Parcel Demand Generation V1.0

Documentation

Author(s): Rodrigo Tapia, Ioanna Kourounioti

Author’(s’) affiliation (Partner short name): TU DELFT

Revision History

|  |  |  |
| --- | --- | --- |
| Version No. | Date | Details |
| 1.0 | 19/01/22 |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Contents

[1 Introduction 4](#_Toc93590418)

[1.1 Scope and objectives 4](#_Toc93590419)

[2 Requirements. 4](#_Toc93590420)

[2.1 Software requirements 4](#_Toc93590421)

[2.2 Input/Outputs 5](#_Toc93590422)

[2.2.1 Inputs 5](#_Toc93590423)

[2.2.2 Outputs 5](#_Toc93590424)

[2.3 Paths structure 6](#_Toc93590425)

[3 Model Description 6](#_Toc93590426)

[4 Instructions to run the model 6](#_Toc93590427)

[4.1 Command line execution of the model 6](#_Toc93590428)

[4.1.1 Instructions and commands 6](#_Toc93590429)

[4.1.2 Arguments 7](#_Toc93590430)

[4.2 Requirements 7](#_Toc93590431)

[4.2.1 Testing requirements 7](#_Toc93590432)

[4.2.2 Input folder (Arg[2]) 7](#_Toc93590433)

List of tables

[Table 1 Parcel Generation– Inputs 5](#_Toc93590434)

[Table 2 Parcel Generation– Outputs 5](#_Toc93590435)

[Table 3 Parcel Generation– Inputs 7](#_Toc93590436)

[Table 4 Parcel Generation– Inputs 7](#_Toc93590437)

# Introduction

## Scope and objectives

The first part consists of Mass-GT’s parcel generation as-is plus data transformation to accommodate for LEAD needs. The case study in The Hague analyses the effect of *Local-to-Local (L2L)* parcels and the crowdshipping case.

From the output of the parcel generator module, we simulate a proportion of those to be L2L. In contrast with the MASS-GT the origin of the parcels is not taken in a depot of a CEP but it is also simulated. For this, we introduce a study area that can include different groupings of neighbouring municipalities. This way we can generate L2L per urban agglomerations. For example, we can simulate that L2L parcels with destination sGravenhage can have origin in Delft, Rijswijk, SGravenhage or Leidschendam\_Voorburg. For the parcels that are not L2L, an origin in the closest depot is adopted, as the parcel generator module dictates.

Additionally, an a priori probability of using crowdshipping is used to generate the parcels that could potentially be crowdshipped. Currently we use a fixed percentage derived from the literature. In future versions this a priori probability can be replaced by a choice model that calculated the probability of a parcel to be crowdshipped as a function of its characteristics.

The parcels that are either L2L OR crowdshipped will be subject to the hyperconnectivity framework. The other parcels will continue to go through the standard hub-spoke network. It is worth noting that the parcel generation makes a first assignment of parcels to courier companies.

Key assumptions:

* C2C parcels destination & origin is based on households distribution
* B2C parcels destination based on households distribution, origin on retail jobs
* Certain percentage of parcel being local to local. ( The percentage is taken from the parcelmonitor)

# Requirements.

## Software requirements

The simulators have been built using Python version 3.8.8.

The following Python libraries need to be installed:

1. networkx==2.6.3
2. pandas==1.3.4
3. pyshp==2.1.3
4. tk==0.1.0
5. Built-in modules: itertools, math, sys, os

## Input/Outputs

### Inputs

The inputs of the parcel generation module are described in Table 1.

Table 1 Parcel Generation– Inputs

|  |  |
| --- | --- |
| Inputs | Description |
| Parameters.txt | Text file containing the paramenters of the model |
| TimeSkimMatrix.mtx | Matrix that contains the time (in seconds) between each pair OD |
| DistanceSkimMatrix.mtx | Matrix that contains the distance (in kilometres) between each pair OD |
| StudyAreaShapefile | Shapefile of the city. The areas delimited should be linked with the areas in the Socioeconomic Data file. |
| SocioeconomicData | CSV with socioeconomic data per area within the city. The mandatory fields are “zone”; "1: woningen"; "9: arbeidspl\_totaal" |
| ParcelNodes | Shape with the location of the distribution nodes of the couriers |
| CourierMarketShares | Couriers table with market shares |
| ExternalZones | Coordinates of the external (super) zones that lie outside of the study area. |

### Outputs

The outputs of the parcel generation module are described in Table 2.

Table 2 Parcel Generation– Outputs

|  |  |
| --- | --- |
| Outputs | Description |
| ParcelDemand.csv | Parcel list with Origin, Destination and Courier |
| KPIs.json | File containing key KPIs estimated in the model |
| Log.txt | Logs of the run |

## Paths structure

The directory where the model is located has the following structure:

── Root

├── Input Folder

│

└── Output folder

# Model Description

This section describes the different files and scripts present in the model

|  |  |  |
| --- | --- | --- |
| File name | Location | Description |
| Parcel\_Generation.py | Root | Main script |
| \_\_functions\_\_.py | Root | External functions |
| requirements.txt | Root | Python packages required |

# Instructions to run the model

## Command line execution of the model

### Instructions and commands

The instruction to install the packages needed:

* pip install -r requirements.txt

The instruction to run the model

python3 Parcel\_Generation.py Label Input Output Params\_ParcelGen.txt TimeSkimMatrix.mtx DistanceSkimMatrix.mtx StudyAreaShapefile.shp SocioeconomicData.csv ParcelNodes.shp CourierMarketShares.csv

### Arguments

The arguments in the instructions to run the model are:

Table 3 Parcel Generation– Inputs

|  |  |
| --- | --- |
| Arg[0] | Script name |
| Arg[1] | Lable (name of scenario) |
| Arg[2] | Input folder name |
| Arg[3] | Output folder name |
| Arg[4] | Parameter text file |
| Arg[5] | Time skim matrx |
| Arg[6] | Distance skim matrix |
| Arg[7] | Zone shapefile |
| Arg[8] | Socioec data |
| Arg[9] | Parcel Nodes shapefile |
| Arg[10] | Courier shares |

## Requirements

### Testing requirements

pip install -r requirements.txt

### Input folder (Arg[2])

Folder 1( e.g. Input)

Which files are in the folder, what do they do, what type of file (csv, etc) what requirements do they have (table with headers, what each row represent, etC)

Table 4 Parcel Generation– Inputs

|  |  |  |
| --- | --- | --- |
| Inputs | Type | Requirements |
| Parameters.txt | .txt |  |
| TimeSkimMatrix.mtx | .mtm | Id of areas ordered increasingly |
| DistanceSkimMatrix.mtx | .mtm | Id of areas ordered increasingly |
| StudyAreaShapefile | .shp | .dbf with the same name |
| SocioeconomicData | .csv | Required cols:  “zone”;  "1: woningen";  "9: arbeidspl\_totaal" |
| ParcelNodes | .shp | .dbf with the same name |
| CourierMarketShares | .csv | Required cols:  “CEP”;  "ShareTotal"; |
| SupCoordinatesID | .csv | Required cols:  “COROP”;  " Xcoor ";  “Ycoor”;  " AREANR "; |