# Manual Plastic Waste Flow Capturing Location Model

This is a manual of the Plastic Waste Flow Capturing Location Model. The input consists of environmental data of the surroundings, the output is a set of optimal locations given the available budget for catching systems for the chosen area.

To run the model, it is necessary to prepare the input data of the study area. This is done in part 1, where open source data is gathered in QGIS and pre-processed using some scripts and a bit of manual editing. In part 2, we move to a python environment. Here we set a few last input parameters such as properties of the catching systems and budget of the case study, and then we run the optimization model to find the best locations. In part 3, we go back to the QGIS project file and show the optimal locations on the map.

## Part 1: Setting up the QGIS project file with all necessary data

We will first set up the QGIS project with all necessary layers of GIS data. In part 1a, we will download layers from Open Street Map (OSM). In part 1b we will manually add some extra data using expert knowledge of the study area.

1. Make a copy of the folder and replace “name\_city” in the name of the QGIS project of the file “name\_city\_PW-FCLM.qgz”. Open the QGIS project.

### Part 1a: OSM layers

**For each layer, export to CRS Amersfoort when exporting to a geojson file after downloading the OSM data. We will show this for the first file, but this should be done for each OSM export.**

1. Waterpolygons

A map of a city

Description automatically generated  
Source: OSM

* 1. Open the **QuickOSM** plugin
  2. Go to **Query**
  3. Insert the following query:  
     [out:xml] [timeout:25];  
     {{geocodeArea:Delft}} -> .area\_0;  
     (  
      way["natural"="water"](area.area\_0);  
      relation["natural"="water"](area.area\_0);  
     );  
     (.\_;>;);  
     out body;
  4. Select ONLY the **Multipolygons**
  5. Click **Run query**  
     \*This example is for Delft, change "*geocodeArea*" value to pick a different city.  
     A screenshot of a computer

     Description automatically generated
  6. The layer will contain polygons of the water for all small ponds and ditches in the area, therefore it is necessary to delete the features of waterpolygons that you do not want to include in the analysis of the area. Moet ik hier uitleggen hoe dit moet met het gele potloodje en selecteren en deleten of verwacht ik dat de user dat zelf kan?A map of a city

     Description automatically generated
  7. After deleting unnecessary waterpolygons, it is probably also necessary to cut off parts of some of the remaining polygons by deleting the vertices. Idem?A map of a city

     Description automatically generated
  8. When you are happy about the final extent of your layer, it needs to be exported to a permanent file with the correct CRS as follows: A screenshot of a computer

     Description automatically generated



A screenshot of a computer

Description automatically generated

Selecting the correct folder



Ensure that the final layer is called “waterpolygons.geojson” and that it is stored in the same folder as the QGIS project!

1. Waterways:
   1. Open the **QuickOSM** plugin
   2. Go to **Quick query** and enter key:waterway. In: name\_city (Delft).
   3. Select only the lines and click **Run query**

A screenshot of a computer

Description automatically generated



* 1. The waterways are also outside of the range of the waterpolygons. So we need to clip the waterways: A map of a city

     Description automatically generated
  2. Click **Vector 🡪 Geoprocessing Tools 🡪 Clip** and sA screenshot of a computer

     Description automatically generated
  3. Change the input layer to the waterways layer and the overlay later to the waterpolygons and then click Run.A screenshot of a computer

     Description automatically generated
  4. Export the waterways layer with the correct CRS as shown in step 1.h. for the waterpolygons. Ensure that the final layer is called “waterways\_detailed.geojson” and that it is stored in the same folder as the QGIS project!
  5. Now we need to simplify the waterways layer because the level of details is too high for the number of nodes in the optimization model in Part 2. To do this, first run the **Vector 🡪 Geometry Tools 🡪 Simplify** algorithm and set the tolerance to 20. Then, manually remove and move vertices to create a layer of waterways with as little nodes as possible. Tips geven voor vertex editing? Split line and add vertex? It is fine if the simplified waterlines do not exactly fit in the waterpolygons, as long as the direction of the waterlines is approximately correct:

A map of a city

Description automatically generated

* 1. Explode the waterlines when you are done editing using **Processing Toolbox 🡪 Vector Geometry 🡪 Explode lines** with the “**Simplified”** layer as input layer and set the advanced settings to “Skip (Ignore) Features with Invalid Geometries) such that non-valid geometries from the manual editing are automatically removed.
  2. Export the exploded and simplified waterways layer with the correct CRS as shown in step 1.h. for the waterpolygons. Ensure that the final layer is called “waterways\_simplified.geojson” and that it is stored in the same folder as the QGIS project!

1. Layer of corner nodes (for deciding number of vertices that are necessary)
   1. Run the algorithm **Vector 🡪 Geometry Tools 🡪 Extract Vertices** with the waterways\_simplified layer as input.
   2. Run the **Processing Toolbox 🡪 Vector general 🡪 Delete duplicate geometries** algorithm with the vertices from step 3a. as input. This is because some vertices will be extracted in double.
   3. Export the final vertices layer (the temporary layer is called “Cleaned”) with the correct CRS as shown in step 1.h. for the waterpolygons. Ensure that the final layer is called “corner\_nodes.geojson” and that it is stored in the same folder as the QGIS project!
2. Layer of intersection nodes (no catching systems are allowed here)
   1. Make a copy of the “corner\_nodes” layer by clicking the layer and exporting it again. Ensure that the new layer is called “intersection\_nodes.geojson” and that it is stored in the same folder as the QGIS project!
   2. Set the layer to editing mode and remove the vertices that are not intersections but dead ends. Remove non null geometries! Save the layer after editing.
3. Houseboats
   1. Open the **QuickOSM** plugin
   2. Go to **Quick query** and enter key: building, value: houseboat). In: name\_city (Delft).
   3. Click **Run query**
   4. Export the houseboats layer with the correct CRS as shown in step 1.h. for the waterpolygons. Ensure that the final layer is called “houseboats.geojson” and that it is stored in the same folder as the QGIS project!
4. Sources
   1. Open the **QuickOSM** plugin
   2. Go to **Quick query** and enter the keys and values as shown below. Make sure that the dropdown menu on the left is set to “Or” and not “And”. In: name\_city (Delft).
   3. Select only the points and click **Run query**A screenshot of a computer

      Description automatically generated
   4. Export the sources layer with the correct CRS as shown in step 1.h. for the waterpolygons. Ensure that the final layer is called “sources.geojson” and that it is stored in the same folder as the QGIS project!

### Part 1b: Manual QGIS editing and pyqgis scripts

1. **Layer of extra locations where no catching systems are allowed. Create points layer**
2. Max boat width polygon (for catching probabilities)
   1. Buffer the waterways polygon using: **Vector🡪Geoprocessing Tools 🡪 Buffer** and run the algorithm with a distance of 30 meters. This is necessary because waterlines might be outside because of the simplificiation of the waterways we did previously.

**A screenshot of a computer

Description automatically generated**

* 1. Make the layer editable, open the attribute table. Add a new column called “max\_boat\_width”.
  2. Go back to the map, select the entire layer and click **Edit 🡪 Edit Attributes 🡪 Modify Attributes of Selected Features**. Change the max\_boat\_width of all features to the value of the max\_boat\_width that is true for most locations.
  3. Deselect the layer and start selecting specific parts of the layer where the max\_boat\_width is different from the value we just used for the whole layer. Edit the attributes for these selected features.
  4. Export the max boat widths layer with the correct CRS as shown in step 1.h. for the waterpolygons. Ensure that the final layer is called “max\_boat\_width.geojson” and that it is stored in the same folder as the QGIS project!

1. Shore properties (wall, vegetation or houseboats)
   1. We will make a vector layer of the outlines of the waterpolygons. First we need to dissolve the waterpolygons using the algorithm **Vector 🡪 Geoprocessing Tools 🡪 Dissolve** with the waterpolygons layer as input layer.
   2. Then we get the outlines of the dissolved waterpolygons layer by using the algorithm **Vector 🡪 Geometry Tools 🡪 Polygons to Lines** with the "**Dissolved**" layer from the previous step as the input layer.
   3. We explode the resulting layer using the algorithm **Processing Toolbox 🡪 Vector geometry 🡪 Explode lines** with the “**Lines”** layer from the previous step as the input layer.
   4. Export the final layer with the correct CRS as shown in step 1.h. for the waterpolygons. Ensure that the final layer is called “shore\_types.geojson” and that it is stored in the same folder as the QGIS project!
   5. Open the attribute table of the lines layer. Set the layer to editing mode and add a new column called “type”.
   6. Go back to the map and select the entire layer and and click **Edit 🡪 Edit Attributes 🡪 Modify Attributes of Selected Features**. Change the type of all features to “wall”.
   7. Deselect the layer and start selecting specific parts of the layer where there are houseboats (which we can see from the houseboats layer) and set the type to “houseboat”
   8. Using expert knowledge from the study area (or your own experience), select the specific parts of the shore where there is shore vegetation and set the type to “vegetation”.
   9. Save the layer after editing.
2. Water vegetation
   1. Copy the water polygons layer by exporting it. Ensure that the copied layer is called “water\_vegetation.geojson” and that it is stored in the same folder as the QGIS project!
   2. Delete the polygons that have no water vegetation. If necessary, it is also possible to edit the shapes of the single parts in the polygon layer. Save it afterwards.
3. Impact factor sensitive areas
   1. If there is an area where it is particularly important that plastic is caught, it is possible to set an impact factor for this area. Create a new polygon layer and make a polygon by clicking to create new vertices, such that the shape of the polygon is the outline of the sensitive area.
   2. Export the sources layer with the correct CRS as shown in step 1.h. for the waterpolygons. Ensure that the final layer is called “sensitive\_area.geojson” and that it is stored in the same folder as the QGIS project!
4. Get wind data from KNMI [www.knmi.nl/nederland-nu/klimatologie/daggegevens](http://www.knmi.nl/nederland-nu/klimatologie/daggegevens)
   1. Find the weather station that is closest to the study area. Download the etmgeg file and save it to the folder: “pulp\_scripts”.
   2. Rename the file to “wind\_data.txt”.
5. Sharp corners
   1. Open the script “pyqgis\_scripts/sharp\_corners.py” in the python window of QGIS and click run.
   2. A temporary layer is created. Load the style file “sharp\_corners\_style” for this layer.

A screenshot of a computer

Description automatically generated



* 1. The large green circles are corners that the script considered as sharp corners but there are still a few errors in the script. Therefore, manually check if all sharp corners are correct and change the attribute value “sharp” of corners that are considered sharp but are not from 1 to 0.
  2. Export the sources layer with the correct CRS as shown in step 1.h. for the waterpolygons. Ensure that the final layer is called “sharp\_corners.geojson” and that it is stored in the same folder as the QGIS project!

1. Run files to simplify network
   1. Open the script “pyqgis\_scripts/create\_final\_network.py” in the python window of QGIS and click run.
   2. Note: the number of nodes is decided by the model and is chosen such that the Gurobi solver can be used without a license. It is possible to change the number of nodes to add more detail, note that it might be necessary to change the solver that is used (to the open source solver CBC instead of Gurobi, it is a lot slower unfortunately).
2. Run files to create MDP input
   1. Open the script “pyqgis\_scripts/create\_final\_network.py” in the python window of QGIS and click run.

## Part 2: Optimization in Python

1. Python create venv with following packages: numpy, scipy, matplotlib, geopandas, pulp, networkx.
2. Set correct budget B, costs for each type, accuracy of each type, and w.
3. Run model for different alpha and plot this, choose alpha and set correctly.
4. Run model for final choices, output is writtin to file solution.txt

## Part 3: Show the solution on the map

1. Open QGIS project and open python file show\_solution in the window. Run it.
2. If you are unsatisfied with the result, it is wise to inspect the nodes\_attributes layer to see if the variables are correctly defined for some of the solution locations that you are doubting. If you see some odd values, just change attribute values, save the layer and go back to Part 2, step 3 and 4.