Route optimization for gas leak detection

Loek Hendriksen

August 23, 2024

S2553635

Bsc Civil Engineering

University of Twente
Léon olde Scholtenhuis

Siers Infraconsult
Mechiel van Manen

Contents

1 Abreviations and definitions

ARP = arc routing problem
CARP = capacitated arc routing problem RPP = rural postman problem
CPP = Chinese postman problem
RPP = rural postman problem
connecting pipelines = pipelines that connect buildings to the gas network

2 Introduction

Every year, 5000 gas leaks are reported in the Netherlands (Netbeheer Nederland, 2022). This can be dangerous, since methane is flammable. Furthermore, gas leaks lead to a loss of gas and thus a loss of money. Therefor it is important to check the pipelines for gas leaks. These checks are carried out by gas leak seekers. Gas leak seekers are people that walk over the location of the underground pipelines with equipment that measures the methane content of the air. They have to walk over each pipeline of the gas network and in the ideal situation they only walk over each pipeline once.

At the moment, there are no tools or programs to determine the fastest route over the pipelines. This means the routes that the gas leak seekers walk have to be created by hand. At Siers Infraconsult for example, the gas leak seekers can see on a digital map where the pipelines are located. They then have to decide themselves what the fastest route is. While experienced gas leak seekers are able to walk close to optimal routes, this is more difficult for beginning gas leak seekers. They might walk longer distances then necessary, and generally spend more time planning their routes. It can take years for gas leak seekers to learn how to walk optimal routes. For this reason, it can be beneficial to create a tool that generates optimal routes.

3 Theoretical background

This research spans across two subjects; gas leak detection and route optimization. While the product will be used for gas leak detection, creating it is mainly a matter of route optimization. Therefor, some theoretical background will be provided for both of these subjects.

3.1 Gas leak detection

By the rules of the eurocodes (NEN7249, n.d.) gas pipelines should be checked at least once every five years. There are several methods to detect gas leaks. Some examples are: visual inspection technology, magnetic flux leakage, acoustic emission method, thermal imaging and the sniffer method. These methods are well described in the book 'Pipeline inspection and health monitoring technology: The key to integrity management' (Lu, Xu, Iseley, Peng, & Fu, 2023).

For visual inspection, often a self propelled vehicle traverses the pipelines and films them from the inside to identify cracks and damages. Advantages of this method are its low costs, simplicity and versatility. Furthermore, its is a relatively fast method and the results can easily be stored. Disadvantages of this method are that the outside of the pipelines can not be monitored and that it does not give quantitative results, so it relies on the judgment of experts.

The method of magnetic flux leakage uses a magnet that traverses through the pipeline and generates a magnetic field around the pipeline wall. A magnetic sensor then converts magnetic signals into electric signals, from which irregularities can be found. Magnetic flux leakage is a very accurate and the pipeline can still be used during the inspection. It is however, a costly method and the pipelines have to be very clean for optimal results.

Another method is the acoustic emission method. Gas leaks produce a sound, which can be detected by sound sensors placed on the pipeline. Computers can then use this information to determine the size and location of the gas leak. This method has the advantages that it has a high sensitivity and can accurately locate the location of a leak. It has the disadvantage that the ground above the pipeline has to be excavated and sometimes the pipeline has to be shut off.

Thermal imaging uses the fact that the soil that surrounds a pipeline will decrease in temperature when there is a gas leak. Vehicles, either on land or airborne, can travel over the pipelines whit an optical and infrared system to detect these changes in temperature and discover gas leaks. The advantage of this method is that it has a low operation time. A disadvantage is the high costs of the equipment and inspection. Additionally, if the surrounding soil is already at the same temperature as the pipeline, a gas leak will not lead to a change in temperature and can not be detected by this method. Therefor, this method can not be used during the winter.

The last method is the sniffer method. This method uses instruments that measure the gas concentration in the air. The location of a gas leak can accurately be located with this method and the costs are low. A downside is that wind can affect the results.

There are many different methods for gas leak detection. Because of the low costs and high precision, the sniffer method is the most used method in the Netherlands. Because the instruments have to be relatively close to the pipelines, this method requires gas leak seekers who walk over the pipelines.

3.2 Route optimization

While there is not much research done about route optimization for gas leak seekers, there are of course more situations in which route optimization is important. The case in which the shortest path over a given set of streets is searched for is known as an arc routing problem (ARP). This also occurs in for example waste collection, street gritting and mail delivery. A lot of research is done about the capacitated arc routing problem (CARP), where multiple vehicles with a limited capacity travel different routes. In this case, all the edges are traversed by one person. This is also known as the Chinese postman problem (CPP). If only a subset of edges need to be traversed, it is the rural postman problem (RPP). A perfect solution for this problem, no edge is traversed twice, is called an eulerian circuit (Vöcking et al., 2010). This is only possible when at most two nodes have an uneven number of edges. Often, this is not the case. That means that some edges have to be traversed twice to create a loop that traverses all edges.

An example of an algorithm to solve this is found on algorithms.discrete (ChinesePostmanMethod, 2015). As a first step, it finds the shortest path to connect each node with an uneven number of edges with another one. These paths are added as additional edges and ensure that an eulerian circuit is now possible. The eulerian circuit can be found with Hierholzer's algorithm (HierholzersAlgorithm, 2015). This algorithm starts at an arbitrary node and then traverses an unvisited edge to a new node and repeats this process until it is back at the starting node. If the cycle that is then formed traverses all edges, the eulerian circuit is complete. If not, a new cycle is made to traverse the remaining edges. These cycles together form an optimal path. When solving routing problems with programming, they are often formulated as an integer program (Hu & Kahng, 2016). The formulation then is:

$$min \sum_{k} \sum_{ij} c_{ij} x_{ij}^{k}$$

Where c_{ij} is the traveling cost, for example distance, of an edge and x_{ij}^{k} is a binary variable, either 1 or 0, for whether the edge is traversed or not.

When we apply this information on the project, it leads to the following conclusions; When only a network of pipelines is considered it can be solved as a CPP. When roads that can be used, but do not have to be traversed are also considered it can be solved as a RPP. There exist algorithms for these problems, these algorithms for route optimization, or optimization algorithms in general, are called heuristics.

3.3 Previous studies

In the past, a lot of research has been done about arc routing problems. One research, already conducted in 1984, uses a parallel insertion method to find an optimal schedule for school buses to pick up children (Chapleau, Ferland, Lapalme, & Rousseau, 1984). It combines two different insertion strategies. The first strategy takes the furthest unvisited edge and selects the route that lies closest to this edge to insert it in and repeats this process until all edges are inserted in a route. The second strategy takes the route with the most vacant spots and selects the edge that lies closest to this route. From their tests came the conclusion that this algorithm works well even for larger networks.

A CPP where some of the edges can only be traversed in one direction is called a directed Chinese postman problem or mixed Chinese postman problem (MCPP). These directed edges are called arcs. Nobert and Picard designed an algorithm to solve this MCPP (Nobert & Picard, 1996). They state that it can only be solved if all nodes and subsets of nodes are balanced in terms of incoming and outgoing arcs and use a quadratic function to find and solve unbalanced subsets. When it is necessary it than uses the blossom algorithm to make sure there are no odd nodes in the graph. The blossom algorithm finds the shortest path between two points. Than the algorithm finds the eulerian tour. Their tests revealed that the algorithm worked well for large networks and is relatively fast and flexible to use.

A more recent research is conducted by Tirkolaee et al. in 2022 (Tirkolaee, Goli, Gütmen, Weber, & Szwedzka, 2023). It uses a hybrid algorithm to find the optimal route for waste collection. Firstly, an initial route is formed with simple heuristics and then it is optimized by using two different optimization heuristics. The first heuristic uses simulated annealing, which is a local search algorithm. The second heuristic is an invasive weed optimization algorithm, which is a new algorithm for finding optimal solutions. The results of their tests shows that while their proposed algorithm does not generate the optimal route, it does generate a route which is close to optimal and it is significantly faster than an algorithm that finds an optimal route.

3.4 Summary

To summarize, in the Netherlands the most used method for gas leak detection is the sniffer method. This method requires people that walk over the pipelines, which is why a tool for route optimization is needed. Depending on whether only a network of pipelines is considered or also roads that do not have to be traversed, the problem can be solved as respectively a Chinese postman problem or an rural postman problem. Over the years, algorithms and heuristics are developed to solve these problems. There are however, some aspects of this project that can not be solved with these existing algorithms. The first aspect is that the network is depicted in a digital map. The second aspect is that there is multi-modality, gas leak seekers can use a bike to cover distances where no measuring is needed. The last aspect is that there is one large network, which they cover over a long time span, the routes have to be divided over the days. That is what this project will try to solve.

4 Research objective

4.1 Problem statement

Since gas leaks can lead to dangerous situations, pipelines have to be checked regularly. These checks are done by foot, which means they take a lot of time. Therefor it is important to follow an efficient route along the pipelines. At the moment there exists no method to generate these routes. This means that the gas leak seekers have to decide themselves how they walk over the pipelines. It can take years for gas leak seekers to learn how to plan a close to optimal route. This means that when they start, they will walk longer distances then necessary and spend more time planning routes. Finding the fastest route that covers the edges of a network is called an ARP or a postman problem. In the past, algorithms and heuristics are developed to solve these problems. There are however, some aspects of this project that can not be solved with these existing algorithms. The first aspect is that the network is depicted as a shape-file in a digital map. The second aspect is that there is multi-modality, gas leak seekers can use a bike to cover distances where no measuring is needed. The last aspect is that there is one large network, which they cover over a long time span, the routes have to be divided over the days. That is what this project will try to solve.

4.2 Research objective

The research objective is to design an algorithm that creates the optimal route for gas leak detection by applying existing route optimization heuristics and adjusting them for this situation as well as validating if it is useful.

4.3 Requirements

Apart from the research objective, there are some design requirements for the tool itself. These requirements are as followed;

1. The tool should generate a continuous route that covers all pipelines.
2. The generated route should be shorter than the routes currently walked by the gas leak seekers.
3. The route should be easy to follow.

5 Methodology

5.1 Defining the problem

To define the problem and to formulate the requirements for the tool, experts from Siers explained how the gas leak seekers currently work and what they expected from the tool. Furthermore, a gas leak seeker was accompanied for half a day to learn about how he worked and to ask him some additional questions. From this information, the requirements are formulated. The requirements are;

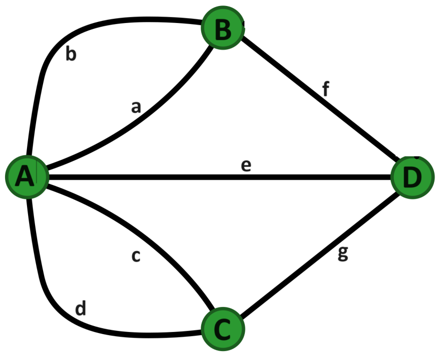
1. The tool should generate a continuous route that covers all pipelines.
2. The generated route should be shorter than the routes currently walked by the gas leak seekers.
3. The route should be easy to follow.

Furthermore, it was concluded that it was better to leave the connecting pipelines, which connect buildings to the gas network, out of the route. This is because they do not provide a challenge for the gas leak seekers and including them would increase the time needed to calculate the route significantly. Therefore the route only includes the main pipelines.

5.2 Design process

5.2.1 Postman problems package

While designing the algorithm for route optimization from the ground up would have a couple of advantages, it has the large disadvantage that it is not time efficient. Especially since there already exist algorithms to solve arc routing problems. This is why the choice is made to use an existing algorithm and adjust it. The python package 'postman problems' was chosen. This package allows to solve both the CPP or the RPP. Since this project also requires solving postman problems, this package is quite suitable. The disadvantage of this package is that it is not possible to make changes to it. This means among other things that it can not accommodate for dual modality, the use of a bike, within the route calculation itself. The postman problems package uses a table, which contains information about the lines as input. This table gives the start and endpoint of the line, the name of the line and its length. For the RPP, there is a fifth column, either one or zero, describing whether the line is required. In Figure 1, an example is given of a graph with its corresponding table. From this table a route is created, which is given as a list, with the order in which the edges are traversed. This list is also shown in Figure 1.



(a) Example graph

```
node1,node2,trail,distance
A,B,a,4
A,B,b,5
A,C,c,4
A,C,d,5
A,D,e,6
B,D,f,3
C,D,g,3
```

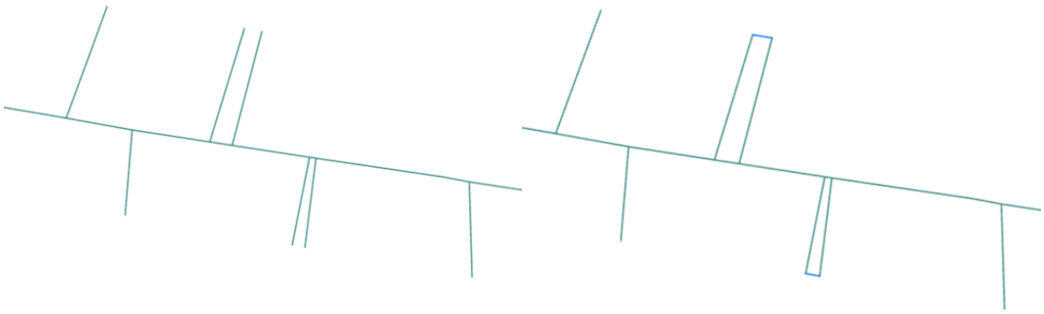
(b) Input table

```
('A', 'C', 1, {'trail': 'd', 'distance': 5, 'id': 3})
('C', 'D', 0, {'trail': 'g', 'distance': 3, 'id': 6, 'augmented': True})
('D', 'C', 0, {'trail': 'g', 'distance': 3, 'id': 6})
('C', 'A', 0, {'trail': 'c', 'distance': 4, 'id': 2})
('A', 'D', 0, {'trail': 'e', 'distance': 6, 'id': 4})
('D', 'B', 0, {'trail': 'f', 'distance': 3, 'id': 5})
('B', 'A', 0, {'trail': 'a', 'distance': 4, 'id': 0, 'augmented': True})
('A', 'B', 1, {'trail': 'b', 'distance': 5, 'id': 1})
('B', 'A', 0, {'trail': 'a', 'distance': 4, 'id': 0})
```

(c) Output table

Figure 1: Example graph with its input- and output table

The package contains two scripts that can create routes. One for Chinese postman problems and one for rural postman problems. The RPP can create shorter routes when additional lines are added. In Figure 2 you can see a pipeline with some branches. Since the route follows the lines, the route would traverse all branches twice when using the CPP for the first figure. When optional lines are added, the blue lines in figure two, a route can be created that traverses each branch only once, using the optional line to move from one branch to the other. This creates a shorter route. The addition of optional lines makes it a RPP. The advantage of a RPP is that it generates shorter routes. The disadvantage is that it requires drawing these optional lines in ArcGIS, which costs time.



(a) Without optional paths (b) With optional paths

Figure 2: Branches with and without optional paths

5.2.2 Creating the input table

The main challenge then is to create a table containing the right information and to convert the output of the postman problems package to a visible route. To create the input table a script is written that uses arcpy. This python module allows to make adjustments to a database in ArcGIS, the program which contains the pipelines.

5.2.3 Creating a visible route

Once the input table could be created and used to generated routes, there was one challenge remaining. The output of the route builder is a list with the order in which the lines are traversed. This is of course not a route that can be used by the gas leak seekers. Therefor the last script converts this list to a route in ArcGIS.

5.3 Verification and validation

An important part in the design process is verification and validation. These steps are crucial to ensure the final product works as is expected and meets the requirements. The verification focuses on the first two requirements;

1. The tool should generate a continuous route that covers all pipelines.
2. The generated route should be shorter than the routes currently walked by the gas leak seekers.

To test if these requirements are met, the following questions are asked;

1. Does the tool generate a continuous route?
2. Does the route cover all of the pipelines?
3. Is the route shorter than the route currently taken by gas leak seekers?

The first two questions can be answered by a visual inspection. It can be seen if the route is continuous and whether the route covers all of the pipelines or not.

To test if the route is shorter than the route currently taken by gas leak seekers, gps data of the gas leak seekers is used. This data can be used to create trails that show how they walked. Since the data is not very precise, these trails are converted to routes that follow the pipelines. Additionally, this allows to leave the connecting pipelines out of the comparison, since they will not be included in the routes made by the tool. Routes will be generated for different areas, both urban and rural, and the distance of these routes will be compared with the routes taken by the gas leak seekers.

Lastly, it is interesting to see the relation between the saved distance and the properties of the graph. The first property of a graph is its cycloramic number. This gives the maximum number of independent cycles in the network. A higher number means that a network is more connected, although this is an absolute number, so larger networks will have larger numbers. The second property is the alpha index. This gives the number of circuits compared to the maximum number of possible circuits. A higher number means that a network is more complex. The third property is the bèta index which is the number of edges divided by the number of nodes. Again, a higher number indicates a higher complexity. The last property is the gamma index, which is the number of edges divided by the highest possible number of edges.

The last requirement, The route should be easy to follow, can not be tested with calculations. It will therefor be tested by the experiences of a gas leak seeker. He will walk routes that are generated with the tool and then share his experiences in an interview. Based on the requirement, the following questions will be asked;

1. Was the route easy to follow?
2. Did it feel like this was the optimal route?
3. Is there a lot of difference between this route and how you would plan your route?
4. Is it a missing that the connecting pipelines are not a part of the route?
5. Is following and reading the route more or less work than planning it yourself?

The answers to these questions will be used to evaluate the way in which the route is indicated.

6 Final design

6.1 Route optimization tool

The route optimization tool uses the postman problems package. It uses the version that solves the RPP, because this version can still solve the CPP when no optional paths are added. The tool exists of three scripts. The first script, called 'Information from ArcGIS', generates a table with all the edges and their start- and end nodes. The second script, called 'Route creator', uses the postman problems package to calculate a route, this route is given in a table as well. The last script, called 'Route to ArcGIS features', converts this table to ArcGIS features, so that it is visible and can be used by the gas leak seekers. The relations between these scripts are shown in Figure 3. The scripts are explained more elaborately in the next sub chapters.

Figure 3: Flowchart of the route optimization tool

6.2 Information from ArcGIS

This script creates a table that contains the identifying numbers of all edges and their start- and end points, as well as their length and if they are required. The script exists of three sub scripts, because it requires two actions in ArcGIS that can not be performed via the script.

To create the table, firstly for each set of lines (high pressure pipelines, low pressure pipelines and optional paths) it is stated whether they are required or not. Then the lines are combined and converted into a set of lines with points at all nodes. The network then has to be scaled up, because for the next step all points need to be at least one meter apart from each other. This has to be done manually. For each line it is than stated what its start and end points are. Then the network is manually scaled down to its original size. Once the attribute table is adjusted so that the information is in the right order, the table can be exported and used to generate a route.

6.3 Route creator

The 'Route Creator' script uses the postman problems package to create a route. It uses the input table that is given by the first script and gives a list with the order in which the lines are traversed as the output.

In larger lines, the algorithm takes the following steps to generate an optimal route. Firstly, it removes the optional edges and it checks whether the graph is connected, whether there are no edges that are not connected to the other edges. It then takes all nodes with an odd number of adjacent edges and connects each node with another one in a way that minimizes the total distance of the paths that connect the nodes of these pairs. It then adds these paths to the graph. Now all nodes have an even number of adjacent edges. An eulerian circuit can now be made. The creation of the eulerian circuit happens by taking an edge adjacent to an in advance chosen starting node. It then keeps adding an adjacent edge until it returns at the starting node. It then seeks a node with untraveled edges and if there is one, it repeats the process to create a sub tour. The first tour and sub tours combined then form a route that traverses all edges and returns at the starting point.

6.4 Route to ArcGIS features

The last script converts the output table of the 'route creator' script to a route that consists of line features in ArcGIS. To do so, it creates two lists, one with the identifying numbers of the lines and one with the identifying numbers of the start nodes of the lines. Each line in ArcGIS has a direction, based on the direction in which it is drawn. This is not always the direction in which the line is traversed. Therefor, the set of lines is copied and flipped, so that there is now a set of lines in the original direction and a set of lines in the opposite direction. With the use of the identifying numbers of the lines and start nodes, a line will be picked from the right set and the lines are, one by one, added to the route. This route is than depicted on the map with arrows. Numbers indicate the order in which the lines have to be traversed and a dot indicates the starting point. An example of such a map is shown in Figure 4. The gas leak seekers can see this map and their location in Field Maps, the software they use.

The blue lines depict the connecting pipelines that lead to the buildings. They are not used while generating the route, since there often only is one way to walk along them, which means gas leak seekers do not have difficulties navigating along them. Furthermore, adding them to the route would increase the time needed for route calculation drastically, which is why it was decided to leave them out of the routes.

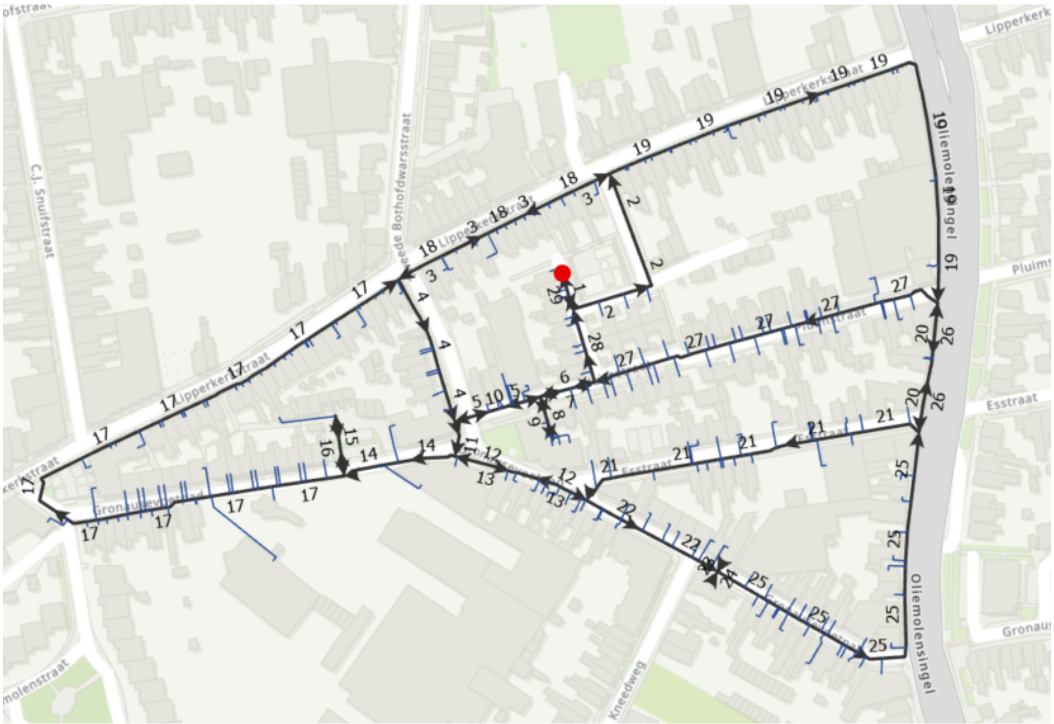


Figure 4: Example of a route

7 Verification

Once the route optimization tool was completed, it was time to verify if it met the requirements. The verification process focused on on three questions, based on the established requirements;

1. Does the tool generate a continuous route?
2. Does the route cover all of the pipelines?
3. Is the route shorter than the route currently taken by gas leak seekers?

7.1 Visual inspection

When using the network of Kwintsheul as an example, a visual inspection showed that the tool generated a continuous route, each line segment connects to the following. This can also be seen in Figure 5. The numbers indicate the order of the lines, when there are two ascending number on one line, this means that the line is walked back and forth.

Figure 5: The generated route with numbers

Furthermore, a visual inspection showed that the route covered all of the pipelines. In Figure 6, you can see the pipeline network. In the right figure, where the route is laid on top of the network, it covers the network completely.

(a) The network

(b) Generated route

Figure 6: The network and generated route of Kwintsheul

7.2 Distance saved

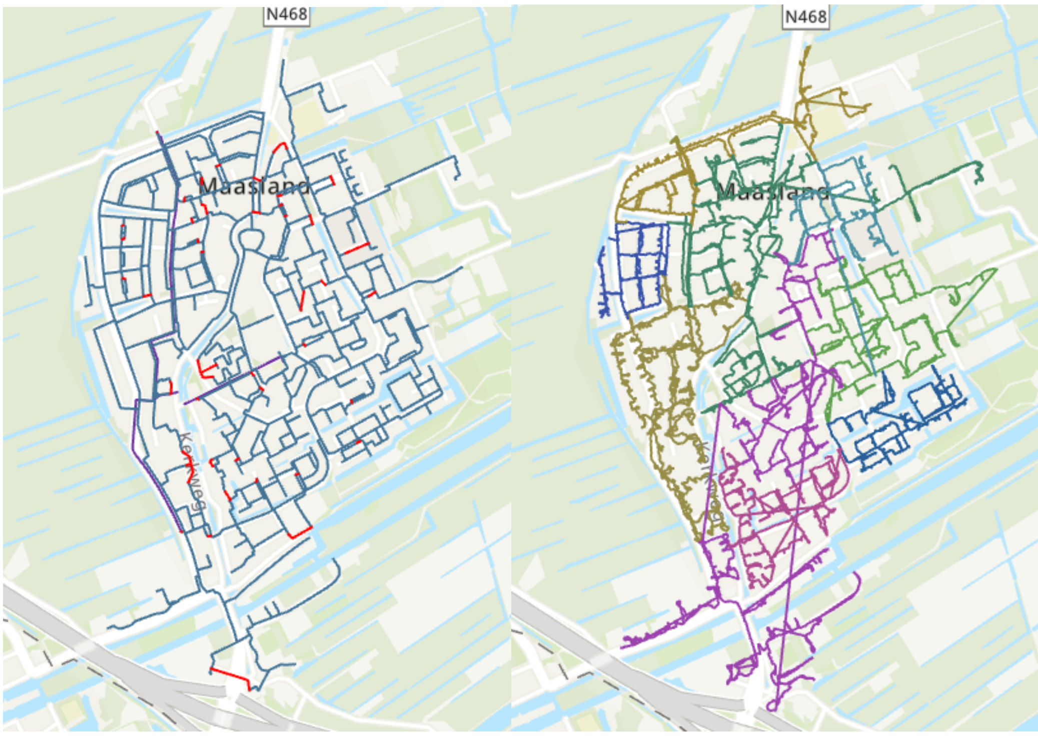
The next question then was, whether the generated routes were shorter than the routes currently walked by the gas leak seekers. To test this, gps data of the gas leak seekers was converted to trails. Since the generated routes do not include the connecting pipelines, the trails were converted to routes that follow the main pipelines. This also allowed for a better comparison, since the gps data is not always precise. With this conversion, the generated route and the current routes followed the same lines. Both the trails as the routes used for the comparison are shown in Figure 7.

(a) Trails

(b) Routes along pipelines

Figure 7: The trails are converted to Routes that follow the pipelines

In the case of the network of Kwintsheul, the length of the current routes added up to 16747 meters, while the length of the generated route was 15104 meters. This meant that the generated route was 9.81% shorter. With a walking speed of 5 km/h this would be a difference of 19 minutes and 42 seconds.



(a) Network (b) Trails

Figure 8: Case 2

The next case was the network in Maasland, which together with the walked trails can be seen in Figure 8. The length of the current routes added up to 42810 meters, while the length of the generated route was 41097 meters. This meant that the generated route was 4.00% shorter and would save 20 minutes and 32 seconds.

(a) Network

(b) Trails

Figure 9: Case 3

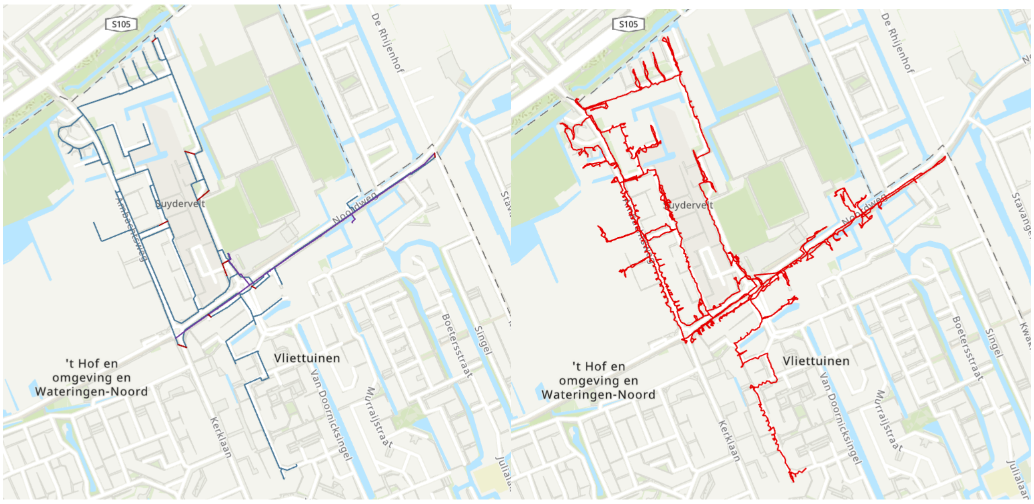
The third case was a network located in a rural area, a polder, and can together with the walked trails be seen in Figure 9. The length of the current routes added up to 44222 meters, while the length of the generated route was 40021 meters. This meant that the generated route was 9.50% shorter and would save 50 minutes and 22 seconds.

(a) Network

(b) Trails

Figure 10: Case 4

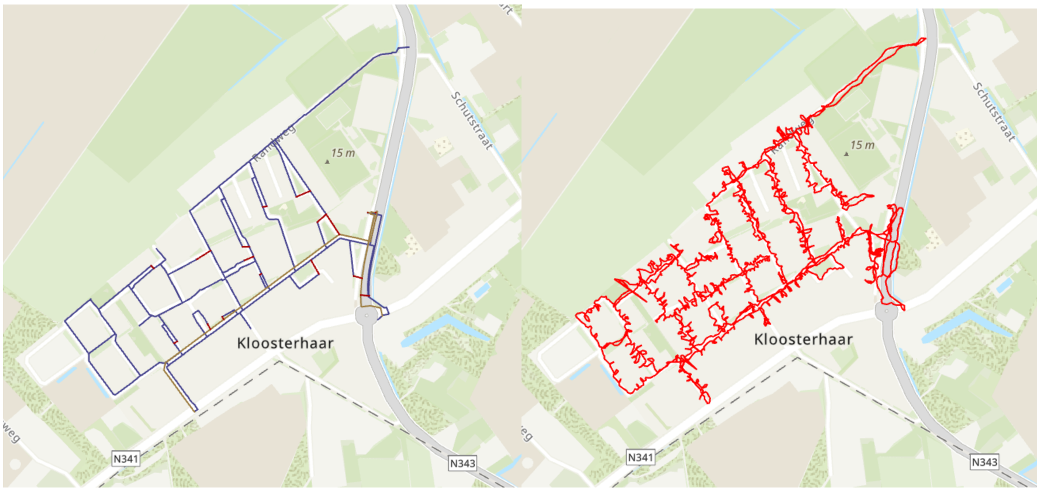
Case 4 was the network in De Krim, which can be seen in Figure 10. The length of the current routes added up to 16515 meters, while the length of the generated route was 14868 meters. This meant that the generated route was 9.97% shorter and would save 19 minutes and 44 seconds.



(a) Network (b) Trail

Figure 11: Case 5

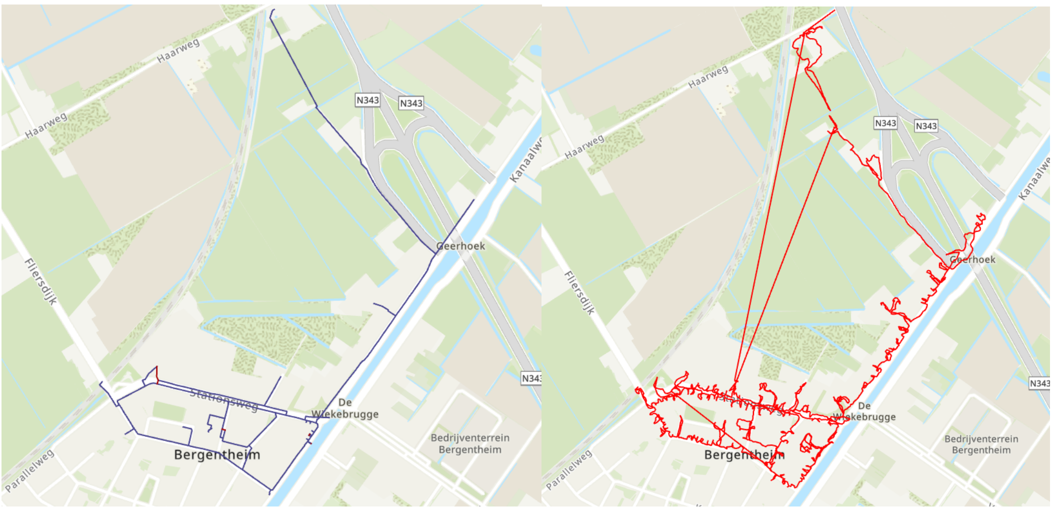
Case 5 was a route that a gas leak seeker walked through an urban area. It can be seen in Figure 11. The route had a length of 7845 meters, while the length of the generated route was 7502 meters. This meant that the generated route was 4.37% shorter and would save 4 minutes and 6 seconds.



(a) Network (b) Trail

Figure 12: Case 6

Case 6 was again a route that a gas leak seeker walked through an urban area.It can be seen in Figure 11. It can be seen in Figure 12. The route had a length of 8387 meters, while the length of the generated route was 7882 meters. This meant that the generated route was 6.02% shorter and would save 6 minutes and 3 seconds.



(a) Network (b) Trail

Figure 13: Case 7

Case 7 was a route that a gas leak seeker walked through an area with both urban and rural characteristics. It can be seen in Figure 13. The route had a length of 6735 meters, while the length of the generated route was 5954 meters. This meant that the generated route was 11.60% shorter and would save 9 minutes and 21 seconds.

In conclusion, in all 7 cases, the generated routes were shorter than the routes walked by the gas leak seekers. On average, the difference was 7.90%, which is significant. There was however, quite some difference between the results, which resulted in a standard deviation of 3.04%. A student t-score of 6.88 and therefor a p-value of 0.000465 meant that there was a 99.95% certainty that the generated routes were shorter than the current routes.

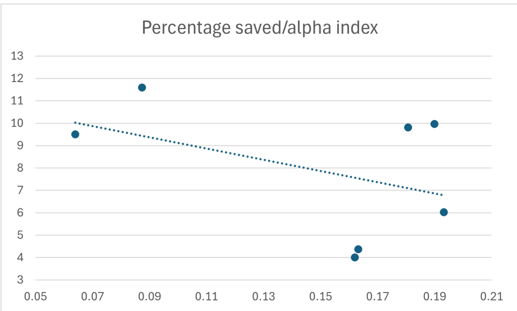
7.3 Influence of network properties

Once it was calculated how the route optimization tool performed, it was interesting to examine the relation between the properties of the network and the relative difference in distance between the routes generated by the tool and the routes currently walked. Firstly, the alpha-, bèta- and gamma indexes were calculated for all cases. These are showed in Table 1. For all three parameters higher values indicate more complex networks.It can be seen that the networks with higher alpha indexes, often also have a higher beta- and gamma index.

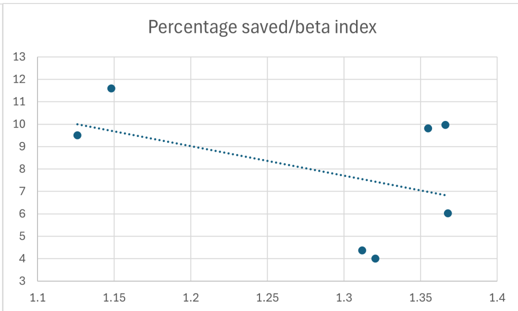
Case	alpha index	beta index	gamma index	Percentage
1	0.180722892	1.354948805	0.454753723	9.499796481
2	0.162032598	1.320610687	0.441890166	4.001401542
3	0.06401766	1.126099707	0.376470588	9.810712366
4	0.189964158	1.366197183	0.461904762	9.972752044
5	0.163265306	1.312	0.444444444	4.3722116
6	0.193236715	1.367924528	0.46474359	6.021223322
7	0.087378641	1.148148148	0.397435897	11.59613957

Table 1: Network properties

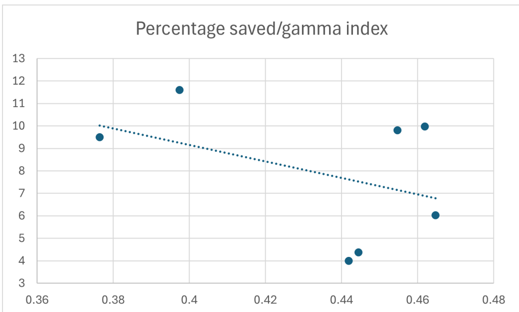
The graphs in Figure 14 show the relation between the network properties and the percentile difference in distance. Surprisingly, their trend lines show a negative relation between the complexity and connectivity of the network and the difference in distance. There are however, large differences between the results of the differences, even when their network properties are similar. It is therefor more likely that the network properties do not have an influence on the results.



(a)



(b)



(c)

Figure 14: Percentile savings compared to the network properties

8 User validation

Usability was another crucial requirement. To evaluate this, the algorithm was employed to generate routes for a gas leak detector. These routes were used over two days, and the gas leak seaker provided detailed feedback on the experience. Five specific questions were asked to obtain information:

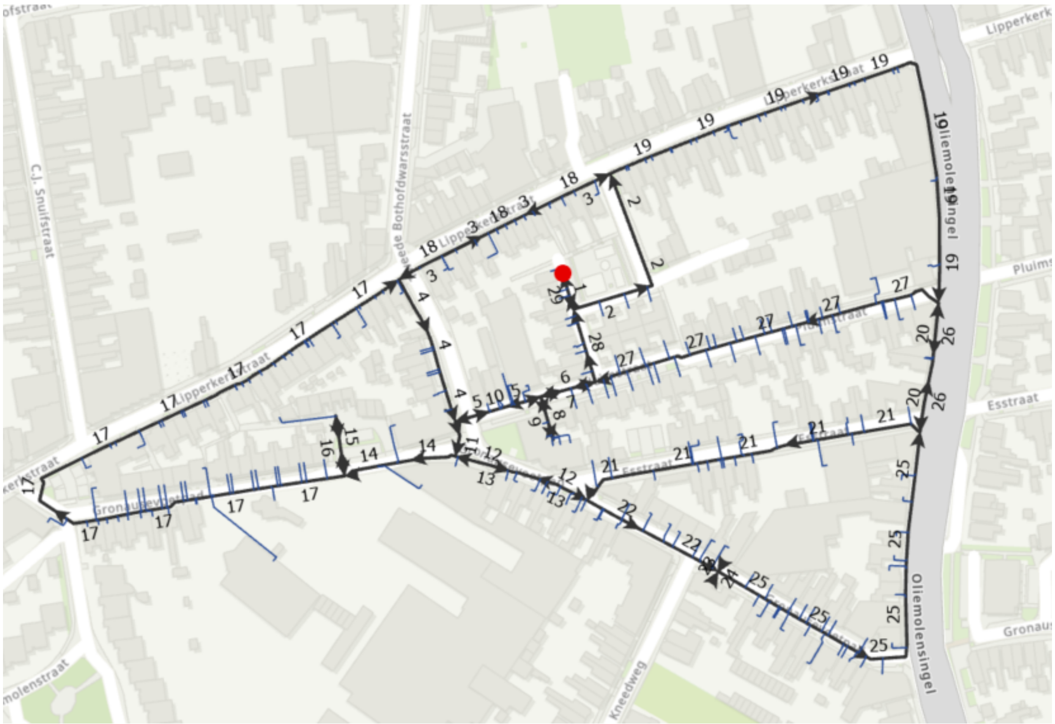
1. Was the route easy to follow?
2. Did it feel like this was the optimal route?
3. Is there a lot of difference between this route and how you would plan your route?
4. Is it a missing that the connecting pipelines are not a part of the route?
5. Is following and reading the route more or less work than planning it yourself?

Before testing, there was a short contact, where the gas leak seeker got an explanation about how the route is depicted. The route that was used as an example to help with the explanation is shown in Figure 15. When asked how the gas leak seeker would walk the route himself, he explained that he would try to create loops. The route as how he would take it is also shown in Figure 15. While at first the two routes appear to be quite different from each other, a closer look reveals that the same lines are walked twice. This means that they have the same length. It is worth mentioning that the gas leak seeker in question is experienced and is therefore able to generate efficient routes. Lastly, the gas leak seeker explained which area he wanted to cover the following days, so that the routes could be created.

After the first day of testing, there was a first feedback moment. The gas leak seeker was positive about the route, but was critical about how the map looked. The large number of numbers made the map a bit chaotic and difficult to read, especially when zoomed out. Additionally, the fact that both the lines and the numbers were black also made it more difficult to read. Based on this feedback, the color of the lines was changed to red and the distance between the numbers was increased. Therefor, the first question after the second day of testing was; Were the adjustments an improvement? The gas leak seeker was positive about the red color of the lines. They stood out more and were better visible. There were still a lot of numbers in the map, which made it difficult to read when zoomed out, especially when a line had to be traversed twice and therefore therefor had two different numbers next to it. Then the main questions were asked.

Was the route easy to follow?

The gas leak seeker said that it took a bit of time to adjust to following the arrows and the numbers, but that overall he was able to follow the route well. He pointed out that when a line was traversed back and forth, which means it has two succeeding numbers, it was hard to tell in which way to go. This could be a point of improvement.



(a) Generated route

(b) Route as the gas leak seeker would walk it

Figure 15: Two example routes, one generated by the tool, one as the gas leak seeker would walk it

Did it feel like this was the optimal route?

The gas leak seeker said that it felt as an optimal route, but that this was difficult to determine.

Is there a lot of difference between this route and how you would plan your route?

The gas leak seeker told that the route is quite different from how he would walk it, especially the order in which he would walk certain loops or lines. Similar to the example before testing, the lines that were walked twice were often the same, although now, with a larger network, this was not always the case.

Is it a missing that the connecting pipelines are not a part of the route?

This is not a missing since the connecting pipelines are often traversed in the same way. It actually makes it easier to read and follow the route, since there are less numbers in the map.

Is following and reading the route more or less work than planning it yourself?

The gas leak seeker answered that when following the route, he spent more time looking where he had to go. When he creates his own routes, he already has in his head how he wants to walk, so he spends less time looking where he has to go. On the other hand, he now did not have to plan the route in advance, which he often does the night before. He did however, still have a look at the route in advance to see how the route runs. This means he spent more time while walking and less time in advance. He was not sure if overall time was added or saved with this new method.

9 Discussion

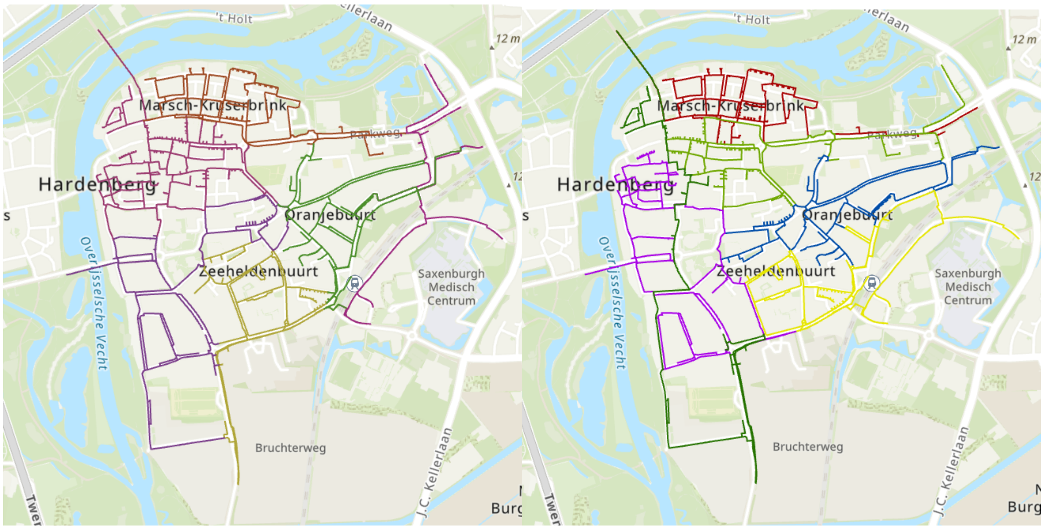
The verification showed that the route optimization tool is able to generate routes that cover all pipelines. Furthermore, these generated routes are on average 7.9% shorter than the routes currently walked by the gas leak seekers. The tool does have a few limitations. Firstly, the tool is not fully automatic. It does require some manual actions. This is because at one point in the process, the network needs to be scaled up, which can only be done in the ArcGIS program itself. Furthermore, the tool only works for a single connected network. This means that when parts of the network are not connected, additional lines have to be drawn to create a connected network. To generate optimal routes, additional optional paths have to be drawn. This means that a bit of work is needed in order to generate optimal routes.

Since the postman problems package that forms the core of the tool can not be adjusted, the use of a bike could not be accounted within the tool. There are however, other ways to account for the use of a bike. The first method is to generate routes for each day and cycle the longest line that is traversed twice. The second method is to generate one route for the complete network and split it up in routes that can be walked in one day. The gas leak seeker will then finish his route at a different location then where he started and cycle back to his starting point. This has as an advantage that it takes less time to create the routes, as only one route has to be generated. Though no elaborate research has been conducted on this, calculations also show that this would create more efficient routes. The network Hardenberg was used as a case for these calculations. In Figure 16a 6 individual routes are generated, while in Figure 16b one route is generated and split up in 6 parts. For method one, it is then assumed that the longest part of the route that is walked twice can be covered by bike, while for method two, the distance from the end point to the starting point is travelled by bike. When a walking speed of 5 km/h and a cycling speed of 15 km/h are assumed, method two is faster, as can be seen in Table 2.

Lastly, while the arrows and especially coloured arrows were experienced as a good way to indicate the route, the numbers may not be the best method, or they may need some fine tuning. For these numbers were not always easy to read.

	meters walked	meters cycled	seconds walked	seconds cycled	total time
method 1	40274	2334	29184	559	29743
method 2	37047	4539	26845	1088	27933

Table 2: Results of the calculations



(a) 6 individually generated routes

(b) One route, split up in 6 parts

Figure 16: Routes for Hardenberg, created through different strategies

10 Conclusion

In conclusion, a route optimization tool was created to generate routes for gas leak detection. This was done by combining existing algorithms with self written scripts to integrate them with ArcGIS. The tool was able to generate routes that covered the gas networks completely. These routes were on average 7.9% shorter than the routes currently taken. Furthermore, a gas leak seeker has already used the tool and was able to use it to perform his daily tests.

References

Becker, M. J. (2015). Technical University of Munich. (https://algorithms.discrete.ma.tum.de/graph-algorithms/hierholzer/index_en.html)

Bovengronds gaslekzoeken - methode sleepmat [Computer software manual]. (n.d.).

Chapleau, L., Ferland, J. A., Lapalme, G., & Rousseau, J.-M. (1984). A parallel insert method for the capacitated arc routing problem. *Operations Research Letters*, *3*(2), 95–99.

Hu, T., & Kahng, A. B. (2016). *Linear and integer programming made easy*. Springer.

Lu, H., Xu, Z.-D., Iseley, T., Peng, H., & Fu, L. (2023). *Pipeline inspection and health monitoring technology: The key to integrity management*. Springer.

Netbeheer Nederland. (2022). *Veelgestelde vragen* (Tech. Rep.). Netbeheer Nederland.

Nobert, Y., & Picard, J.-C. (1996). An optimal algorithm for the mixed chinese postman problem. *Networks: An International Journal*, *27*(2), 95–108.

Tirkolaee, E. B., Goli, A., Gütmen, S., Weber, G.-W., & Szwedzka, K. (2023). A novel model for sustainable waste collection arc routing problem: Pareto-based algorithms. *Annals of Operations Research*, *324*(1), 189–214.

Vöcking, B., Alt, H., Dietzfelbinger, M., Reischuk, R., Scheideler, C., Vollmer, H., & Wagner, D. (2010). *Algorithms unplugged*. Springer Science & Business Media.

Zabodin, R. (2015). Technical University of Munich. (https://algorithms.discrete.ma.tum.de/graph-algorithms/directed-chinese-postman/index_en.html)