redundancy ratio $\rho \geq 1$ is defined by:

$$\mathrm{R}^{\rho}[\mathrm{O},\mathrm{D}] = \frac{\sum_{\mathrm{e}\in\mathrm{E}} \mathrm{w}[\mathrm{e}]\cdot\zeta^{\mathrm{O},\mathrm{D}}[\mathrm{e},\rho\cdot\mathrm{d}[\mathrm{O},\mathrm{D}]]}{\sum_{\mathrm{e}\in\mathrm{E}} \mathrm{w}[\mathrm{e}]\cdot\zeta^{\mathrm{O},\mathrm{D}}[\mathrm{e}\cdot\mathrm{d}[\mathrm{O},\mathrm{D}]]},$$

where $\mathrm{w}[\mathrm{e}]$ is the weight of edge e in G; $\mathrm{d}[\mathrm{O},\mathrm{D}]$ is the shortest path distance from $\mathrm{O}$to D in G; and $\zeta^{\mathrm{O},\mathrm{D}}[\mathrm{e},\mathrm{x}]$is 1 if there exists a *simple* path from O to D in G that goes through edge e and whose weight is at most x, and 0 otherwise. A *simple* path is a path where no nodes are repeated. The enumerator of the index sums the lengths of all redundant path segments between the origin and destination; the denominator sums the lengths of the shortest path segments between the same origin and destination.

The *Redundant Paths* tool computes exactly this value. However, the *Redundancy Index* tool relaxes the *simple* path restriction of the value $\zeta^{\mathrm{O},\mathrm{D}}[\mathrm{e},\mathrm{x}]$ and allows nodes to be used more than once on the same path. In the Redundancy Index tool, $\zeta^{\mathrm{O},\mathrm{D}}[\mathrm{e},\mathrm{x}]$ is 1 if there exists *any* path (not necessarily *simple*) from O to D in G that goes through edge e and whose weight is at most x.

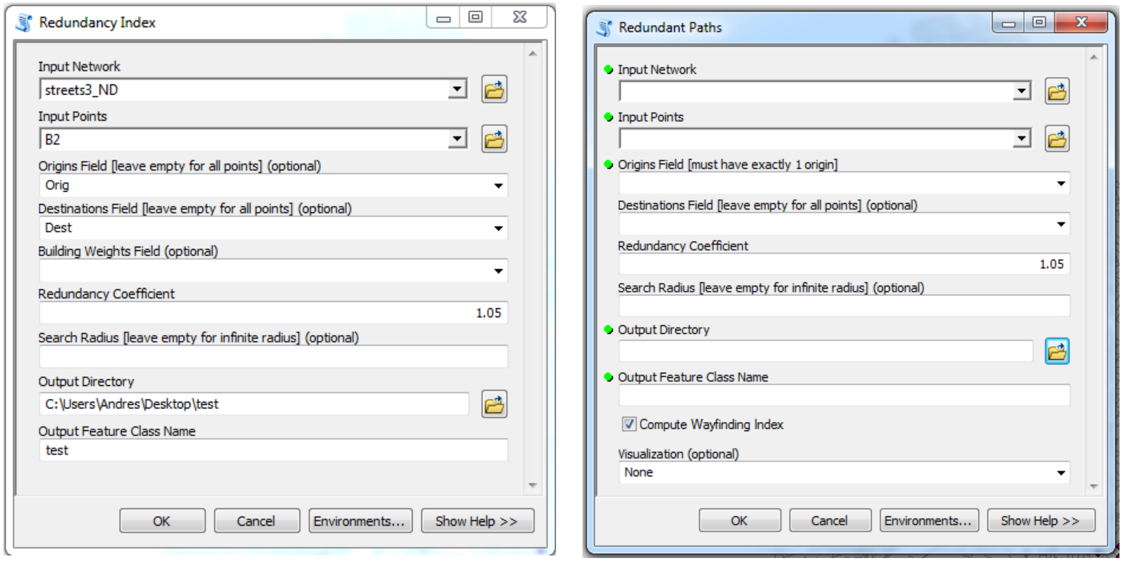
If more than one destination are included, then the redundancy values to individual destinations are averaged. Given one origin O and a non-empty set of destinations Δ (all within a given search radius from the origin O), we compute the average redundancy index for O with a redundancy ratio $\rho \geq 1$as:

$$R^{\rho}[O,\Delta] = \frac{\sum_{D\in\Delta} R^{\rho}[O,D]}{|\Delta|},$$

where |Δ| is the size of the set Δ, i.e. the number of destinations.

The Redundant Paths tool can additionally return all the individual paths that are found between the given origin and destination (given as polylines) within the given distance quota. The Redundant Routes tool uses a more calculation intensive *Depth First Search* to return all simple paths that fit the distance quota. The number of alternative paths that satisfy the constraints may be very large between points that lie on a highly connected network, relatively far from each other, or when the distance quota for search is large. Two locations that are a kilometer apart on the Manhattan grid, for instance, may have hundreds of thousands of unique route alternatives between them within a +20% distance quota. The Redundant Paths tool may very well take a very long time as the problem it tires to solve is very difficult by nature. The Redundant Paths tool is therefore restricted to only one origin and one or more destinations at a time.

The Redundancy Index, on the other hand, is much faster and can be calculated for a number of origins and a number of destinations at a time. Despite being an approximation (where routes are not necessary *simple* paths), the Redundancy Index is usually close to Redundant Paths and can be run on much larger datasets. The index tells us how many times more linear street length becomes available from an origin to a destination if the walk is extended by a given percentage above the shortest path. If extending walks by 10%, for instance, yields a result of "Redundancy = 5.5", then this means that 5.5 times more street length becomes available on redundant paths compared to the shortest path alone.



Redundancy Index

Input Network: streets3_ND

Input Points: B2

Origins Field [leave empty for all points] (optional): Orig

Destinations Field [leave empty for all points] (optional): Dest

Building Weights Field (optional):

Redundancy Coefficient: 1.05

Search Radius [leave empty for infinite radius] (optional):

Output Directory: C:\Users\Andres\Desktop\test

Output Feature Class Name: test

Redundant Paths

● Input Network:

● Input Points:

● Origins Field [must have exactly 1 origin]:

Destinations Field [leave empty for all points] (optional):

Redundancy Coefficient: 1.05

Search Radius [leave empty for infinite radius] (optional):

● Output Directory:

● Output Feature Class Name:

☑ Compute Wayfinding Index

Visualization (optional): None

Figure 14. Left: Redundancy Index tool user interface. Right: Redundant Paths tool user interface.

2.3. Inputs

2.3.1 Input Network

Network Dataset is the path network in which the analysis will be carried out. 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 versions 10 and later make it particularly easy to convert a polyline layer to a Network Dataset. Simply open the ArcCatalogue Tree in ArcMap (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.

2.3.2. Input Points

The point shapefile that defines the points on the network to be used for Redundancy analysis. The attribute table of the Input Points shapefile is also used for defining which points are used as origins and destinations in a following step. Note that Redundancy analysis, especially Redundant Paths, can be computationally very expensive and large datasets can be difficult to analyze at once.

Note that unlike the Centrality tool, the Redundancy Tools can take Input Points from either featureclass or geodatabase (gdb) shapefiles.

In order to compute Redundancy measures for the nodes of the network instead of buildings, then nodes illustrating street intersections can also be used as inputs.

2.3.3. Origins Field [leave empty for all points]

This field, part of the Input Points shapefile, designates which points are used as origins in Redundancy analysis. The chosen field has to be numeric (integer or float). Those point features that have a positive value (greater than zero) in the Origins Field are set as origins; those that have a zero or Null will not be included as origins. The field can be left empty to include all points as origins.

If it doesn’t pre-exist, then the field can be created in the Input Points shapefile by opening the attributes table and choosing “Add New Field” and adding it a name (e.g. “Origins”). Graphic selections or selections by attributes can be used in ArcMap to identify points that should be included in the set of origins. While the selected points are active, the Calculate Values function can be called by right-clicking on the new field name in the attribute table, assigning a positive value to those points that active in the selection.

2.3.4. Destinations Field [leave empty for all points]

This field, part of the Input Points shapefile, designates which points are used as destinations in Redundancy analysis. The chosen field has to be numeric (integer or float). Those point features that have a positive value (greater than zero) in the Destinations Field are set as destinations; those that have a zero or Null will not be included as destinations

2.3.5. Building Weights Field

Redundancy analysis typically estimates the ratio of redundant path distances over the shortest path distance. Using weights allows the analysis to alter the ratio from comparing distances to comparing some other attributes of the paths. If the chosen weights describe the number of retailers in each input point, for instance, then the Redundancy Index will return the ratio between the number of retailers found on all redundant paths over the number of retailers on found on the shortest path. The weighted index describes how many more weights can be encountered on redundant paths that are up to a given percent longer than the shorted path.

Note that the Weights Field option is not offered in the Redundant Paths tool, since the aim of the latter is to simply return the redundant routes as polylines.

2.3.6. Redundancy Coefficient

This input defines how much longer redundant paths may be compared to the shortest path. The coefficient is given as a length multiplier – a coefficient of 1.05 allows redundant paths to be up to 5% longer than the shortest path; a coefficient of 1.25 allows redundant paths to be up to 25% longer and so on. Note that the higher the coefficient, the more paths are typically found and the more computation intensive the analysis becomes.

2.3.7. Search Radius [leave empty for infinite radius]

The Search Radius defines how far along the network the destinations may be from an origin in order to be included in the analysis. Even though the some of origins and destinations indicated in the Origins Field and Destinations Field may be far apart from each other, the Search Radius allows us to filter in only those O-D pairs for calculations that are within a chosen network radius from each other. A 600-meter search radius, for instance, will limit the redundancy analysis to only those destinations that are up to 600m from the origins. Note that the units of Search Radius are defined by the units of the Network Dataset (ND). If the ND is defined in meters, then the Search Radius will also work in meters.

2.3.8. Output Directory

Defines the folder where to save the analysis results.

2.3.9. Output Featureclass Name

Defines the name of the output shapefile.

2.3.10. Compute Wayfinding Index (only in the Redundant Paths tool)

The Redundant Paths tool additional allows the user to compute the Wayfinding Index for each origin from all surrounding destinations. The Wayfinding index for a pair of nodes O and D is the probability of reaching D starting from O and making uniformly random decisions at every intersection, within a given distance quota above the shortest path and with the restriction that all traveled paths must be simple (no repeating nodes). Figure 15 below illustrates the how the index is computed graphically. The chances of reaching D from O with random walks and within a 1.2 distance coefficient in the image is 7 out of 72, or 9.72%.

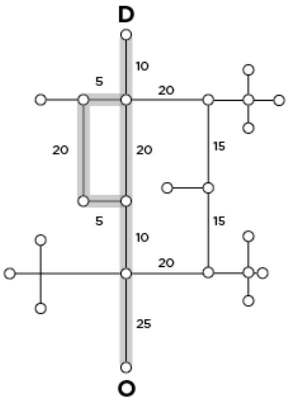


Figure 15. Wayfinding Index computes the probability of reaching a destination from an origin while taking random choices at all intersections and limiting the search distance to the given Redundancy Coefficient and allowing only simple paths. The probability of reaching D from O within a Redundancy Coefficient of 1.2 in the image is 1/3 * 1/2 *1/3 + 1/3 * 1/2 * 1/2 * 1/2 = 7/72, which is 9.72%.

The Wayfinding Index is only defined for individual O-D pairs; it is not computed from one origin to many destinations. The resulting values are given in the output table of the point featureclass that is returned at the end of the analysis.

2.3.11. Visualization (only in the Redundant Paths tool)

The visualization option in the Redundant Paths tool allows one to choose whether and how to visualize the network edges that were found in the set of redundant paths. Since the number of paths can grow very large, the default option is kept at no visualization. Choosing *Visualize Paths* from the dropdown menu will return all redundant paths as polylines. If 10,000 paths were found, 10,000 polylines are returned. Visualizing a high number of partially overlapping polylines can get computationally heavy for ArcGIS.

It is less costly to choose to *Visualize Segments,* which returns only the network segments that participate in the set of redundant paths. Each segment is only represented once.

3. TROUBLESHOOTING

Though we continuously keep improving the UNA Toolbox code, it is not entirely perfect and errors may appear. We have created a Google Group user Forum for collecting your feedback and discussing the problems you might encounter. Please look through the existing threads on the Forum to see if you find an answer to your question and send us any new questions that you might encounter.

http://groups.google.com/group/urban-network-analysis

We thank you for using the toolbox and hope you enjoy it.

City Form Lab team.

4. 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.

Harary, F. (1969). Graph Theory. Massachusetts: Addison-Wesley.

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.

Sevtsuk, A., Mekonnen, M. (2012). “Urban Network Analysis Toolbox“, International Journal of Geomatics and Spatial Analysis, vol. 22, no. 2, pp. 287-305, 2012.

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