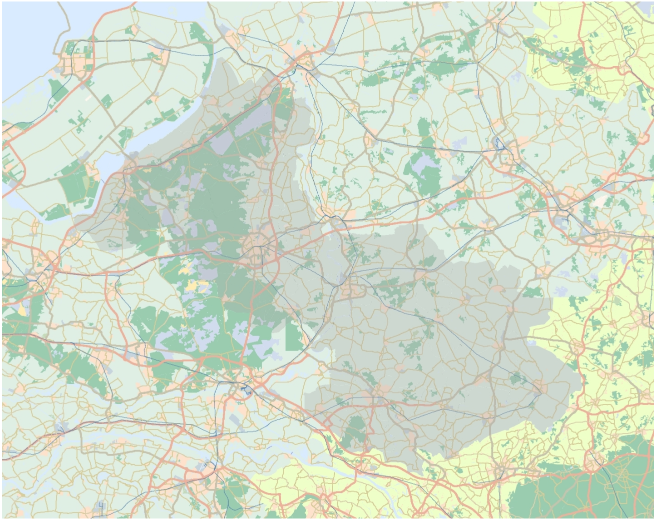
The Shortest Path Problem in Emergency Vehicle Routing

Henk Post

connexxion

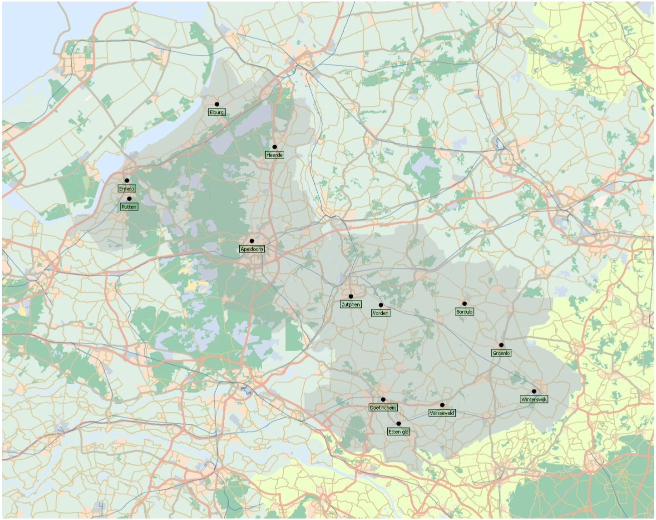
June 26^{th}, 2014

The Shortest Path Problem in Emergency Vehicle Routing



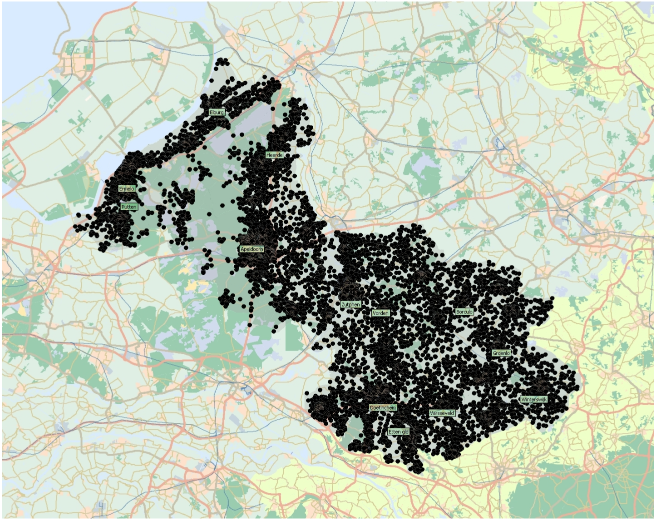
21 Municipalities

The Shortest Path Problem in Emergency Vehicle Routing



21 Municipalities, 13 depots

The Shortest Path Problem in Emergency Vehicle Routing



21 Municipalities, 13 depots, 24167 demand points.

Let r be a given time radius.

A demand point is covered by a depot if the driving time from that depot to the demand point is less than r.

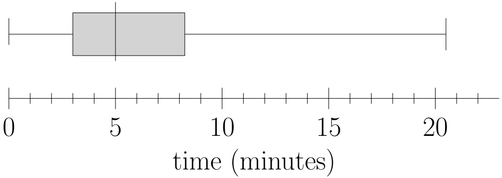
Typical questions:

- Which demand points are covered by at least one depot?
- Which demand points are covered by at least two depots?
- What is the minimum driving time to any demand point?
- What is the route from one location to another?
- How to draw the area covered from any depot?

Coverage

	#demand points	%
Total	24,167	100.0%
Uncovered	1,160	4.8%
Covered	23,007	95.2%
Double covered	13,999	57.9%

Driving times



The Shortest Path Problem in Emergency Vehicle Routing

Introduction
What we need
The SPP
Preprocessing
Bidirectional search
Concave hull
Moving car
Sources

Algorithm 1: DIJKSTRA $(\mathcal{G} = (\mathcal{V}, A), s)$

$d_v \leftarrow \infty \, \forall v \in \mathcal{V} \setminus \{s\}$, $d_s \leftarrow 0$, $\mathcal{Q} \leftarrow \{s\}$
repeat
    Select a pivot node $v \leftarrow \arg\min\{d_v | v \in \mathcal{Q}\}$
    $\mathcal{Q} \leftarrow \mathcal{Q} - \{v\}$
    foreach $(v, w) \in \mathcal{A}$ **do**
        if $d_v + c_{vw} < d_w$ **then**
            $d_w \leftarrow d_v + c_{vw}$
            if $w \notin \mathcal{Q}$ **then**
                $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{w\}$
until $\mathcal{Q} = \emptyset$

Coverage

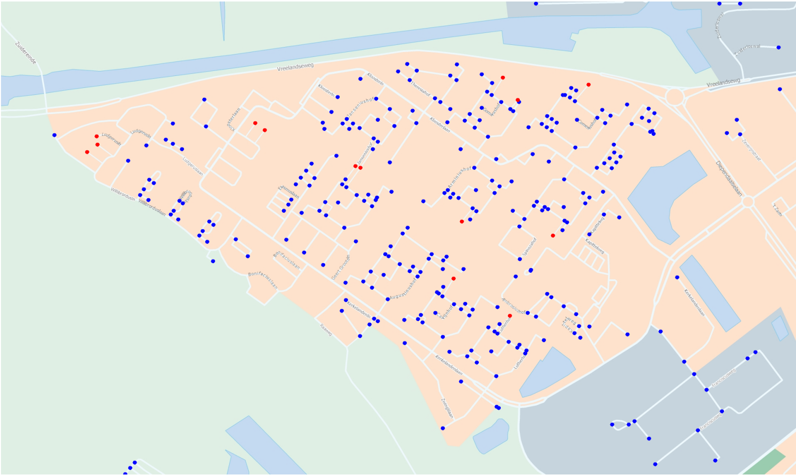
Determining the coverage of each demand point can be done by growing a partial shortest path tree from each depot. Since r is usually small, this takes only a small amount of time.

Driving times

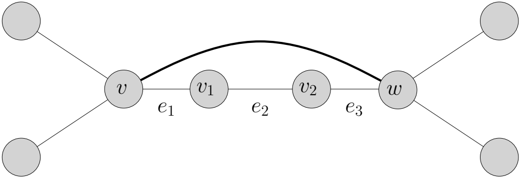
The shortest driving time to each demand point can be calculated by an updated initialization step. All nodes v that represent a depot are initialized with a zero label $d_v \leftarrow 0$. Furthermore, we stop the algorithm as soon as all demand points are chosen as a pivot node.

The Shortest Path Problem in Emergency Vehicle Routing

Quarter reduction (find the largest biconnected part)



Add shortcuts for ‘degree 2 paths’



Core nodes

A node that is not a ‘quarter-node’ and not a ‘degree 2 node’ is a *core* node.

Partitioning of the node set $\mathcal{V}$

Let $\mathcal{V}^0 \subset \mathcal{V}$ be the set of core nodes. A partition $\mathcal{V}^0, \mathcal{V}^1, \dots \mathcal{V}^K$ of $\mathcal{V}$ is determined, where $v, w \in \mathcal{V}^k (1 \leq k \leq K)$ if and only if there exists a path from v to w that contains only non-core nodes.

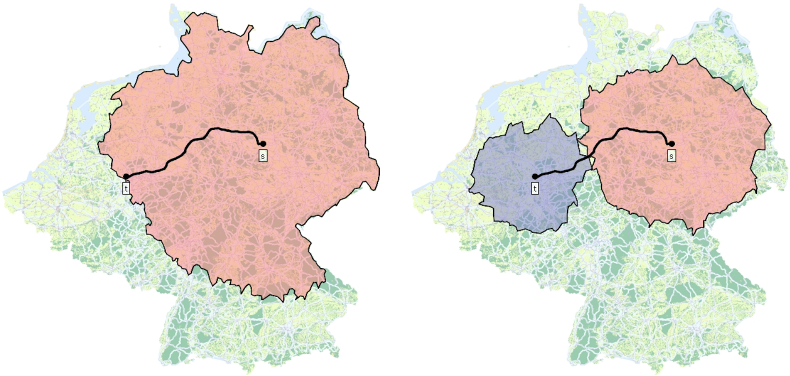
Introduction
What we need
The SPP
Preprocessing
Bidirectional search
Concave hull
Moving car
Sources

Roadmap of Germany, The Netherlands, Belgium and Luxembourg

Action	Entity	Amount	%	CPU
Input	$\|\mathcal{V}\|$	8,678,011	100	
Biconnect	Quarter nodes	2,503,779	29	11.8
BypassDegree2	Degree 2 nodes	3,516,724	40	4.1
	Core nodes $\mathcal{V}^0$	2,657,508	31	
Partitioning	Number of subsets	1,672,076		3.8
	$\max_{1\leq k\leq K} \|\mathcal{V}^k\|$	2,127		

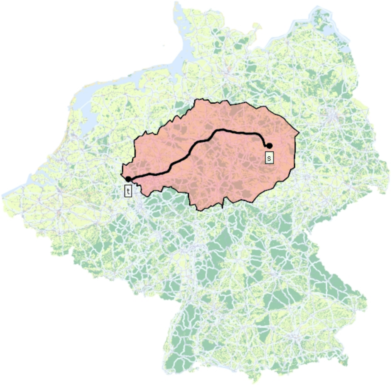
The Shortest Path Problem in Emergency Vehicle Routing

Bidirectional search



Guided search A*

A heuristic estimator h_v is provided. The pivot node is now selected according to $v \leftarrow \arg\min\{d_v + h_v | v \in \mathcal{Q}\}$.



Let $\delta(v, w)$ be an underestimate for the distance from v to w.

Balanced approach

Forward search uses the estimate $h_v \leftarrow \frac{1}{2}\delta(v, t) - \frac{1}{2}\delta(s, v)$
Backward search uses the estimate $h_v \leftarrow \frac{1}{2}\delta(s, v) - \frac{1}{2}\delta(v, t)$

As soon as both searches meet each other, a simple postprocessing step can be used to determine the shortest s-t distance.

Symmetric approach

Forward search uses the estimate $h_v \leftarrow \delta(v, t)$
Backward search uses the estimate $h_v \leftarrow \delta(s, v)$

A new stop condition is needed to be competitive with the balanced approach.

Introduction
What we need
The SPP
Preprocessing
Bidirectional search
Concave hull
Moving car
Sources

Scalar projections

Let $\mathbf{x}_v$ be the coordinates of node v.

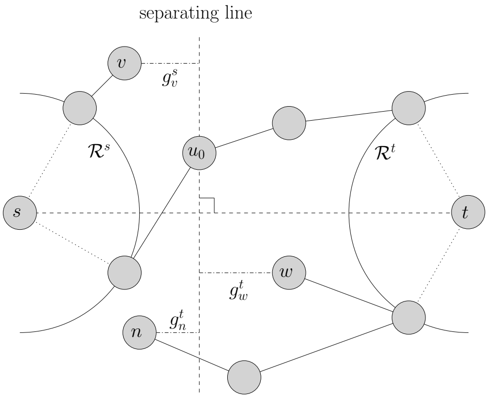
$$g_v^s = \frac{(\mathbf{x}_t - \mathbf{x}_s)^{\mathrm{T}}(\mathbf{x}_{u_0} - \mathbf{x}_v)}{||\mathbf{x}_t - \mathbf{x}_s||_2}, \quad v \in \mathcal{V}$$

$$g_w^t = \frac{(\mathbf{x}_s - \mathbf{x}_t)^{\mathrm{T}}(\mathbf{x}_{u_0} - \mathbf{x}_w)}{||\mathbf{x}_s - \mathbf{x}_t||_2}, \quad w \in \mathcal{V}$$

Properties

- $g_v^s + g_w^t = \frac{(\mathbf{x}_t - \mathbf{x}_s)^{\mathrm{T}}(\mathbf{x}_w - \mathbf{x}_v)}{||\mathbf{x}_t - \mathbf{x}_s||_2}$
- $g_v^s + g_w^t \leq ||\mathbf{x}_w - \mathbf{x}_v||_2$

The Shortest Path Problem in Emergency Vehicle Routing



We assume $c_{vw} \geq \alpha||\mathbf{x}_w - \mathbf{x}_v||_2$, for some $\alpha > 0$.

As soon as a meeting node u_0 is settled from both sides we reject a new pivot node in the forward search if:

$$d_v^s + \alpha g_v^s \geq d_{u_0}^s$$

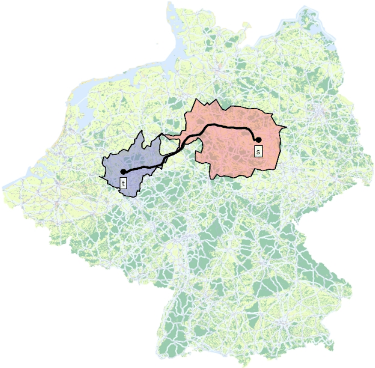
Similarly, a pivot node in the backward search is rejected if:

$$d_v^t + \alpha g_v^t < d_{u_0}^t.$$

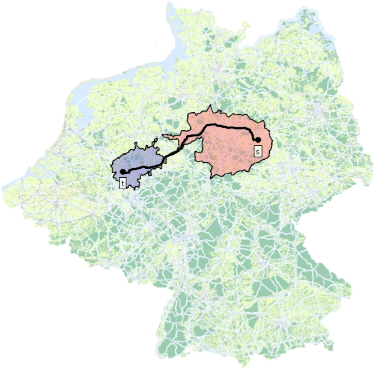
In our thesis, we present a proof that a node on a shortest path will not be rejected from both sides.

The Shortest Path Problem in Emergency Vehicle Routing

Balanced



Symmetric



The Shortest Path Problem in Emergency Vehicle Routing

Algorithm 2: Concave hull (Rosén et al.)

Require: List A with edges for the convex hull

Sort list A on the length of the edges

$B \leftarrow \emptyset$

while *list A not empty* **do**

    Select the longest edge e from list A

    Remove edge e from list A

    if *e is suitable to be split in e_2 and e_3* **then**

        Add e_2 and e_3 to list A

    else

        Add e to list B

Ensure: List B contains the edges of a concave hull

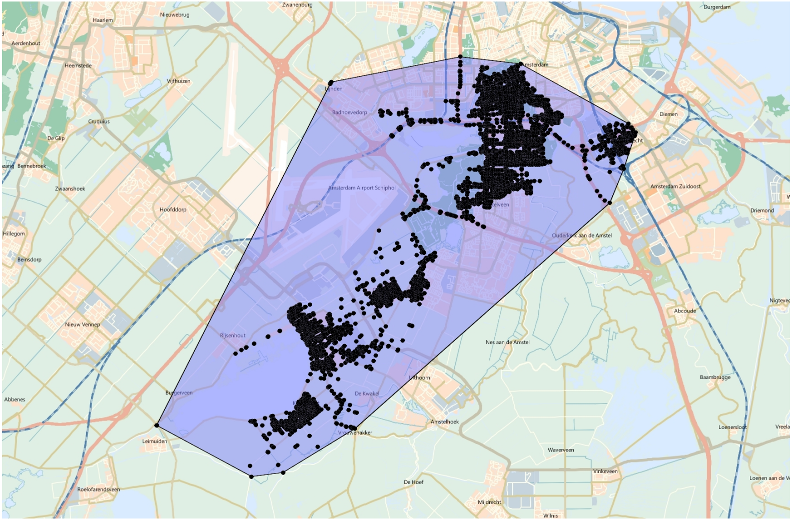
When is it suitable to split an edge e?

- It depends on the length of e (global or local check)
- All points still remain inside the hull
- The hull has to be a simple polygon

Implementation

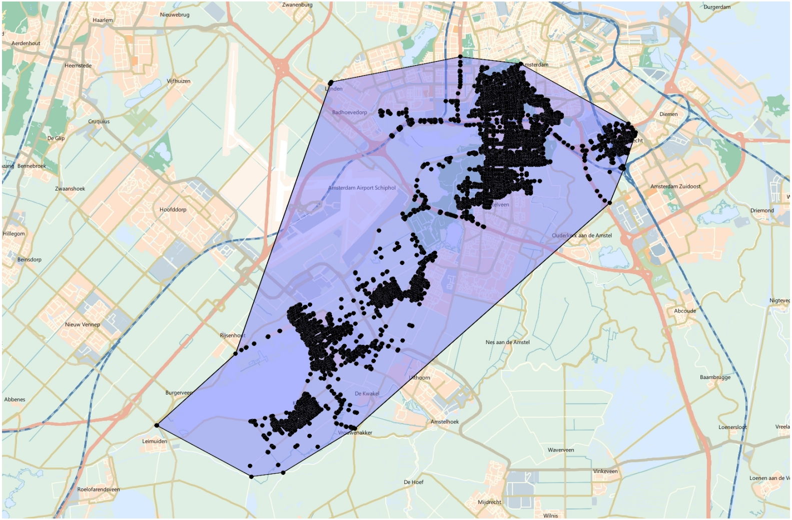
Place all points in a grid. In this way:

- The time needed for determining a new boundary node can be reduced.
- The maximum 'suitable' edge length can be determined locally.

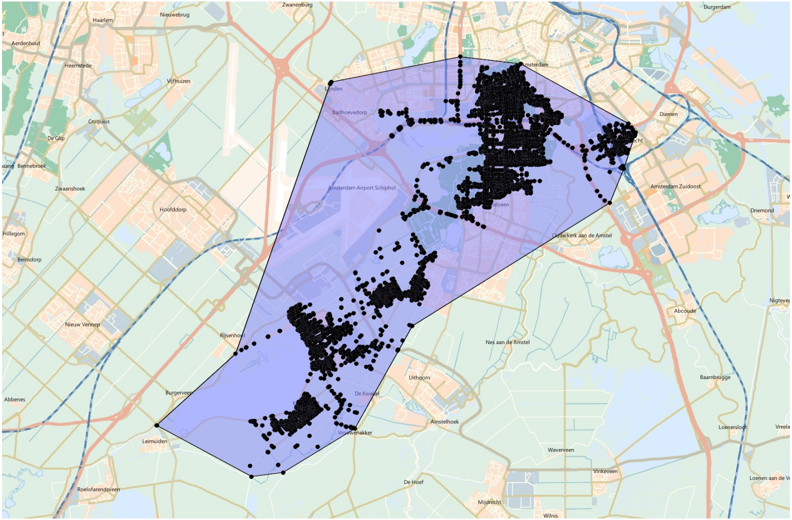


Zwanenburg
Durgerdam
Haarlem
Nieuwebrug
Aerdenhout
Heemstede
Vijfhuizen
Badhoevedorp
Diemen
Cruquius
De Glip
Bennebroek
Zwaanshoek
Amsterdam Airport Schiphol
Amsterdam Zuidoost
Hoofddorp
Driemond
Hillegom
Ouderkerk aan de Amstel
Beinsdorp
Abcoude
Nieuw Vennep
Rijsenhout
Nes aan de Amstel
Baambrugge
Abbenes
Burgerveen
Uithoorn
De Kwakel
Amstelhoek
Loenersloot
Leimuiden
Vrouwenakker
Waverveen
Vinkeveen
De Hoef
Roelofarendsveen
Mijdrecht
Wilnis

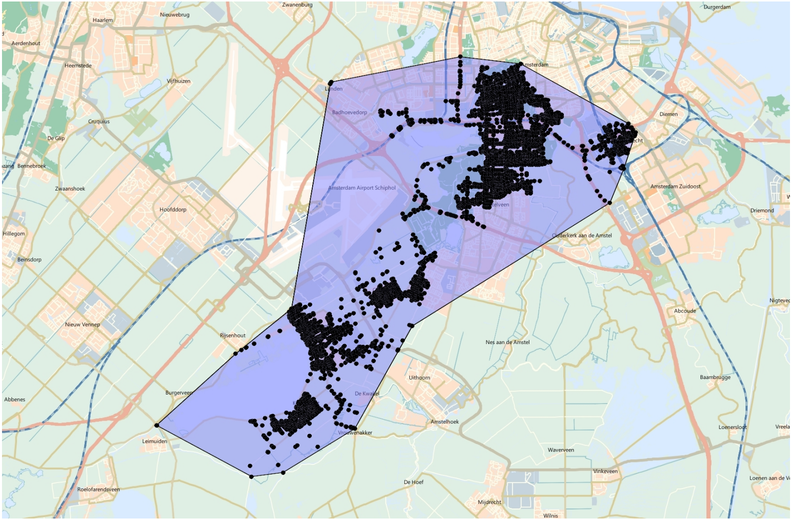
The Shortest Path Problem in Emergency Vehicle Routing



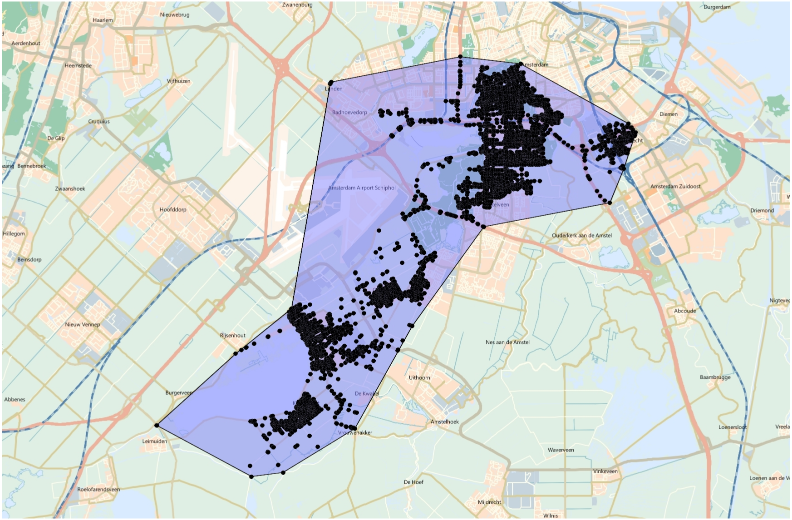
The Shortest Path Problem in Emergency Vehicle Routing



The Shortest Path Problem in Emergency Vehicle Routing



The Shortest Path Problem in Emergency Vehicle Routing



The Shortest Path Problem in Emergency Vehicle Routing

Introduction

What we need

The SPP

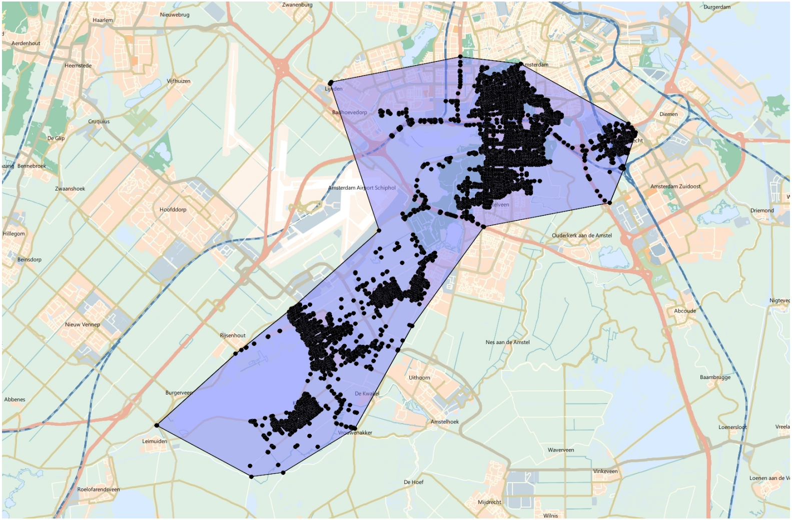
Preprocessing

Bidirectional search

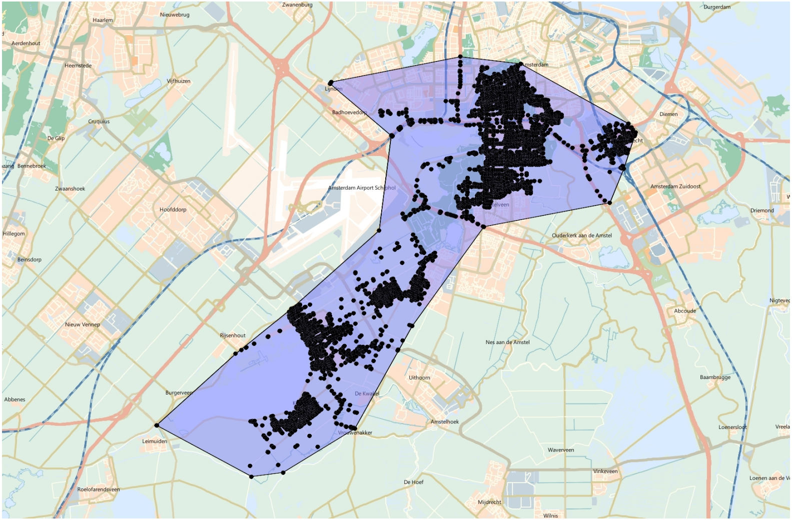
Concave hull

Moving car

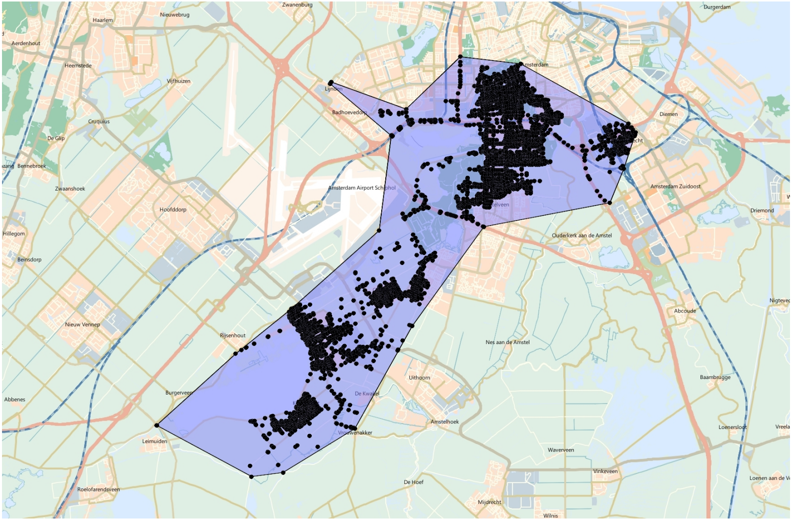
Sources



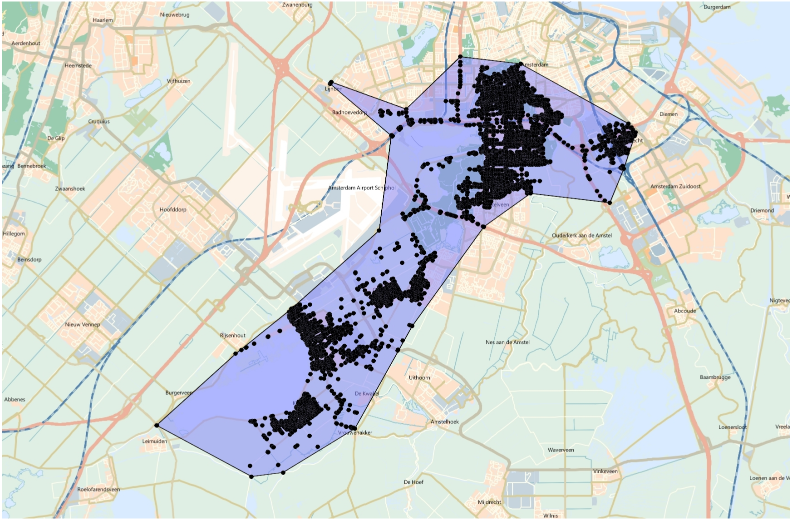
The Shortest Path Problem in Emergency Vehicle Routing



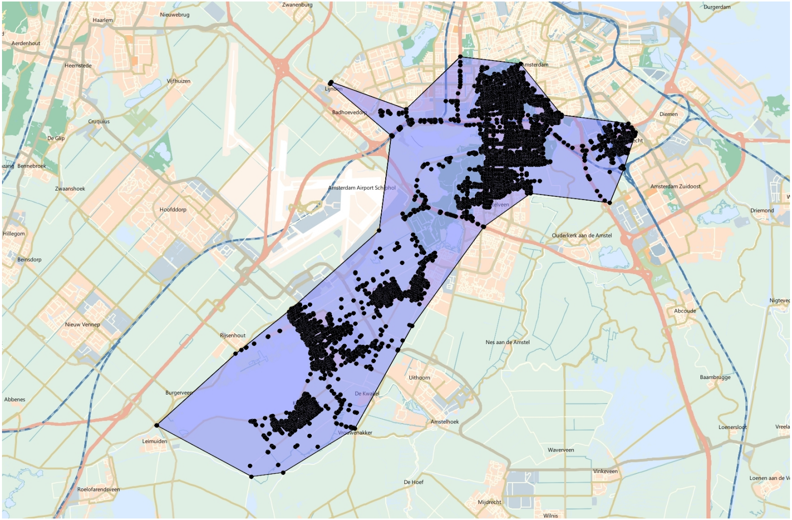
The Shortest Path Problem in Emergency Vehicle Routing



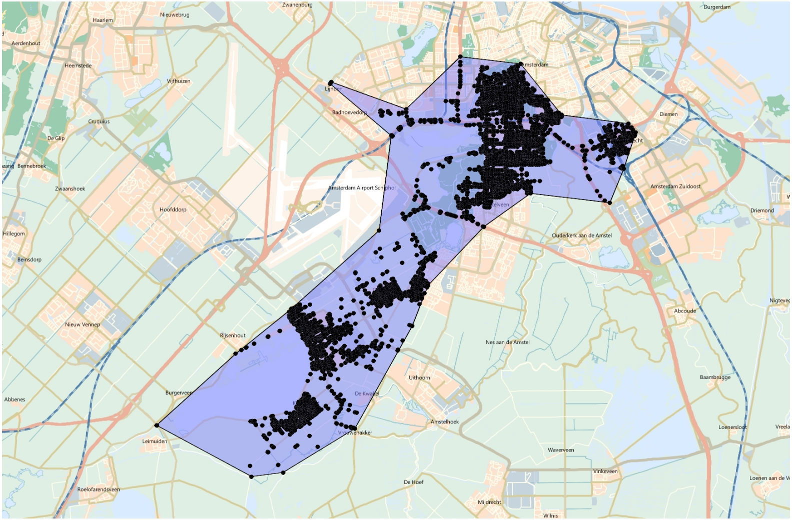
The Shortest Path Problem in Emergency Vehicle Routing



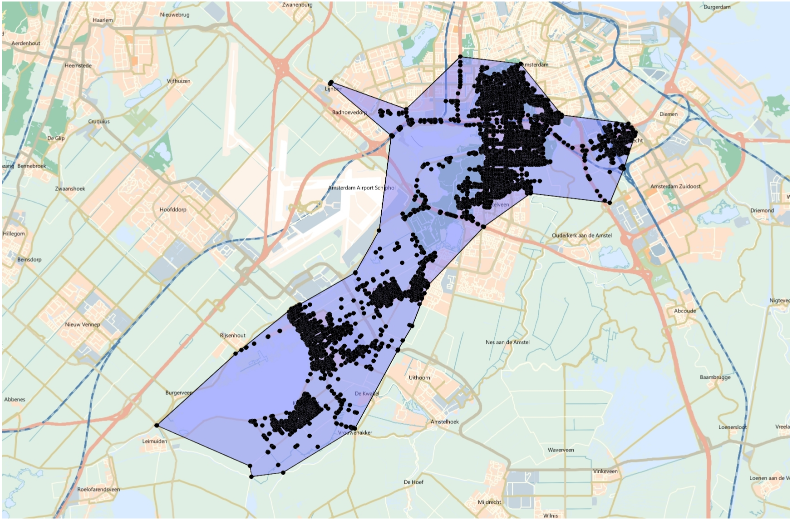
The Shortest Path Problem in Emergency Vehicle Routing



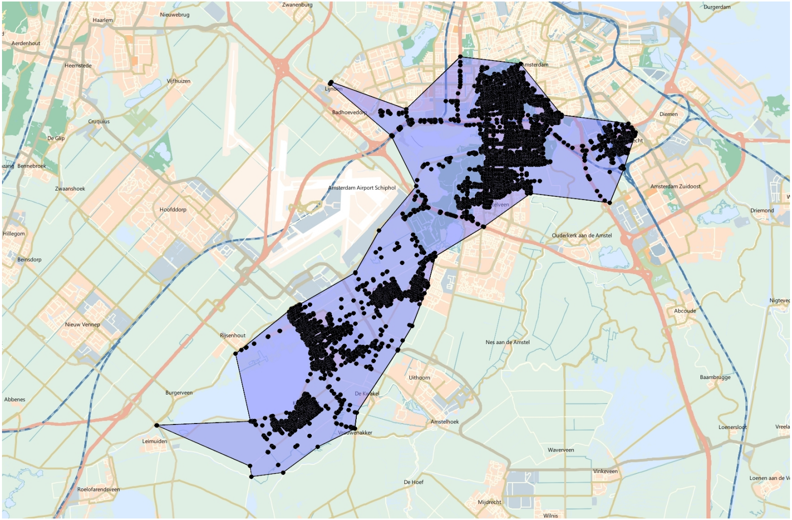
The Shortest Path Problem in Emergency Vehicle Routing



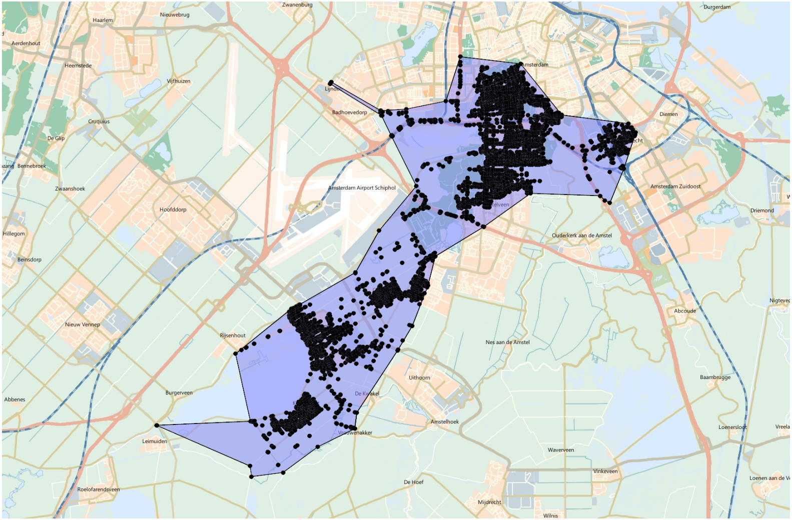
The Shortest Path Problem in Emergency Vehicle Routing



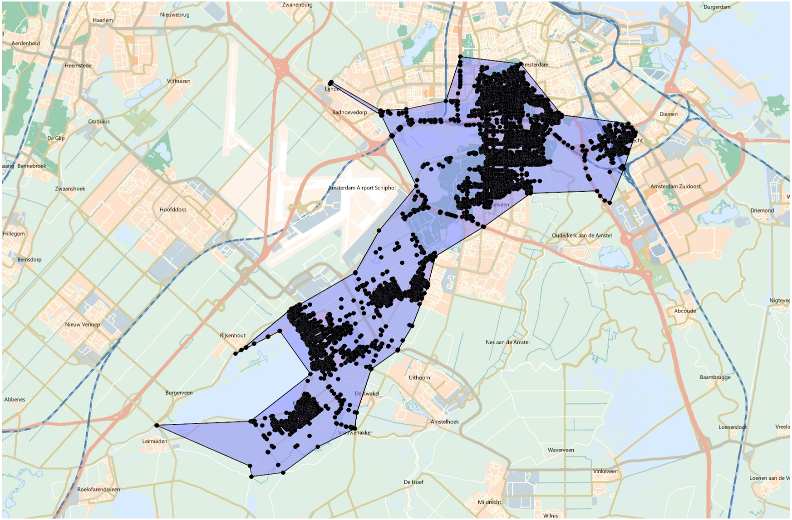
The Shortest Path Problem in Emergency Vehicle Routing



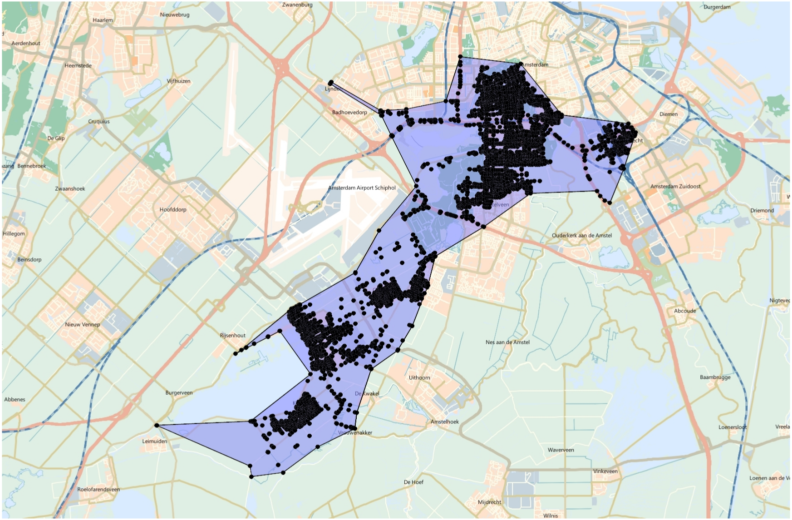
The Shortest Path Problem in Emergency Vehicle Routing



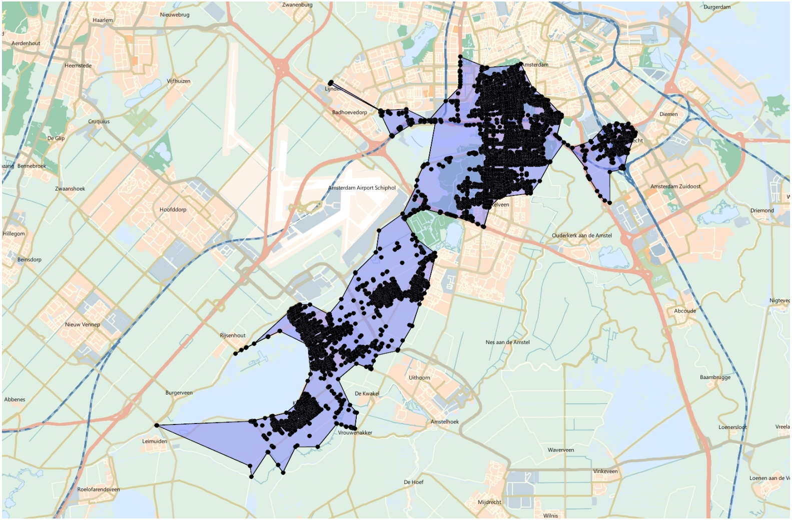
The Shortest Path Problem in Emergency Vehicle Routing



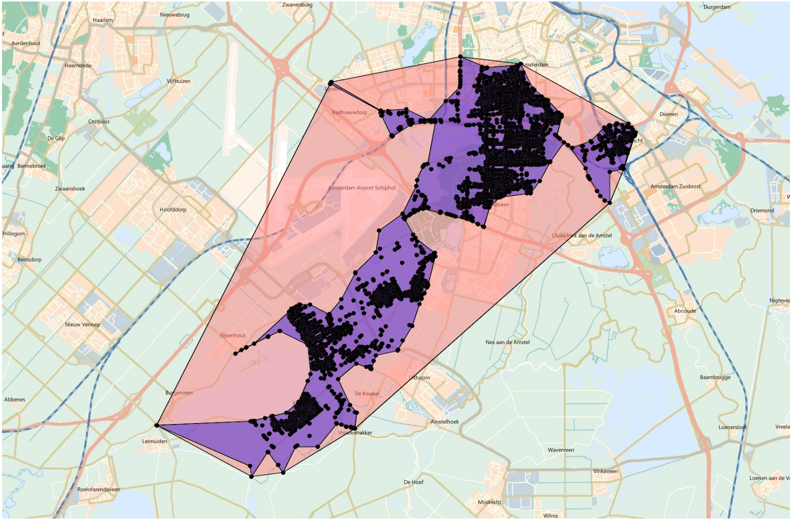
The Shortest Path Problem in Emergency Vehicle Routing



The Shortest Path Problem in Emergency Vehicle Routing



The Shortest Path Problem in Emergency Vehicle Routing



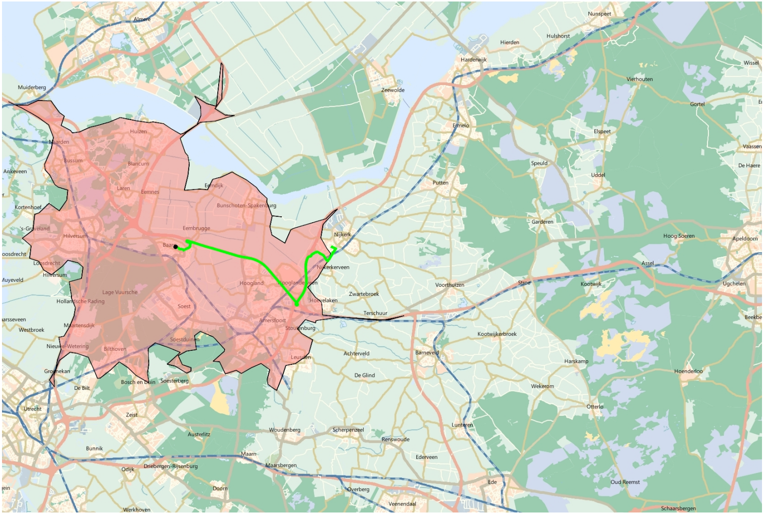
Trajectory
A *trajectory* consists of a polyline together with its length and duration. The average speed on a traject is considered to be constant.

Route
A *route* is a sequence of trajectories.

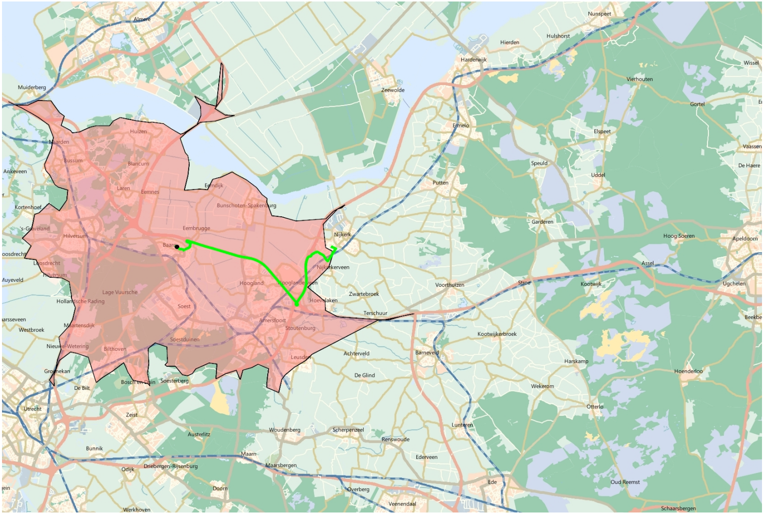
Usage
If the route of a vehicle is known, the expected position of the vehicle at a certain time t can be determined.

- This can be used in a real-time environment to determine whether or not certain recalculations are necessary.
- In an off-line environment (e.g. a simulation) one has a more realistic position to determine statistics.

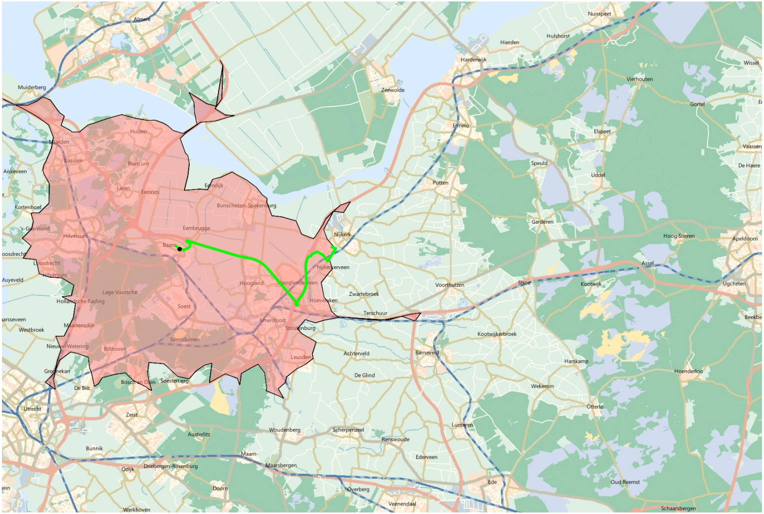
The Shortest Path Problem in Emergency Vehicle Routing



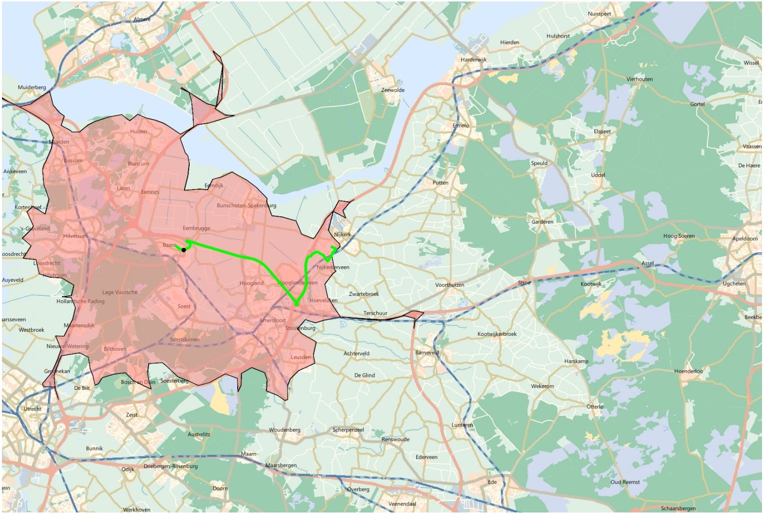
The Shortest Path Problem in Emergency Vehicle Routing



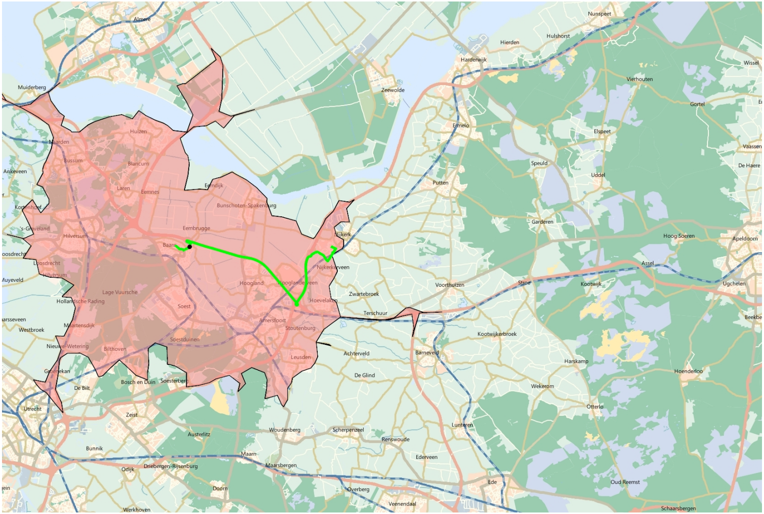
The Shortest Path Problem in Emergency Vehicle Routing



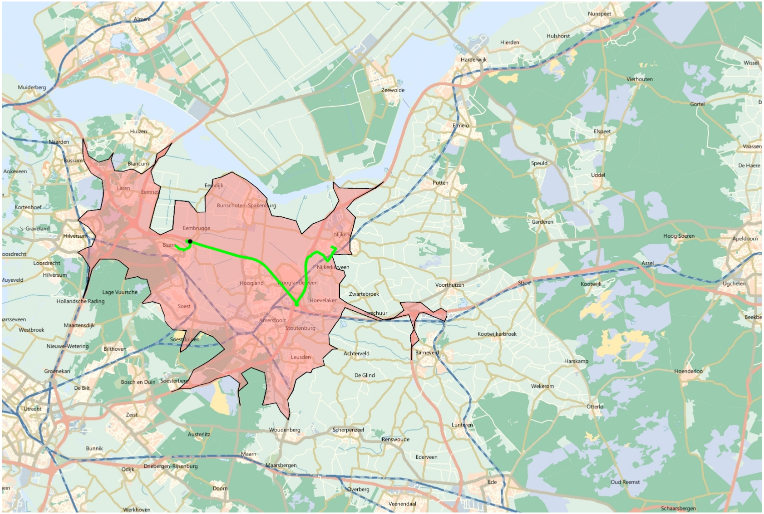
The Shortest Path Problem in Emergency Vehicle Routing



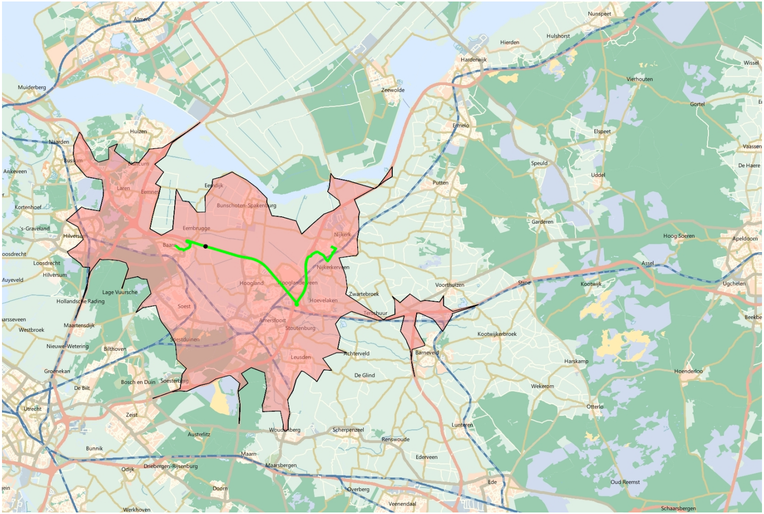
The Shortest Path Problem in Emergency Vehicle Routing



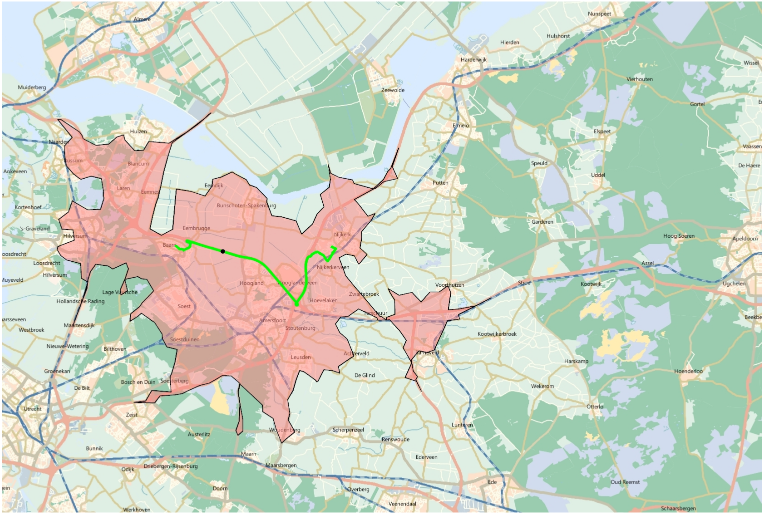
The Shortest Path Problem in Emergency Vehicle Routing



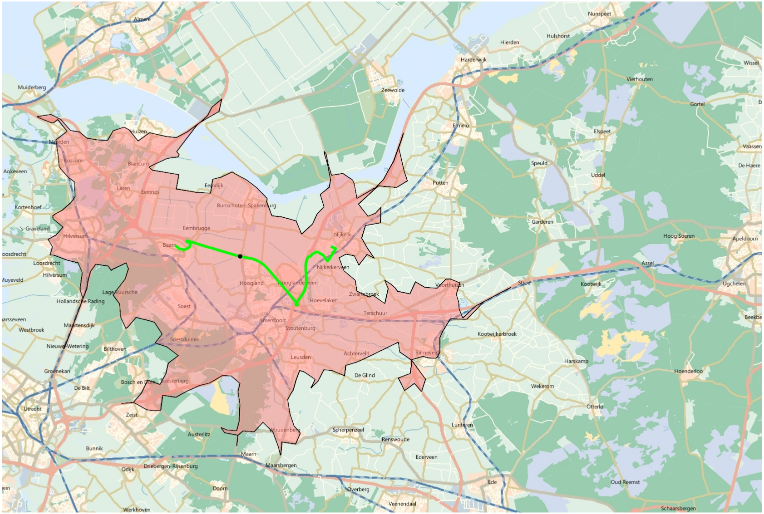
The Shortest Path Problem in Emergency Vehicle Routing



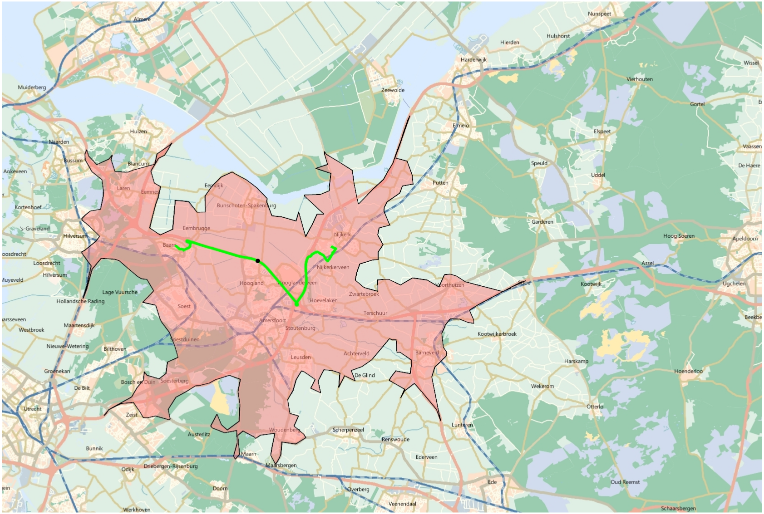
The Shortest Path Problem in Emergency Vehicle Routing



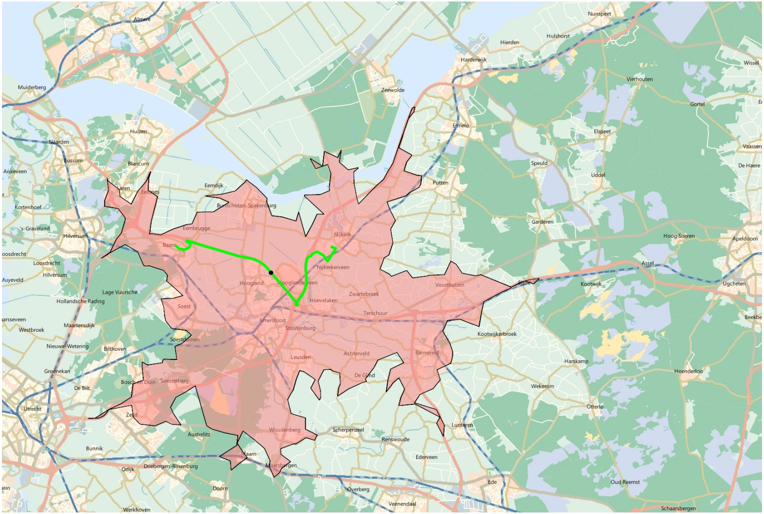
The Shortest Path Problem in Emergency Vehicle Routing



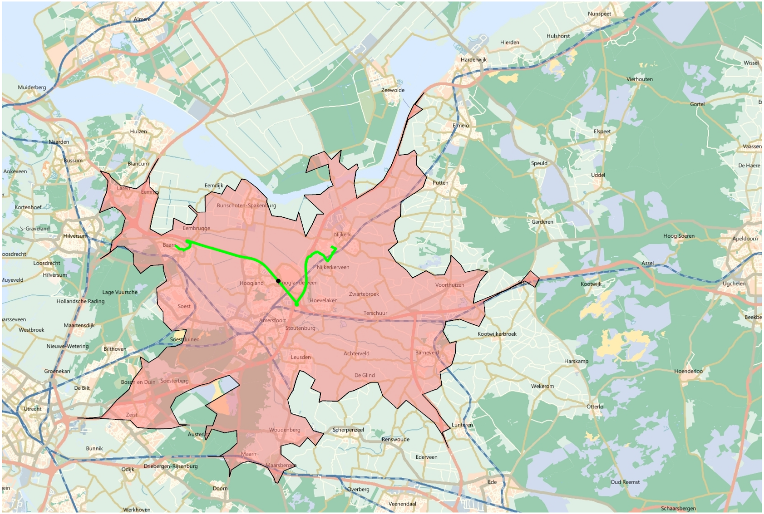
The Shortest Path Problem in Emergency Vehicle Routing



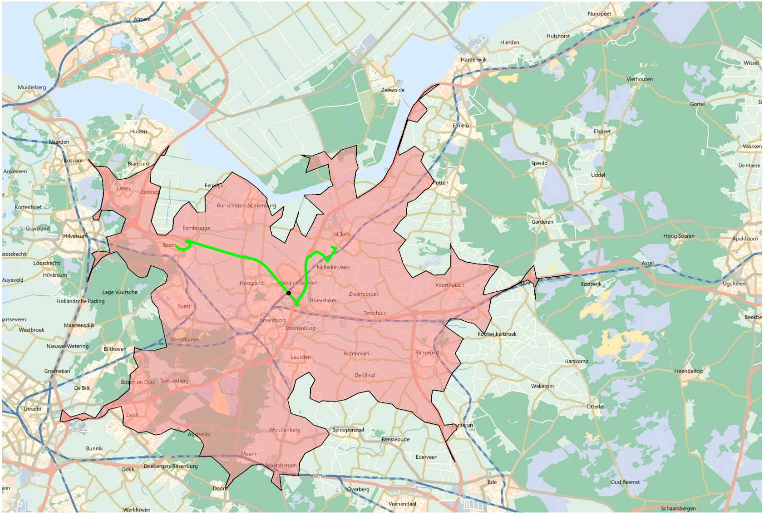
The Shortest Path Problem in Emergency Vehicle Routing



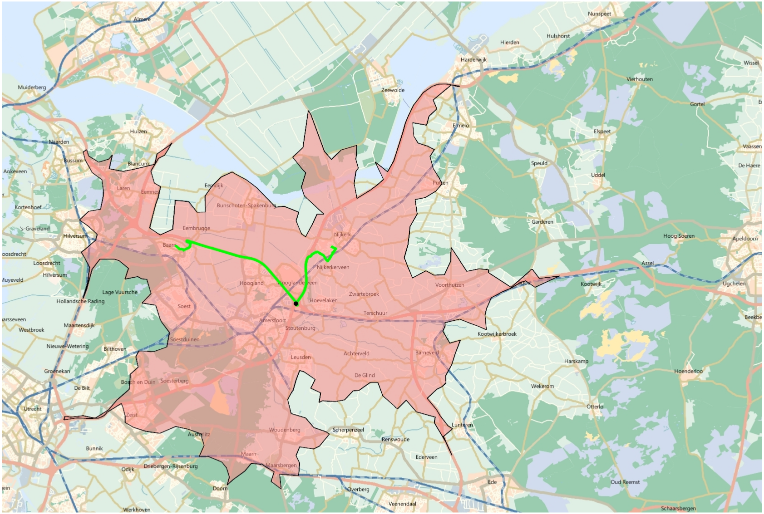
The Shortest Path Problem in Emergency Vehicle Routing



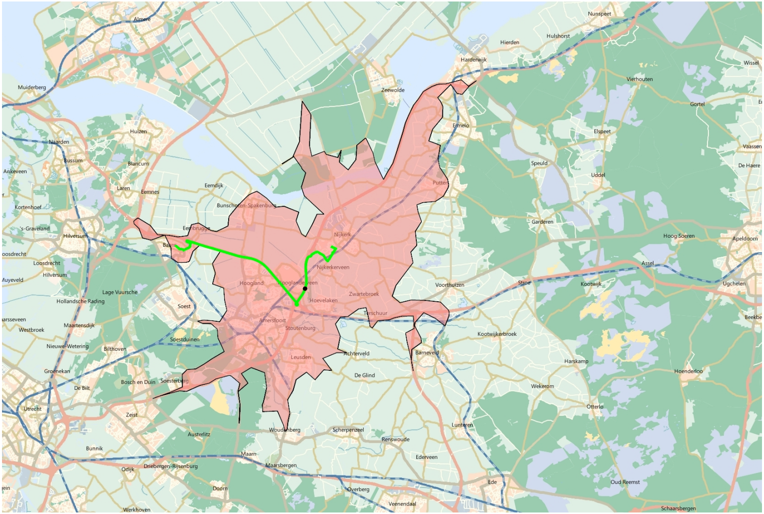
The Shortest Path Problem in Emergency Vehicle Routing



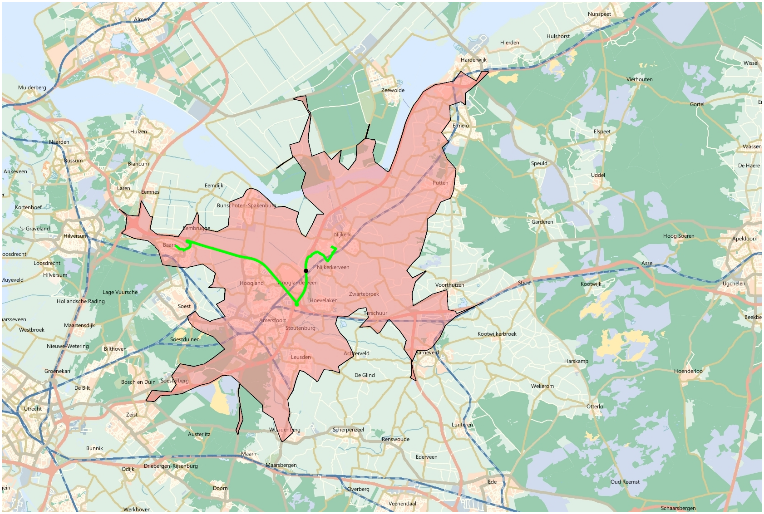
The Shortest Path Problem in Emergency Vehicle Routing



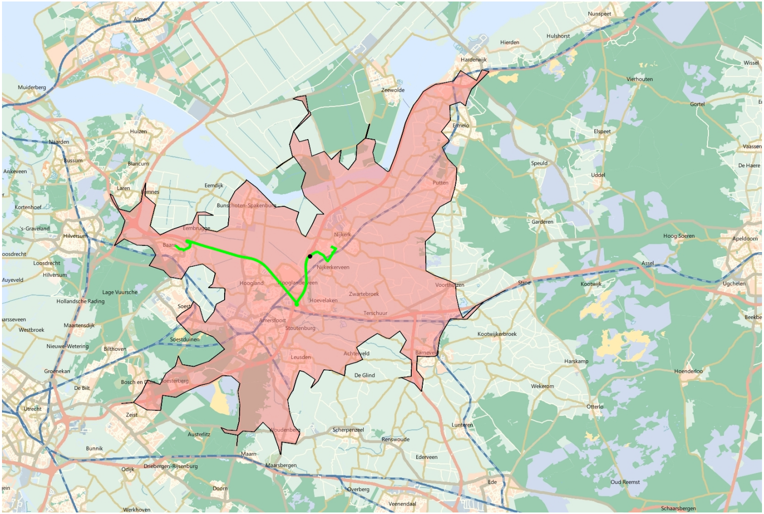
The Shortest Path Problem in Emergency Vehicle Routing



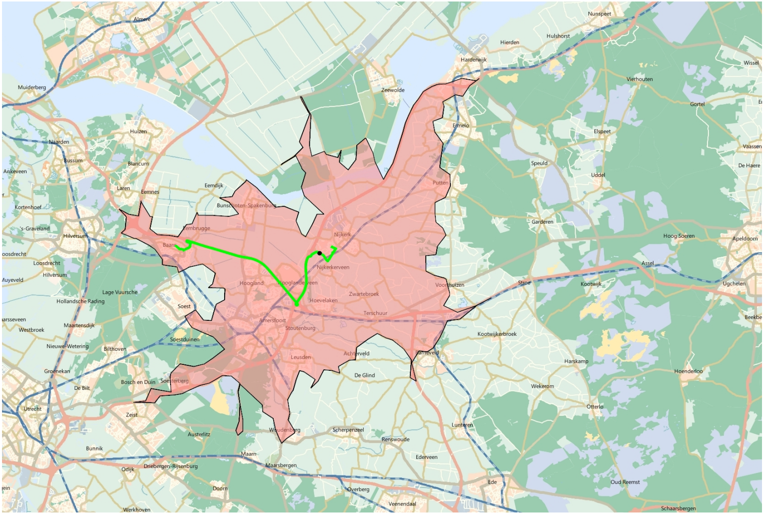
The Shortest Path Problem in Emergency Vehicle Routing



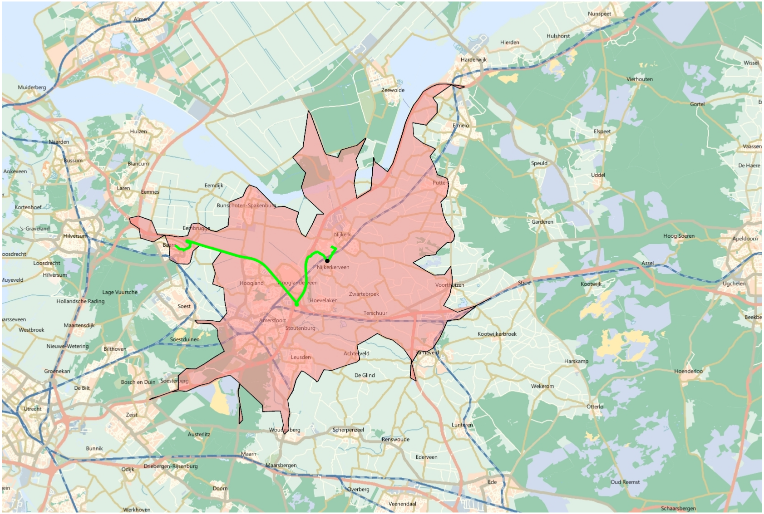
The Shortest Path Problem in Emergency Vehicle Routing



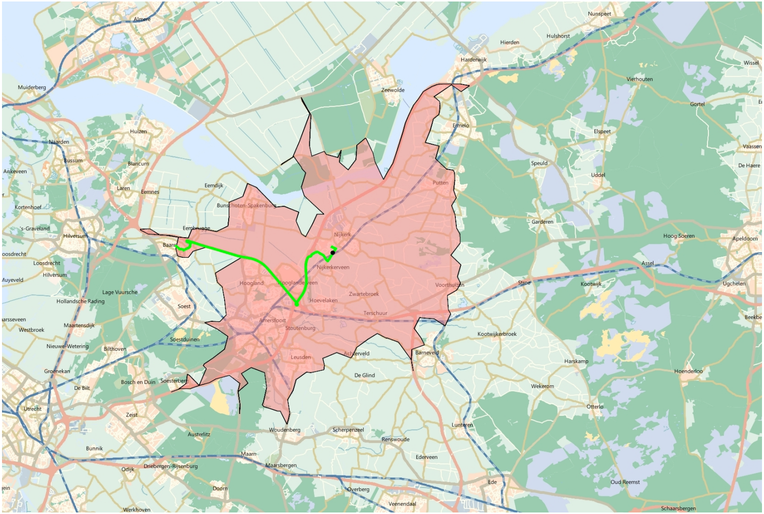
The Shortest Path Problem in Emergency Vehicle Routing



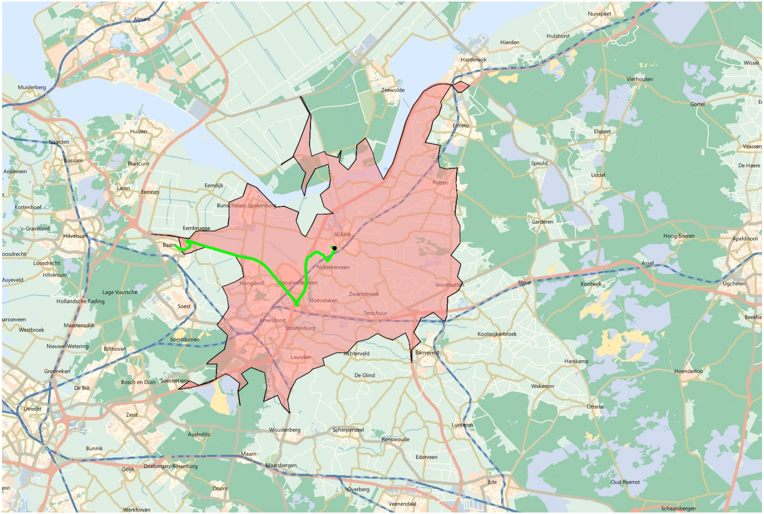
The Shortest Path Problem in Emergency Vehicle Routing



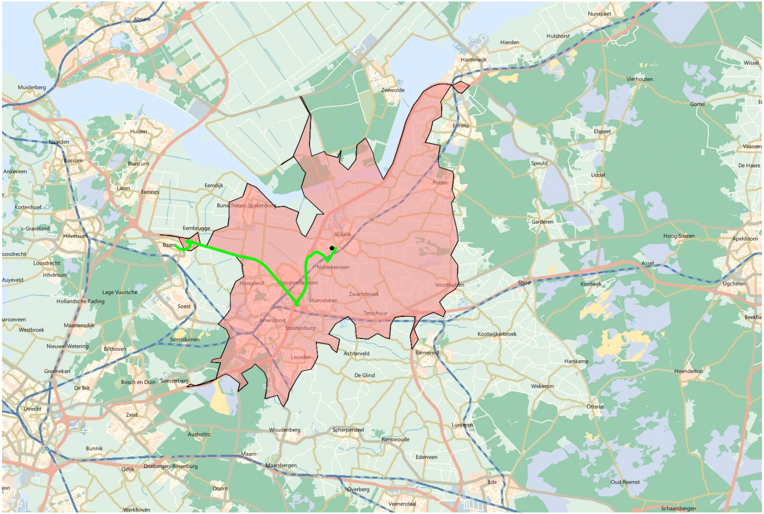
The Shortest Path Problem in Emergency Vehicle Routing



The Shortest Path Problem in Emergency Vehicle Routing



The Shortest Path Problem in Emergency Vehicle Routing



R.E. Tarjan
Depth-First Search and Linear Graph Algorithms
SIAM Journal on Computing 1 (2), 146–160 (1972)

T.-K. Whangbo
Efficient Modified Bidirectional A^* Algorithm for Optimal Route-Finding
In: New Trends in Applied Artificial Intelligence
Lecture Notes in Computer Science 4750, 344–353 (2007)

H.N. Post
The Shortest Path Problem on Real Road Networks
Theory, Algorithms and Computations
PhD Thesis, Delft University of Technology (2014)

Jin-Seo Park and Se-Jong Oh
A New Concave Hull Algorithm and Concaveness Measure for n-dimensional Datasets
J. of Information Science and Engineering 28, 587–600 (2012)

Emil Rosén, Emil Jansson and Michelle Brundin
Implementation of a Fast and Efficient Concave Hull Algorithm
Uppsala Universitet, Project report (2014)