

Caseformulering (company CH)

The mathematics and computer science study association Christiaan Huygens organizes multiple events during the year, giving students opportunities to make new social contacts and to learn and improve their skills. The MaCHazine committee writes reports about these events and assembles them with interesting articles into a beautiful maCHazine. Every quarter the maCHazines are printed and distributed to the members of the association. Christiaan Huygens wants to know the optimal planning on how to deliver the maCHazines such that all the participants of this hackathon receive this report in the future.

In this case, the definition of an optimal planning is free for interpretation. One could think of fastest delivery, most sustainable, lowest cost etcetera. In order to find a solution you will need the distances between addresses in some way. You can choose yourself how you would like to implement this. To get you started we have the following suggestions. You could use an API that uses Google Maps, the links of these API's can be found here:

Another possibility is to use the example address list and the respective distance matrix provided.

The input of your optimization problem will be a list of addresses. You can use a distribution point (for example Mekelweg 4) as input, however you can also optimize the place of the distribution centre. You have access to some resources, such as transporting vehicles, people delivering and a budget. More information about the requirements of these resources can be found below.

As a result of your optimization algorithm an answer in the command line is sufficient. However, you will receive more points if you visualize your solution in an appealing way. Think of a dashboard where you can fill in the amount of vehicles available or a map that visualizes your solution. You are free to use your own imagination and creativity to think of more features to implement and/or to make the delivery problem (and solution) more realistic.

Requirements

Inputs:

- Car
 - Amount of cars available
 - (Average) speed
- Bike
 - Amount of bikes available
 - Capacity of bike
 - (Average) speed (per bike or all bikes)
- People
 - Amount of people available
 - (Average) speed
- Distribution point
 - Address
- Delivery address
 - Address

Outputs:

An optimal solution, taking all factors into account. Could be in the form of visualization or just a data structure. In this solution either all maCHazines must be delivered or every delivery address has received a maCHazine.

Address list

For the distribution point you could use: Mekelweg 4, 2628 CD Delft

Example locations for deliveries:

- Markt 87, 2611 GS Delft
- Olof Palmestraat 1, 2616 LN Delft
- Troelstralaan 71, 2624 ET Delft
- Van Leeuwenhoeksingel 42A, 2611 AC Delft
- Kleveringweg 2, 2616 LZ Delft
- Schieweg 15L, 2627 AN Delft
- Westeinde 2A, 2275 AD Voorburg
- Herenstraat 101, 2271 CC Voorburg
- Haags Kwartier 55, 2491 BM Den Haag
- Spui 70, 2511 BT Den Haag
- Liguster 202, 2262 AC Leidschendam

Generating a distance matrix

To generate your own distance matrix you could use the provided python script. To run the script you would need to install matplotlib. The script will generate a distance matrix using 10 points. You can alter the code to increase or decrease the amount of points.

API List

<https://www.mapbox.com/>

<https://developer.apple.com/maps/>

<https://developer.here.com/>

<https://blog.hubspot.com/website/free-api-maps>