

Urban Network Analysis Toolbox for Rhinoceros 3D

3. Graphic user interface



GUI는 편집, 임포트/익스포트, 네트워크 셋업, 분석, 그리고 그래픽 세팅의 다섯 개 카테고리로 나뉘어져 있다.

처음 7개의 아이콘은 개체특성에 관한 것이며 이것은 분석하는데 이용된다.

UNA Toolbox에서 "attributes" 항목은 대상에 이름, 숫자, 무게와 같은 특성을 부여한다. 이러한 attributes는 GIS의 쉐이프 파일과 유사하다. 하나의 대상은 다양한 특성을 가질 수 있다. 그러나 GIS와는 달리 테이블로 분류되어지지 않지만 사전으로는 분류될 수 있다.
Attributes는 숫자 혹은 텍스트 기반 모두 될 수 있다.

3.1 Add Text Attribute to Objects

Attributes는 공간의 개체에 현실세계와 같은 특성을 부여하여 구별할 수 있게 만든다. 예를 들면 라이노에서 Text Attribute를 개체에 배정하는 것은 비즈니스 이름을 넣거나 청결도를 A, B, C 등의 등급으로 나눌 수 도 있다. Text Attributes에는 분석에 도움되는 어떠한 항목도 쓸 수 있다. 단지 웨이트로는 사용할 수 없다.

툴은 한번에 한 개 혹은 여러 개의 개체를 선택할 수 있다. 선택하면 커맨드 라인에 name과 text가 나타난다.

"Attribute Name"은 사전적 키이며 테이블에서 항목과 비슷하다. "Text Value"는 Attribute에 대응하는 값이다.

3.2 Add Numeric Attribute to Objects

이 아이콘은 개체에 수량의 특성을 부여할 수 있다. 각 개체의 값에 따라서 분석하기 때문에 특히 공간네트워크 분석에 유용하다. 예를 들어 "Numeric Attribute"는 빌딩의 층수를, 공간의 넓이를, 빌딩의 사람수 등을 넣을 수 있다. 분석 시에 "weight"로 사용될 수 있다. 툴은 한번에 한 개 혹은 여러 개의 개체를 선택할 수 있다. 선택하면 커맨드 라인에 name과 weight가 나타난다.
"Attribute Name"은 사전적 키이며 테이블에서 항목과 비슷하다. "Text Value"는 Attribute에 대응하는 값이다. 주) 숫자는 정수, 소수, 음수, 양수로 입력해야 한다.

3.3 Add Tag Attribute to Objects

이 아이콘은 선택한 개체에 태그를 넣을 수 있게 한다. 개체를 선택한 후에 원하는 이름을 넣는다. 이 태그는 이름만 있을 뿐 어떠한 값도 가지지 않는다.

3.4 Remove Attribute from Objects

이 아이콘은 선택한 개체의 Attribute를 제거한다. 화면 안의 모든 개체를 선택하고 이 아이콘을 사용한다면 모든 개체의 태그가 제거 된다.

여러 개의 태그가 있을 경우 제거를 원하는 값을 입력하면 그 값이 제거된다. 하지만 라이노 개체의 고유 속성은 지울 수 없다.

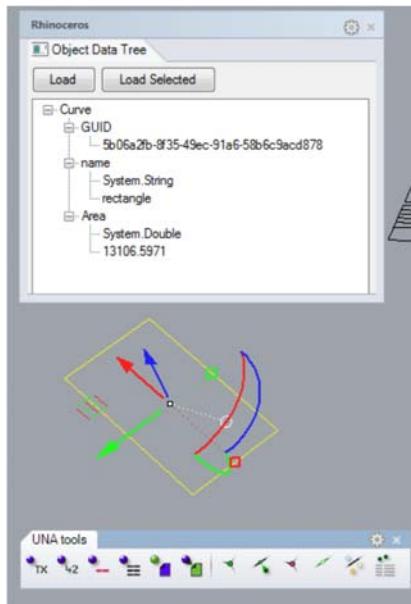
3.5 Save Result as Weight

This tool allows you to one result at a time to the object attributes, optionally assigning it a custom column name in the table.

3.6 Show Attribute Tree

이것은 오브젝트가 지닌 attribute를 볼 수 있게 해준다. 트리 구조로 이름과 값, 값의 타입을 보여준다. 그리고 모든 개체들은 라이노 고유의 GUID를 가지는데 맨 처음 볼 수 있는 값이다. 아래 이미지에는 선택된 사각형이 지닌 GUID와 각각의 값을 보여준다. Load 버튼은 라이노 화면에 나타난 모든 개체의 속성을 표시하는데 개체가 많을 경우 느려질 수 있다. Load selected 버튼은 선택된 개체의 속성만 보여준다.

In the image below, the rectangle that is selected has a GUID, a Text string Attribute called "Name" with a value of "rectangle" and a Numeric Attribute called "Area" with a value of "13106.5972".



3.7 Import points

이것은 GIS의 데이터나 다른 테이블 파일을 라이노로 불러오는데 사용된다. 불러온 테이블 시트에 XYZ 좌표가 있다면 라이노에 포인트로 그려진다. 원본의 테이블 시트의 다른 모든 항목들도 특성을 지닌 포인트로 불려온다.

라이노에서 테이블의 항목을 인식하기 위해서는 몇 가지 제한되는 것들이 있다.

- 띄어쓰기는 무시된다. (e.g. "My Name" is converted to "MyName").
- 강조된 첫 번째 형태는 자동으로 제거된다.
- 항목의 이름에는 숫자가 들어갈 수는 있지만 전체가 숫자라면 안될 수 도 있다.
- 몇 가지 이름들은 UNA에서 먼저 사용되었을 수 있다. "none" and "count" 등은 항목이름으로 쓸 수 없다.

3.8 Import table

이 툴은 기존의 attributes 데이터나 GIS나 엑셀 등의 데이터 셋을 가져올 수 있다. 다른 attribute 테이블을 불러오기 위해서는 tsv나 csv 포맷으로 변환을 해야 한다. 엑셀에서 쉽게 변환할 수 있다. 커맨드 라인의 기본 포맷은 "tsv"이며 csv로 바꾸려면 커맨드 라인에서 "Format=tsv"를 클릭한다.

File을 클릭하면 브라우저 창이 열린다. "Update=all"은 임포트된 모든 데이터를 업데이트하고 선택한 개체만 임포트 할 수 도 있다.

3.9 Export

이 툴은 입력한 개체의 특성이나 라이노 UNA분석으로 얻어진 특성을 테이블 시트로 내보내는데 이용된다. 커맨드 라인에는 몇 가지 옵션들이 있는데 "tsv" 또는 "csv" vaot으로 내보내거나, 전체 시트를 클립보드 혹은 파일로 내보낼 수 있게 한다. 클립보드에 복사된 데이터는 엑셀 데이터 시트에 복사할 수 있다. 이 기능은 개체의 weight를 조절하거나 엑셀에서 다른 특성 값과 합칠 때 유용하다.

다음의 툴들은 공간 네트워크를 만들고 네트워크에서 출발과 목적지의 포인트들 더하거나 제거하는데 이용된다.

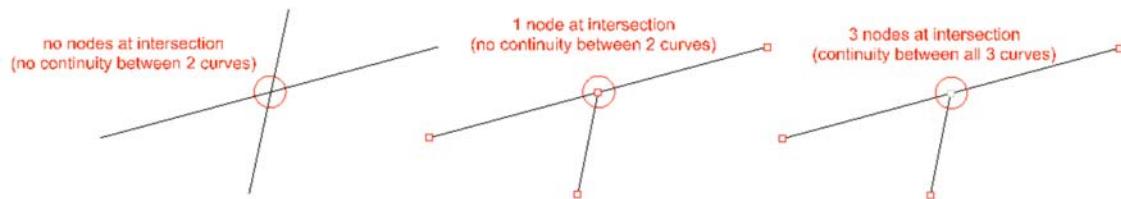
3.10 Add Curves to Network

라이노 UNA 툴박스에서 제공하는 공간네트워크 방법은 GIS UNA 툴박스와 비슷하다. 다른 네트워크 분석이라도 두 가지 입력 값이 필요한데 첫째는 네트워크 분석 입력이고 두 번째는 위치 입력이다. GIS에서는 여러단계가 필요하고 자동으로 하기 어렵다. 게다가 GIS에서 각 중심도 분석 계산을 시작할 때 네트워크상의 입력된 점들 사이에서 근접행렬이 계산된다. 데이터셋의 크기에 따라 상당한 시간이 소요될 수 있다. 이러한 두가지 절차- 네트워크를 만들고 근접행렬을 만드는 과정-는 라이노에서 꽤 효과적이다.

네트워크 툴의 add curves는 네트워크 상에서 당신이 만들고자 하는 커브를 선택할 것을 요청한다. 원하는 모든 커브를 선택하고 엔터를 누르면 자동으로 선택된 커브들이 네트워크로 들어가고 나중에 분석을 위한 근접행렬이 만들어진다.

왼쪽 클릭은 더 많은 커브를 선택할 수 있게 해준다. 먼저 선택된 커브들도 커브의 GUID 값에 따라 자동으로 두 번 계산되지 않는다. 오른 클릭은 네트워크로부터 커브를 제거할 수 있게 한다.

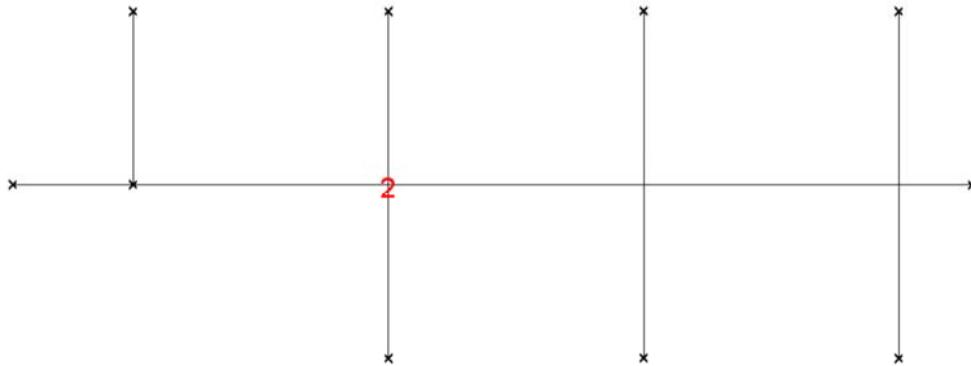
네트워크 분석을 하기 위해서 네트워크 커브들은 당신이 분석한 “입력위치” 사이에서 연속성이 필요하다. 만일 네트워크 커브들이 공통의 끝점(end node)를 가지지 못한다면 다른 네트워크 세그먼트에 기초한 입력위치는 위상학적으로 서로 연결되지 못할 것이다. 한 커브가 다른 커브의 위에서 끝난다면 후자는 교차점에서 노드를 가지지 못하여 위상학적으로 두 커브 사이의 연속성이 없을 것이다. 공통의 노드를 가지지 못한 커브들은 3차원의 고가나 지하 차도의 모델로 사용될 수 있다.



이 툴은 어떠한 커브라도 인식할 수 있다. (line, polyline, curves, arcs, etc.) 네트워크 상의 커브는 공통의 노드를 공유하기만 한다면 2차원, 3차원도 인식한다. 2차원 네트워크는 도시 내에서 거리와 빌딩을 연결하는 방법으로 쓰일 수 도 있다. 3차원의 경우에는 도시의 인프라 스트럭쳐나 건물에서 순환시스템과 레이아웃을 분석할 수 있는 기회를 제공한다.

시각적 명확성을 위해 UNA 그래픽 옵션의 디폴트 값은 네트워크의 dead-end 또는 naked end를 검정 엑스로 표시한다. 노드 옵션을 끄면으로써 그래픽 옵션에서 검정엑스를 끌 수 있다. 검정 엑스는 네트워크에 위상학적 에러가 있는지 쉽게 확인 할 수 있다.

예를 들어 아래 그림에서 왼쪽 첫 번째와 두 번째 교차에 위상적 이슈가 있는 것을 알 수 있다. 검정 엑스가 그려진 첫 번째 교차점에서, 하나 또는 그 이상의 커브의 노드들이 dead end이며 다른 커브들과 연결이 안되었음을 나타낸다. 두 번째 교차점에서 빨간색의 숫자 2가 나타나는데 이는 그래픽 옵션에서 꺼다 켤 수 있는 경고메시지이다. 빨간 숫자는 노드가 dead end는 없지만 겹쳐 있으며 연결되지 않은 두 개의 커브가 있다는 것을 나타낸다. (고가도로 등)



3.11 Add Origins

네트워크에 커브들을 더할 때 네트워크에 위치점을 수행할 수도 있다. 대부분의 분석 기능은 분석의 단위로서 별개의 위치(점, 건물, 입구 등)를 이용한다. 접근성 결과들은 일반적으로 요구된 포인트 위치에 별도로 계산된다. (다만 Redundancy와 Betweenness 툴의 경우도 네트워크 연결을 보여줄 수 있다.)

라이노 UNA에서는 세 개의 입력포인트가 있다. 오리진 포인트. 목적지 포인트 그리고 관찰자 포인트

이 툴은 네트워크에 원점(origin points)을 더한다. 중심점 분석을 위해서 원점들은 분석결과가 계산된 포인트들이다. Betweenness, Redundancy, Closest Facility 분석을 위해 원점은 어디서 시작하는지를 알려준다.

실행하면 먼저 포인트 선택을 요청한다. 마우스로 포인트를 선택한 후에 엔터를 치면 다음과 같은 옵션들이 나타난다.

Press Enter to add origins to network (Search=2D SavedEdges=On):

먼저 “Search” 옵션은 포인트가 2d 혹은 3d공간에 위치하는지를 결정한다. 포인트나 네트워크가 Z축 값을 지니고 있으면 분석은 3d가 된다. 3d와 2d를 마우스로 결정할 수 있다.

“SavedEdges=On” 옵션은 포인트가 사전에 지정되고 연결된 엣지를 지나고 있다면 체크할 수 있다. 라이노 UNA에서는 네트워크의 특정한 엣지에 포인트를 묶을 수 있지만 최단거리에 있는 엣지에는 필요하지 않다. 엣지가 드로잉파일에서 발견되지 않는 포인트를 참조한다면 무시되고 대신 일반적인 최단거리의 엣지가 사용된다. 일반적으로 이 입력은 무시해도 된다.



원점들을 더했을 때 각각의 원점은 파란선과 함께 가장 가까운 네트워크 요소에 묶여진다. 이 파란 연결선은 그래픽 옵션에서 꺼다 켤 수 있다. 네트워크 라인과 마찬가지로 원클릭은 원점을 더하고 오른클릭은 원점을 제거한다.

3.12 Add Destinations

네트워크에 목적지를 더할 때 사용한다. 중심점 분석을 위해 목적지는 분석이 도달하려는 점들이다. 각각의 목적지는 원점의 결과에 영향을 미친다. 하지만 목적지 그 자체는 어떤 결과도 주어지지 않는다. Betweenness, Redundancy, Closest Facility 분석을 위해 목적지는 여정이 끝나는 점들을 제공한다. 커맨드 라인의 옵션은 Add Origin과 같다. 네트워크에 더해진 목적지 포인트는 빨간 화살표로 표시된다. 역시 Add Origin과 같이 빨간 연결선을 꺼다 켤 수 있다.



네트워크 라인과 마찬가지로 원클릭은 목적지를 더하고 오른클릭은 목적지를 제거한다.

3.13 Add Observer points

Observer points는 오직 Betweenness 분석과 관계가 있다. Observer points는 원점과 목적지가 아닌 다른 점의 결과를 Betweenness 분석에 넣을 수 있다. 원점과 목적지 사이에 Observer points를 포함하는 잠재적인 경로를 나타낸다.

예를 들어 버스정류장에서부터 출발하여 가게까지 가는 경로라면 버스정류장과 가게 사이에 건물들이 Observer points로 사용될 수 있다. 얼마나 많은 경로들이 각각의 빌딩에 의해 바뀔 수 있는지도 나타낼 수 있다.

Betweenness 분석은 각각의 observer point가 몇 번이나 지나치는지 계속 추적할 수 있다. Observer points의 weight는 분석의 부분으로 사용되지는 않는다. observer point는 네트워크에 묶여져 있을 때 회색 연결선으로 나타난다. 역시 그래픽 옵션에서 켜고 끌 수 있고 원클릭은 observer point를 더하고 오른클릭은 observer point를 제거한다.

observer point는 원점과 목적지가 같은 엣지에 있을 때 중요해진다. Betweenness 는 observer point가 항상 정확해야만 결과가 나온다.

Note that Observer points can be important if the origin and destination points are located on the same edge segment. The betweenness results for observer points are always exact, while the betweenness values for edges are approximations, found by taking the average of betweenness values of the segment's end nodes.



3.14 Delete Network

이 툴은 라이노 화면에 더해졌던 모든 네트워크들과 원점, 목적지 그리고 observer points 들을 지울 수 있다.

다음 툴은 네트워크 분석 결과를 생산하는 분석적 기능들을 설명한다.

centrality tools, betweenness, redundancy etc.

3.15 Centrality Indices

BuildCentralityIndices 툴은 네트워크와 원점-목적지를 위한 중심도 계산을 수행한다. 중심도 툴은 원점에서 주어진 목적지에 대하여 얼마나 중심에 있는지를 분석한다. 분석결과들은 원점으로 귀속된다. 만일 분석에 가중치가 주어졌다면 가중치는 목적지의 개체에 적용된다. 예를 들어 건물로부터 정류장까지 접근성을 분석할 때 건물은 원점이 되고 버스정류장은 목적지가 된다. 이때 얼마나 많은 버스가 각 버스정류장에 서는지를 숫자로 가중치를 줄 수 있다.

먼저 커맨드 라인에서 분석을 위해 원점을 선택(select the Origins) 하거나 미리 선택(pre-selection)된 것을 수락하라는 메시지를 볼 수 있다. Pre selection 은 자동으로 모든 이전에 네트워크상에 더해진 원점들을 감지한다. 분석을 위해 그것들 모두 사용하기를 원한다면 pre selection을 선택해라. 만일 현재 분석을 위해 포인트를 선택하기 원한다면 포함시키길 원하는 원점들을 선택해라.

모든 중심도 결과는 선택한 점들에 의해 계산된다. Selected Origins은 잠시 파란색 엑스로 표시된다.



다음, 목적지를 같은 방법으로 선택한다. 선택된 목적지는 빨간 엑스로 표시된다.



원점과 목적지를 선택하고 나서 다음과 같은 입력창을 볼 수 있다.

Search Radius <600> (Reach=On Gravity=On Closeness=On Straightness=On Weight=Count Beta=0.004 Alpha=1):

Search Radius 입력은 선택한 접근성 측정을 계산하는데 이용될 네트워크 반지름을 정의한다.

분석할 때 각각의 원점을 위해 Search Radius 보다 작은 원점으로부터 최단거리에 있는 목적지만을 고려한다.

Search Radius 단위는 도면 단위를 따른다. 도면이 미터로 되어있다면 Search Radius 역시 미터 단위이다. 활성화된 Search Radius는 커맨드 라인의 시작에서 보여진다. Search Radius는 커맨드 라인에서 바꿀 수 있다.

다음은 Centrality 툴에서 제공하는 측정 기준옵션들이다. 각 옵션들은 on상태가 디폴트이며 총 4개의 중심도 측정기준이 있다. **Reach, Gravity Index, Closeness and Straightness** 각 측정기준들은 클릭으로 켜고 끌 수 있다.

가중치 옵션은 목적지 특성에 가중치를 더할 수 있다. 예를 들어 네트워크 위에 각 건물로부터 600m 이내에 걸어서 갈 수 있는 얼마나 많은 직업이 있는지 측정하기 위해서 Reach 분석을 할 때 각 건물에 직업의 개수를 표시하는 직업이라는 가중치를 줄 수 있다. 이 결과는 600m 거리 이내에 각 원 건물로부터 도달된 직업의 개수를 표시할 수 있다.

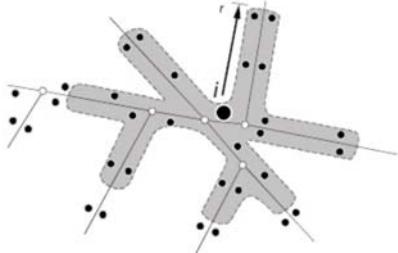
마지막의 Beta와 Alpha 같은 Gravity Index에 영향을 미친다.

Reach

Reach 측정은 네트워크 위에 주어진 Search Radius 안에서 얼마나 많은 주변포인트들(건물, 문, 버스정류장 등)이 각 건물로 도달하는지를 알아낸다. 그래프 G 에서 원점 i 의 중심도 도달(reach centrality) $R^r[i]$ 은 최대 개수 r 의 최단거리에 원점 i 으로부터 도달할 수 있는 그래프 G 에 목적지의 개수를 나타낸다.

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

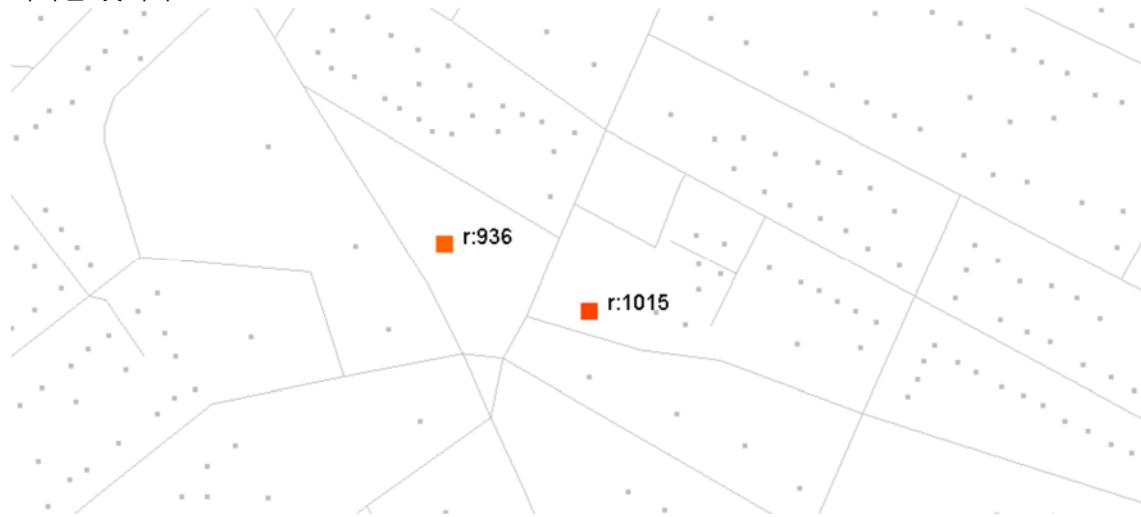
$d[i,j]$ 는 그래프 G 에서 원점 i 과 목적지 j 사이의 최단거리 경로이며, $W[j]$ 는 목적지 j 의 가중치이다. 그림은 Reach Index가 어떻게 시각적으로 표시되는지를 보여준다. 버퍼는 각 빌딩들로부터 네트워크 상에 모든 방향들을 제한하는 반경 r 에 도달할 때까지 추적한다. Reach Index는 네트워크 위의 Search Radius 안의 원점들로부터 발견된 목적지 j 의 개수에 반응한다.



Reach는 다른 어떤 형태의 목적지에 접근을 측정하기 위해 조정될 수 있다. 주어진 Search Radius 안에서 얼마나 많은 목적지의 개수가 도달되는지 간단한 계산을 위하여 Weights=count를 정할 수 있다. 그래서 가중치를 없게 하고 오직 목적지 건물의 총계만을 나타낼 수 있다. 예를 들어 빌딩 사이즈에 따라 측정에 가중치를 준다면, "건물 GFA" 특성을 목적지에 줄 수 있다. 그리고 가중치로 쓸 수 있다. 그러면 Reach 측정은 네트워크 위의 각각의 원점 주변에 Search Radius 내에서 얼마나 큰 빌트 볼륨(built volume)이 얹어지는지를 계산해 줄 것이다. 활동이나

목적지의 토지이용을 보기 위해서 가중치로써 목적지에 직업의 개수나 주민의 숫자, 그리고 설립된 사업의 숫자 등을 이용할 수 있다.

아래 그림은 케임브리지의 두 개의 지하철 입구로부터 Search Radius 600m 이내의 Reach 값을 나타낸 것이다.



Gravity Index

Reach 측정이 단순히 주어진 Search Radius내에서 원점 주변의 목적지의 개수를 카운트하는 것이라면(목적지의 특성에 따라 가중치를 줄 수는 있다) Gravity는 각 목적지에 도착하는데 필요한 이동비용(travel cost)의 요소를 더하여 측정한다. 한센(1959)이 처음 소개한 Gravity index는 교통 연구에서 가장 유명한 공간 접근성 측정 방법 중 하나이다.

Gravity 측정은 원점 i 에서 접근성이 목적지 j 의 끌어당기는 힘(가중치)까지 비례하고 원점 i 과 목적지 j 사이의 거리에 반비례하다고 가정한다.

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

여기서 $Gravity[i]^r$ 는 그래프 G 내의 Search Radius r 에 원점 i 에서의 Gravity index이며, $W[j]$ 는 목적지 j 의 가중치이다. $d[i,j]$ 는 원점과 목적지 사이의 측지(최단)거리이다. α 는 목적지의 가중치나 끌어당기는 힘의 효과를 조절할 수 있는 지수이다. 그리고 β 는 거리감쇠의 효과를 조절하는 지수이다. Gravity index는

앞에 언급한 바와 같이 접근성의 혼합된 측정안에서 양측의 목적지의 끌어당김($W[j]^\alpha$)뿐 아니라 그러한 목적지($d[i,j]$)에 도달하는데 요구되는 이동의 공간적인 저항도 포함하여 보여준다. 만일 가중치 특성이 주어지지 않는다면 각 목적지의 가중치는 1로 결정된다. α 의 기본 값은 마찬가지로 1이며, 목적지의 가중치는 직선모양으로 나타난다.

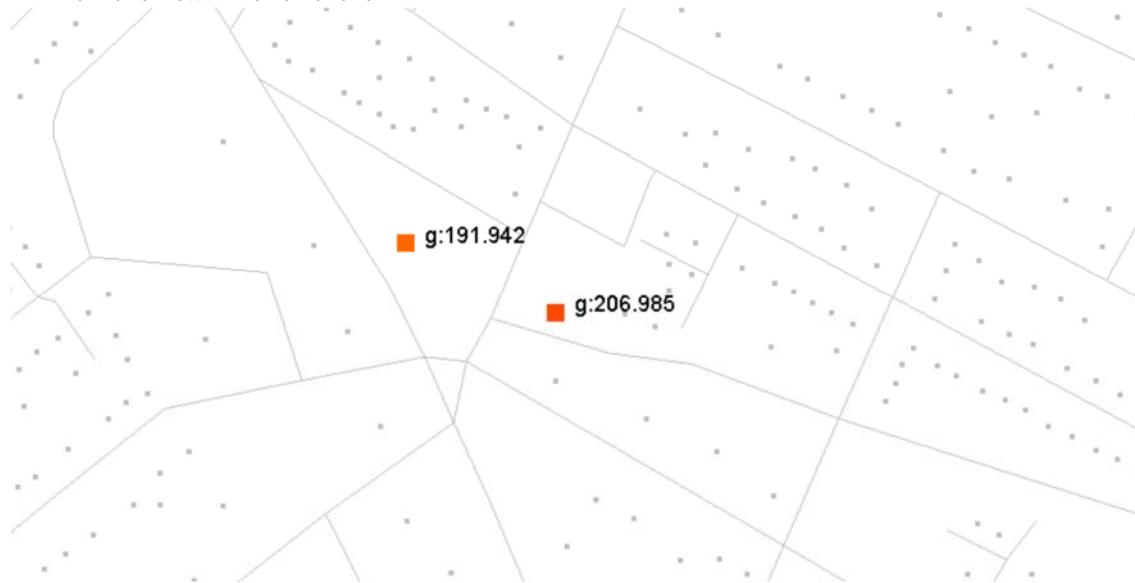
Gravity index에서 거리의 역효과는 기하급수적으로 감소한다. 거리감소의 정확한 형태는 지수 β (거리조락계수)로 조절 될 수 있으며 Centrality 툴이 실행될 때 커맨드 라인에서 특정될 수 있다. β 와 거리감소에 일치하는 형태는 이동의 추정된 모델로부터 이끌어 내진다. 예를 들어 걷는 것은 "minute"으로 측정되고 연구자들은 대략 0.1813 범위의 β (마일단위)를 발견했다.

(researchers have found to fall around 0.1813), Handy and Niemeier 1997) 4

이것은 미터에서 0.00217과 일치한다. 예를 들어 싱가폴의 상황에서 보면, 우리는 걷기를 위한 β 값은 0.005 사이에서 결정해왔다. 높은 β 값(1에 가까운)은 걷기에 매우 안 좋은 환경이다.

4. The equivalent value of Beta for impedance units in "feet" 0.000663; in "kilometers" 2.175, and in "miles" 3.501.

아래 그림은 케임브리지의 두개의 지하철 입구로부터 Search Radius 600m 이내의 Gravity 접근성의 결과를 나타낸 것이다. Gravity 결과 값이 거리 감소효과에 기인하여 얼마나 더 Reach 값보다 작게 나왔는지 주목하자



Closeness

원점의 Closeness는 원점으로부터 가장 짧은 경로를 따라 Search Radius 범위를 포함하는 모든 특정된 목적지까지 도달하는데 요구되는 평균 거리로 정의된다. (Sabidussi 1966) Gravity와 비슷하며 Closeness 측정은 원점이 주어진 거리 한계 내에서 목적지까지 얼마나 가까운지를 나타낸다. 하지만 사용되는 거리는 단순한 직선 거리이며, 결과는 모든 목적지 사이에서 평균적인 가까움(closeness)를 보여준다. Closeness 측정은 아래와 같이 정의된다.

$$\text{Closeness}[i]^r = \frac{\sum_{j \in G - \{i\}, d[i,j] \leq r} \frac{W[j]}{d[i,j]}}{n}$$

여기서 $\text{Closeness}[i]^r$ 는 Search Radius r 내에서 원점 i 의 가까움(closeness)을 의미하며,

$d[i,j]$ 는 원점 i 과 목적지 j 의 노드 사이의 최단거리이다. 그리고 $W[j]$ 는 목적지 j 의
가중치이다. 그리고 n 은 발견된 목적지의 개수이다.

* **Closeness Centrality (근접 중심성)**

► Wikipedia의 Closeness Centrality에 대한 설명 : http://en.wikipedia.org/wiki/Centrality#Closeness_centrality

Straightness

Straightness 측정법(Vragovic, Louis, et al. 2005)는 원점에서 목적지까지 최단 경로가 유클리드
기하학의 직선과 비슷한 범위를 나타낸다. Straightness 측정법은 평면에서 직선거리를 비교하여
네트워크의 기하학적 제약이 원인이 되는 이동거리의 명확한 편차를 보여준다. Straightness
측정은 일반적으로 다음과 같이 정의 된다.(Porta, Crucitti et al. 2005)

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

여기서 $\text{Straightness}[i]^r$ 는 Search Radius r 내에서 원점 i 노드의 Straightness 를 의미하며,
 $\delta[i,j]$ 는 원점 i 과 목적지 j 사이의 유클리드 기하학의 직선을 의미한다. 그리고 $d[i,j]$ 는
네트워크에서 같은 노드 사이의 최단거리이다. Straightness index는 원점에서 주변의 목적지까지
최단경로연결이 원점 i 과 목적지 j 사이의 직선거리와 비교했을 때 얼마나 차이가 나는지를
나타낸다.

자연스럽게 노드 사이의 거리가 길어짐에 따라 네트워크 거리와 직선거리 사이의 차이가
줄어들기 시작한다. 보스턴에서 로스앤젤레스까지의 도보가 직선거리로는 MIT에서 보스턴
다운타운까지 보다 훨씬 더 가깝다.

아래 이미지는 Search Radius 600m 이내의 케임브리지의 모든 건물들로부터 모든 다른
건물들까지 Straightness 결과를 보여준다. 따뜻한 색은 발견된 주변의 모든 목적지까지 더 큰
직선 값을 가진 원점을 나타낸다.



3.16 Service Area

Service Area 툴은 목적지와 원점으로부터 주어진 네트워크 반경 범위를 포함하는 경로 세그먼트를 선택하거나 복사한다.

예를 들어 이 툴은 지하철 정거장(원점)으로부터 반경 200m 범위내의 모든 레스토랑(목적지)을 선택하는 데 이용될 수 있다. Search Radius는 <200>이 기본 값이며 커맨드 라인에서 변경할 수 있다. 이 툴은 4가지 옵션을 가지고 있다.

```
Service area <600> ( SelectPoints=On SelecCurves=On Copy=Off Tight=Off )
```

SelectPoints와 SelectCurves는 특정된 네트워크 반경 범위를 포함하는 목적지나 커브들을 선택할 수 있게 해준다. 커브들은 전체의 길이를 가지고 선택된다는 점에 주의 하자. 그리고 Service Area 반경에 정확하게 일치하는 더 짧은 커브로 자르면 안된다. 추가적인 옵션 Tight는 반경 범위내에 완전히 들어간 커브만 선택할 수 있게 만든다.

Copy 옵션은 활성화된 라이노 레이어에 선택된 개체들의 복사본을 만드는 기능을 가진다. 이 툴은 GIS의 네트워크 Analysis Service Area 툴과 비슷하다. 하지만 이것은 목적지뿐만 아니라 주어진 네트워크 반경내에 도달하는 네트워크 커브도 선택한다.

아래 이미지는 Search Radius 600m 내에서 케임브리지의 지하철 입구에서(녹색) 주변의 모든 건물들(목적지)까지의 Service Area 결과를 보여준다. 노란색의 선택된 건물과 거리의 세그먼트들은 원점(녹색 지하철 입구)으로부터 네트워크 반경 600m안에서 도달할 수 있는 목적지를 나타낸다.



3.17 Redundancy Index

Redundancy Index는 **Detour Ratio**라 불리는 주어진 퍼센트에 따라서 원점과 목적지 사이의 최단거리가 연장될 때 사용할 수 있는 선형 거리의 증가를 계산한다. 2014년 GIS UNA toolbox에서 처음 소개되었고

Index는 원점과 목적지 사이에서 대안적인 풍부한 루트를 찾는다. 가능한 모든 루트가 발견되었을 때 index는 원점과 목적지 사이 모든 루트의 결합된 길이를 원점과 목적지 사이의 최단거리로 나눈다. (Sevtsuk, Mekonnen et al. 2014)

결과는 최단거리보다 몇% 정도 증가하는 경로를 따라 걷는다고 가정했을 때 얼마나 더 많은 선형거리들의 길이가 산책으로 활용 가능할지 가능할 수 있는 인자로서 설명된다. **Detour Ratio**라 불리는 이 퍼센테이지는 사용자가 입력할 수 있고 1보다 크거나 같은 숫자로 나타난다. Detour ratio 1.2 는 모든 루트가 최단거리보다 20% 증가하는 루트를 의미하고 이를 분석하여 나타낸다.

Index가 observer points와 함께 가중치가 주어지면 결과는 최단 경로와 비교하여 다른 많은 경로들을 따라서 원점에서 목적지까지 얼마나 더 많은 점의 가중치들이 도보로 접근할 수 있게 되는지를 보여준다.

한가지 이상의 목적지를 설정하면, 커맨드 라인의 Search 옵션은 각 원점에서 가장 가까운(Nearest) 목적지만을 선택할 것인지 모든 목적지 또는 원점으로부터 주어진 반경(Radius)에 포함되는 모든 목적지를 선택할 것인지를 묻는다. Draw 옵션은 발견된 루트를 시각화 시킬 것인지를 조절한다.

Redundancy Index가 발견한 루트들은 단순한 루트일 필요가 없음에 주의하자. – 참조된 논문(Sevtsuk, Mekonnen et al. 2014)에서 말한 것과 같이 그들은 순환루트와 되돌아가며 반복하는 노드들을 포함하고 있을지도 모른다. 만약 detour ratio가 적용되었다면 Redundancy Index와 Redundant Route 총계 역시 Betweenness 툴에 의해 나온 결과들이다.

아래 예시에서 1.2 의 Detour Ratio를 사용하며 왼쪽의 원점에서 오른쪽 목적지까지의 Redundancy Index는 3.30이다. 이것은 최단거리보다 20% 길어진 루트를 고려했을 때 최단 루트와 비교하여 원점에서 목적지까지 루트의 루트 길이가 3.3배 확장되었음을 의미한다. 커맨드 라인에 Draw 기능이 켜 있다면 경로의 세트는 폴리라인으로 그려지고 그룹으로 저장된다.



3.18 Redundant paths

이 툴은 원점들과 목적지들의 세트 사이의 최단거리 보다 X% 더 긴 모든 개별적인 대안 경로들을 찾는다. 이 툴은 보행로 루트 선택을 연구할 때 사용될 수 있다. 예를 들어, 원점과 목적지 사이에서 사람들이 걷기를 기대할만한 가능한 모든 산책로를 구별할 수 있게 한다. Redundancy Index와는 달리 여기서 발견된 경로들은 단순한 경로이다. 그것들은 순환하거나 반복하는 노드를 지니지 않는다. 나온 경로들은 그룹 지어지지 않고 분리고 원점에서 목적지까지 정확한 각각의 개별 루트가 그려진다.

아래 그림은 최단 경로보다 +10%의 detour ratio를 지닌 왼쪽의 원점으로부터 오른쪽의 목적지까지 다양한 경로(붉은색 커브)를 표시한다.



3.19 Betweenness

이 툴은 공간네트워크에 특정한 위치에서 우회 교통량이나 사람들 수(footfall)와 비슷한 Betweenness index를 계산하고 시각화한다. 건물의 Betweenness는 특정한 위치를 지나는 네트워크에 다른 원점들과 목적지들 사이 최단 경로들의 비율에 의해 정의된다(Freeman 1977). 만일 두 개의 노드 사이에 한 개 이상의 최단 경로가 발견된다면, 종종 바둑판 그리드 형태의 거리에서 발견되는, 그렇다면 각각의 같은 길이의 경로는 가중치가 하나로 되는 그런 동등한 가중치가 주어진다. Betweenness 측정은 다음과 같이 정의된다.

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

여기서 $\text{Betweenness}[i]^r$ 는 Search Radius r 내에서 위치 i 에 대한 betweenness이다. (Search Radius에서 지정); $n_{jk}[i]$ 는 i 를 지나가는 원점 j 에서 목적지 k 까지 최단경로의 개수이다. 그리고 n_{jk} 는 원점 j 에서 목적지 k 까지 최단경로의 총 개수이다. 위치 i 의 Betweenness는 원점 j 과 목적지 k 로부터 거리 r 반경 내 있는 건물 j, k 의 모든 쌍을 고려하여 계산된다. 하지만 위치 i 로부터 거리 r 반경 내 있는 건물 j, k 의 모든 쌍을 고려하여 계산되지 않는다. 왜냐하면 우리가 경로를 선택할 때 거리 r 보다 더 멀리 떨어진 원점들과 목적지들 사이의 경로는 고려하지 않기 때문이다. 만일 j, k 가 서로로부터 r 반경 내에 위치하고 있다는 것을 알았을 때 그리고 위치 i 를 지나는 j 에서부터 k 까지 최단경로(혹은 그 반대로도)는 그러면 j 와 k 모두 i 로부터 거리 r 이내에 있게 된다.

이 툴을 실행하기 전 먼저 네트워크를 만들고 장소를 그것에 더해야만 한다. 먼저 원점(Origin)을 클릭한 다음 경로를 만들기 위한 원점으로 이용할 포인트를 선택할 수 있다. 그 다음엔 목적지를 선택하라.

Betweenness (Search=Nearest DetourRatio=1.2 Weight=Count):

Search 옵션엔 세가지 항목이 있다. : "All", "Nearest" or "Radius" 이 항목들은 어떤 목적지가 각 원점들에 의해 사용될 것인가를 결정한다. "All"은 각 원점들로부터 주어진 모든 목적지를 사용하는 루트를 나타낸다. 예를 들어 원점이 집들이고 목적지가 버스정류장들이라면 "All"은 그것들이 얼마나 멀리 떨어져있던 상관없이 각각의 집으로부터 모든 버스정류장까지의 경로를 만든다. "Nearest"는 각 원점으로부터 가장 가까운 가능한 목적지까지의 오직 한가지 루트만을 나타낸다. 앞의 예에서 "Nearest"를 선택하면 각각의 집으로부터 각기 제일 가까운 곳의 버스정류장까지의 경로를 만든다. "Radius"는 반경을 결정할 수 있게 한다. 원점으로부터 주어진 반경내의 목적지를 선택하여 루트를 나타낸다. 앞선 예에서 "Radius"를 선택하고 600을 입력하면 각 집으로부터 반경 600m 내의 버스정류장만을 선택하여 도보 모형을 만들 수 있다.

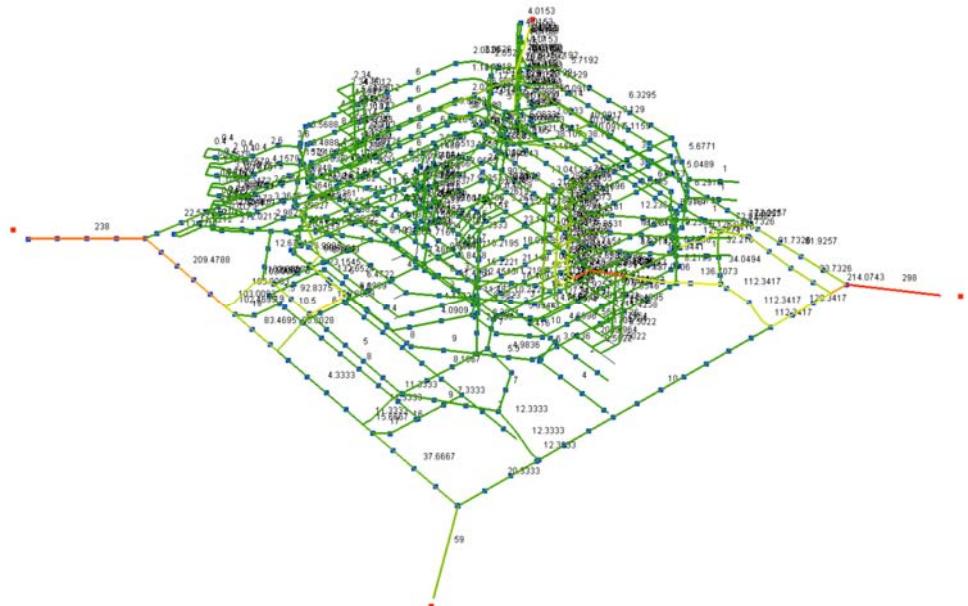
라이노 UNA에서 Betweenness 알고리즘은 공간 네트워크 상에서 현실적인 보행로를 추정하기 위한 유용하고 실용적으로 만들기 위해 계속 조정해왔다. 전통적인 Betweenness index가

원점들의 세트로부터 목적지의 세트까지의 경로를 최단경로를 따라 얼마나 많은 여정들이 각 루트들을 따라오는지 계속 파악하며 추정했지만, 라이노 UNA에서 Betweenness 알고리즘은 최단 경로 추정을 쉽게 만들어준다. "Detour Ratio"는 원점과 목적지 사이의 최단경로보다 일정 퍼센트 이상 늘어나는 경로를 만들 수 있게 한다. (맥시멈은 200%로 고정되어 있다.) "Detour Ratio"를 사용할 때 1.1은 최단경로보다 10% 늘어나는 경로를 의미한다. 모든 대안 경로들 사이에 원점의 가중치를 나누어 각 대안 경로들은 동등한 가능성이 주어진다. 이는 도시에서 사람들이 최단경로를 선택할 필요성이 없을 때 유용하다. 선행 연구에서 일반적으로 보행자들은 유용하거나, 기쁨을 즐기거나, 또는 안락함을 위해 최단경로 보다 대략 10~20% 정도 긴 경로를 선택하는 것이 발견되었다.

다른 중심도 지수(centrality indices)와는 달리, Betweenness 툴은 "observer points"를 이용할 수 있다. Observer는 원점이나 목적지 자체가 아닌 한 점이다. 하지만 Betweenness 결과는 계산된다. 이것은 루트상에서 지나치는 도시나 거리 혹은 건물 안의 방을 의미하는 점이 될 수도 있다. 먼저 Observer는 원점과 목적지와 마찬가지로 네트워크에 더해져야 한다. Observer가 사용되지 않는다면 Betweenness 결과는 네트워크 경계로 되돌아간다.

Weight는 분석에 원점의 특성에 따라 가중치를 준다. 원점이 100의 가중치를 가지고 있다면 그리고 그것이 의미하는 것이 주민의 수라고 한다면, 가중치가 주어진 Betweenness는 원점으로부터 목적지까지 하나의 여정 대신에 100개의 여정을 만들 것이다. Weight가 "Count"로 세팅이 되어 있으면 각 원점은 1로 카운트 된다. , routing one trip from each origin to its destination(s).

아래 이미지는 3D 건물 내부에 712개 사무실 도어(파란 점)로부터 가장 가까운 비상구(붉은 점, 4개의 코너)까지의 Betweenness 값을 보여준다. 그리고 detour ratio는 +20%를 적용하였다. 이 결과는 비상대피 시 어느 복도가 제일 혼잡스러운가를 보여준다. 붉은 색에 가까울수록 혼잡하다.

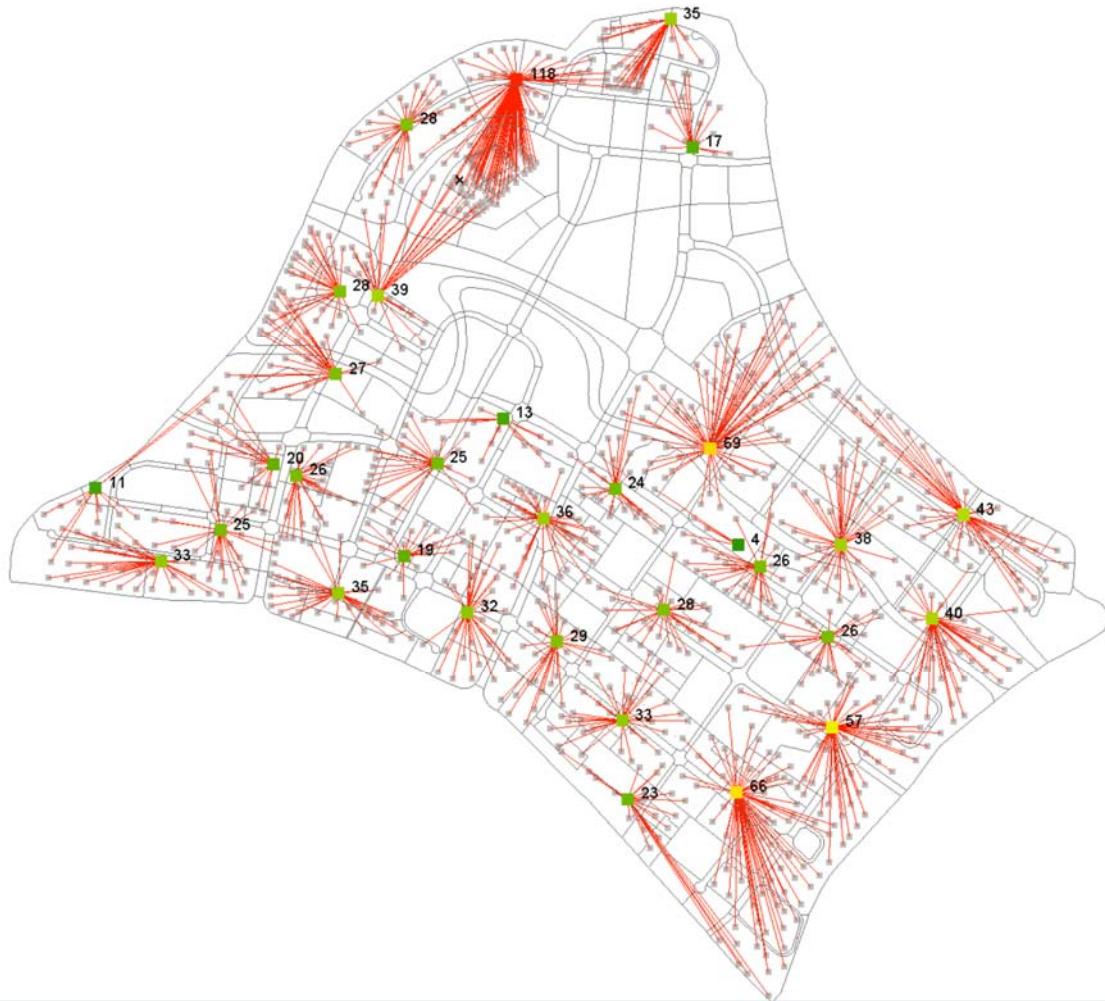


3.20 Closest Facility

이 툴은 각 원점에서 네트워크를 따라 가장 근접한 목적지의 시설을 발견하고, 얼마나 많은 목적지 시설의 마켓 구역에 독자적인 원점이 Reach 혹은 Gravity 입구 내에 있는지를 요약한다. 일반적인 Reach 와 Gravity 측정기준과 달리 각 원점은 그것이 가장 근접하게 기댈 단 하나의 시설에 묶여있다. 커맨드 라인에 다음과 같은 옵션들이 있다.

Search Radius <1500> (Weight=Count Gravity=Off Beta=0.004 Lines=On SaveAs):

Search Radius는 목적지 시설들까지 원점 연결을 위한 맥시멈 네트워크 반경을 결정한다. Weight는 분석이 원점에 숫자 특성에 의해 가중치가 반영된 결과를 얻게 해준다. 기본 결과값은 각 시설에서 고유한 원점의 Reach 결과를 지시한다. 하지만 Gravity를 켰을 때 또한 gravity 결과를 더한다. Beta 값은 Gravity 측정에 감소 효과를 조절한다. (Gravity 측정기준 참조) Lines 옵션은 툴이 각 목적지의 시설로부터 각기 연결된 원점까지 직선을 그린다.- 이 선들은 단지 시각적 간결함을 위한 것임에 유의하자. 시설에 가장 근접한 점은 여전히 네트워크상에서 결정된다. SaveAs는 결과값에 이름을 부여할 수 있다.



3.21 Graphic Option

이 툴은 UNA의 그래픽 표현에 변화를 줄 수 있다 : 결과의 컬러 계획, 라벨 디스플레이, 점과 연결선 등. 아래의 옵션은 커맨드 라인에서 볼 수 있다.

```
Graphics ( Color=GreenToRed Results=Reach Weight=SumCount Mode=Weight Node=On NodeId=Off NodeD2=Off DotConnections=Off  
DotConnections=Off DotArrow=Off DotLabel=On DotId=Off Dots=On Edges=On EdgeLabels=On Font=18 DotSize=14 );
```

- Color는 UNA 분석 결과의 시각화 컬러를 변경할 수 있다.
- Result 옵션은 어떤 분석결과가 디스플레이 될지를 조절한다.
- Weight 옵션은 어떤 개체의 가중치(숫자 특성)이 디스플레이 될지를 조절한다.
- Mode 옵션은 분석결과를 그래픽으로 표시할지 숫자로 보여줄지 결정한다.
- Node 옵션은 네트워크의 dead-end에 작은 검정 X 심볼을 표시할지 결정한다.
- NodeId는 디스플레이 모드로부터 라이노 노드 ID를 켜거나 끈다. NodeId는 네트워크 노드이다. 사용자가 일반적으로 보는 것을 원하지 않는 자동적으로 생성되는 값이다.
- NodeD2 옵션은 . 어떤 노드가 다른 것과 연결되지 않으며 degree 2를 가진 노드인지를 말해주는 숫자값 2와 함께 뜨는 경고를 켜거나 끌 수 있다
- Dot Connections 옵션은 원점, 목적지, 그리고 observer point에서 가장 가까운 네트워크에 연결될 때 보여지는 파란색, 빨간색, 회색 연결선이 보여질지를 결정한다.
- DotArrow 옵션은 각 노드에서 방향을 보여주기 위해 파란색 빨간색 회색 화살표를 네트워크 위에 더할지를 조절한다.
- DotLabel은 숫자로 된 UNA 분석결과가 노드 다음에 보여질지 말지를 조절한다. DotLabel을 끄면 분석결과값이 디스플레이에서 사라진다.
- DotId는 각 네트워크 노드를 위한 UNA ID 숫자가 보여질지를 조절한다.
- Dots 는 결과 값을 가진 네트워크 노드가 보여질지를 조절한다.
- Edges는 컬러값이 네트워크 경계에 보여질지를 조절한다 이 옵션은 Betweenness 분석에만 적용된다.
- EdgeLabels 옵션은 에지로부터 숫자결과값을 보여질지를 조절한다. 이 옵션은 Betweenness 분석에만 적용된다.
- Font는 결과값의 폰트 사이즈를 조절한다.
- DotSize는 UNA결과값의 도트 사이즈를 조절한다.

3.22 Paint weight color

이 툴은 임시의 가중치나 결과값의 색을 라이노 개체의 색으로 만든다. 이는 다른 소프트웨어로 파일을 익스포트 할 때 유용하다.