# Accessibility to Green Urban Areas Analysis

Technical description

Emile Robe

*This note provides a technical description of the workflow applied to produce indicators on access to green urban areas. It accompanies the Python scripts used for this analysis.*

## Input Data

**A) 786 Copernicus Urban Atlas 2018 feature classes**
To these Urban Atlas 2018 feature classes (UATL) we added the attribute **[POPL_20 18]** (= population figures 2018 calculated by JRC, compatible with the JRC-GEOSTAT 2018 population grid)

Many urban centres overlap parts of more than one Urban Atlas feature class, each covering a FUA (Functional Urban Area). Consequently, for analyses requiring results at urban centre (HDC = high-density cluster) level (or involving their immediate surroundings), it is interesting to group contiguous UATL feature classes before performing the analysis.

Many Urban Atlas feature classes do touch each other, but some contiguous groups are too large to be manageable or relevant (for instance a group reaching from Amsterdam to Munich).
We have created a feature dataset containing 476 feature classes of grouped Urban Atlas feature classes: 786 UATL feature classes are grouped into 476 groups.
**-> UATL_2018_GROUP_[xxx] with xxx in [001..476]**

We create the following table
**MERGED_UATLS_2018_TABLE** with the following fields :
**UATL_MERGED_GROUP** : the group ID
**CROSS_CNTR** : Flag with the values:
0 : the grouped UATL 2018 feature class extent is entirely within the national boundaries of a country
1 : the grouped UATL 2018 feature class extent crosses several national boundaries
*This field is used in step one of the procedure* ***<u>Creation of a pedestrian network</u>*** *described later in this document.*
**TOUCH_NGHBR_CNTR** : Flag with the values:
0 : the grouped UATL 2018 feature class extent does not touch the national boundaries
1 : the grouped UATL 2018 feature class extent not touch the national boundaries
*This field is used in step one of the procedure* ***<u>Creation of a pedestrian network</u>*** *described later in this document.*
**NB_UATL** : nb of Urban Atlas 2018 feature classes grouped
**LIST_UATL** : list of the original Urban Atlas 2018 feature classes in the group
for example : DE003L1#DE028L1#DE033L1#DE067L1#DE069L1#DE534L1
**HDC_2011_AVAILABLE** : flag to indicate that the merged UATL features classes contain urban centres that are sufficiently covered by UATL features to allow for a meaningful analysis.
*Only grouped UATL feature classes with* **HDC_2011_AVAILABLE** = 1 *are used for the analysis.*

**B) Road network: TomTom 2019 PedestrianPath**
This network represents all streets that are assumed to be accessible for pedestrians (i.e. the entire street network, excluding motorways and similar highways).

## Analysis steps

We use two python master scripts that call slave scripts to prevent memory errors.
The master script **GUA_Accessibility_Analysis_only_create_NW.py** calls the slave script:

**GUA_network_creation.py**

The master script **GUA_POPL_2018_Accessibility_Analysis_cloud_SA_400m.py** calls the slave scripts:

**GUA_POPL_2018_Calculate_SA_400m_cloud.py**

**GUA_POPL_2018_Overlay_SA_400m_cloud.py**

### Step 1: Creation of a pedestrian network

The master script **GUA_Accessibility_Analysis_only_create_NW.py** calls the slave script:**GUA_network_creation.py**
with the following input parameters depending on the values of the fields **CROSS_CNTR,TOUCH_NGHBR_CNTR** in the table **MERGED_UATLS_2018_TABLE** :

<u>Input network</u>
If **CROSS_CNTR** or **TOUCH_NGHBR_CNTR** = 1
NW_europe_2019_SP.gdb\Routing\Streets (whole Europe)
The selection of the pedestrian streets is done in **GUA_network_creation.py**
Else:
NW_<country code>.gdb\<country code>\nw
The pedestrian streets are already selected in the input network covering the selected country.

<u>The clip feature class</u>
Selection of the boundary of the grouped FUAs feature class in:
**UATL_2018_GROUP_RG_DIS** based on **[GROUP_ID]** (dissolved version of the grouped UATL feature classes).

Creation of a 5 km buffer around the clip feature class and clip of the pedestrian streets inside this area.
Creation of the network dataset by applying the following steps:

Selection of the streets which are accessible for pedestrians, using three queries:

FOW : Form of Way : 1: Part of Motorway 3: Part of a Single Carriageway (default)
FRC : Functional Road Class :
0: Motorway, Freeway, or Other Major Road 1: a Major Road Less Important than a Motor 2: Other Major Road
FEATTYP : Feature Type 4165: Address Area Boundary Element
F_ELEV : Begin Level :

T_LEVEL : End Level :
SPEEDCAT : Speed Category: 6: 31 - 50 km/h 7: 11 - 30 km/h 8: < 11 km/h

NEW_SELECTION: ("FOW" <> 1 AND "FRC" not in (0, 1, 2) AND "FEATTYP" <> 4165) AND ("F_ELEV" <>-1 AND "T_ELEV" <> -1)
ADD_TO_SELECTION: "FOW" = 3
ADD_TO_SELECTION: "SPEEDCAT" >= 6
Calculate the time (in minutes) needed to walk along each segment of the network, based on the length of the segment and a walking speed of 5 km/h, and create a network dataset.
Copy the selected streets in the output feature class in
NW_UATL_GROUP_<**GROUP_ID**>.gdb_<**GROUP_ID**>\nw and build the new network dataset.

## Step 2: Calculation of pedestrian service areas of 400 m around green urban areas

The objective is to create an indicator of the accessibility of green urban areas within 400 meters of walking distance. From the green urban areas polygons, we create points along lines, using a fixed distance of 50 m between the points with the option: include endpoints.

This method obviously produces a large number of contour points. This number is reduced by selecting those close to the network. Contour points that are far from the streets are in principle useless, because they are supposed to be located in places where one cannot access the GUA (along rivers, railways, motorways, etcetera).

From the points along lines, we select those that are within a distance of 25 m from the road network.

We check if all GUA polygons are represented by points. If not, we add the centroid point of the GUA polygon.

Long GUA features in parallel with a street are well represented by such points. Elsewhere, the points obviously follow the shape of the GUA contour, as long as that contour is sufficiently close to a street.

### Script GUA_POPL_2018_Calculate_SA_400m_cloud.py

**Input polygons: grouped UATL feature class** with building blocks and disaggregated population data for each block (Field [POP_2018] in our case)
**Input Network dataset** with the streets accessible for pedestrians created in the first point
**The output feature class** is stored in **<output path>\<ID_GROUP>"_sa"** where the field[UATL_ID] (unique identifier of an UATL polygon) is added and calculated.

**To avoid memory leaks errors during the process of large datasets with arcpy, we use in-memory feature classes inside a loop to calculate the service areas and during the overlay analysis.**

Selection of the green urban areas (GUA): CODE_2018 field value = 14100 or 30000 or 31000 in the input grouped UATL feature class.

Creation of points from GUA borders used to create the service areas
-> step 1: use of ArcGIS tool GeneratePointsAlongLines with a distance of 50 meters between these generated points

-> step 2: use of ArcGIS tool SelectLayerByLocation to reselect points within a distance of 25 meters from the input network dataset

-> step 3: Insert centroids from GUA polygons where no point was selected in the 2 previous steps

Creation of service areas within 400 meters of walking distance from these selected points

## Step 3: Overlay between inhabited UATL polygons and pedestrian service areas of 400 meters

We dissolve the service areas based on field [UATL_ID] value of the green urban areas to get a single service area polygon for each GUA. We add the surface info of the GUA around which the service area has been created. Then we intersect the service areas and, for each overlapping part, calculate the sum of the green areas' surface. This means that, for each part of the territory covered by one or more service areas, we know what the total surface is of the green urban areas that are within 400 metres walking. Finally, we intersect the combined service areas with the inhabited UATL polygons, computing (by means of simple area weighting) the population living in each of the service area polygons.

### Script GUA_POPL_2018_Overlay_SA_400m_cloud.py

**Input polygons: grouped UATL feature class** with building blocks and disaggregated population data for each block (Field [POP_2018] in our case)

**Input Service Areas feature class <ID_GROUP>_SA** created in the second step.

**The final table** is stored in **<ID_GROUP>_SA_SP_UNION_STAT**

We dissolve the service areas with the option MULTI_PART by [UATL_ID] field values of the green urban areas (to get a single service area for each GUA) in the temporary feature class **outPath\SA_DIS_MP.**

Selection of the green urban areas (GUA): CODE_2018 field value = 14100 or 30000 or 31000 in the input grouped UATL feature class in the temporary feature class **outPath\GUA**

Add and calculate [GUA_Area] = SHAPE_Area in **outPath\GUA**

Joining field [GUA_Area] from **outPath\GUA** in **outPath\SA_DIS_MP**

Make the geometric union of the service areas in **in_memory\SA_UNION**

Make a new feature class from **SA_UNION** with only single part features to simplify complex shapes in **outPath\SA_SP**.

At intersection (where Union creates identical polygons), we calculate the sum of GUA Areas. For this purpose, we add a text field with the combined XY coordinates of the polygon centroid, from fields INSIDE_X and INSIDE_Y (The tool Dissolve is not used because of possible crash with complex topology).

We make a table **outPath\SA_STAT** with the sum of GUA Areas observed at each location using the ArcGIS tool Statistics_analysis with the input feature class **SA_SP** and the Statistic field parameter [[GUA_Area], "SUM"], the case field parameter [CENTROID_COORD]. We use the ArcGIS tool DeleteIdentical to keep one polygon when we have polygon overlaps.

We join the value of the field [SUM_GUA_Area] from the table **outPath\SA_STAT** to the feature class **SA_SP** where overlapping polygons are deleted based on field [CENTROID_COORD] value.

We intersect the combined SA dataset with the dataset containing the population:

1) use of ArcGIS tool MakeFeatureLayer with the input grouped UATL feature class where field [POPL_2018] value greater than 0 with the option [POPL_2018] VISIBLE RATIO to create the feature layer **input_popl_Layer**

2) we make the geometric union of the service areas’ single parts **SA_SP** and the feature layer **input_popl_Layer** (with the option "RATIO" to distribute population between cut polygons) in **outPath\<ID_GROUP>_SA_SP_UNION**

Finally, we make a summary statistics table based on the feature class **<ID_GROUP>_SA_SP_UNION** grouped by [HDC_2011_code] and [sum_GUA_area] fields values with the sum of population in **outPath\<ID_GROUP>_SA_SP_UNION_STAT**