Route planning using multiple attributes

Finding routes other than the shortest for bicycles

EMIL PLYNNING
Abstract

Congestion and pollution are two ever increasing problems in our would of urbanization. Bicycling is one of the most sustainable means of transport and it is a great way of reducing congestion in cities. Route planners which brings out the best aspects of bicycling and promotes them during route calculation is a good way of increasing the attractiveness of bicycling.

When planning a route today there are online services which can perform regular route planning taking only the length of each route into consideration. The objective of this paper is to introduce an alternative way of performing route planning using other traits than just the length when determining the best route.

This paper introduces an algorithm which is able to perform route planning which takes several aspects into account. The algorithm uses two additional attributes together with the length to determine the cost of traversing each polyline. The additional attributes used are named nature and slope index. The nature index is supposed to give a numerical representation of the environment surrounding the polyline, this is calculated using a land cover map and a buffer around each polyline which represents its neighborhood. The nature index is used to make the route planner prefer more beautiful paths along water, park or forests over shorter ones which passes through high density development for instance. The slope index is stores the average slope of the entire polyline. It is calculated using a digital elevation model and dividing the height difference between the start and end point with the length resulting in a slope percentage. The slope index is used to find paths which are as flat as possible.

The two indexes together with the length attribute are then combined in seven different cost functions, each weighting the attributes differently to accommodate different preferences. Each of the calculated costs is stored as an attribute for every polyline. Networks on which route planning can be performed are then built using these cost attributes.

The resulting networks are inspected visually using the indexes as display factors and determined to be accurate. Route planning using these networks results routes do follow areas with higher nature index and avoids slopes. In this paper a few examples of routes with an orthophoto as background are included which clearly illustrates that the nature index promotes the correct type of environment.
**Sammanfattning**

Trafikstockning och föroreningar är två växande problem när vår värld fortsätter att urbaniseras. Cykel är ett av de mest hållbara transportmedlen och är samtidigt väldigt effektiv för att minska trängseln i städerna. Rutplanerare som framhäver de bästa aspekterna med att cykla och använder dem vid beräkningen av den bästa rutten är ett bra sätt att öka attraktiviteten med att cykla.

Vid planering av rutter med hjälp av online-verktyg så tas i dagsläget endast längden av rutten i beaktning. Den här rapporten syftar till att introducera ett alternativt sätt att beräkna rutter som tar fler aspekter än bara längden i beaktning för att hitta den optimala rutten.


De resulterande nätverken inspekeras sedan visuellt där de olika indexen används som visnings faktorer. Resultatet bedöms vara korrekt baserat på inspektionen. När dessa nätverk används för rutplanering resulterar det i vägförslag som faktiskt följer områden med högre nature index och föredrar platta vägar. I rapporten ingår några exempel av rutter där ett ortofoto används som bakgrund för att få en visuell referens. Det är tydligt att nature index faktiskt framhåver rutter som följer rätt typ av miljö i exemplet.
Acknowledgements

This paper has been produced during the spring of 2016 and is the result of a 15 credits bachelor thesis. The bachelor thesis is a part of a five-year master education oriented towards geoinformation technology at the Royal Institute of Technology in Stockholm, Sweden.

I would like to thank my supervisor associate professor Gyöző Gidofalvi for his help during the course of this project. His help and comments have been of great importance for the outcome of this project.
Contents

Abstract ................................................................................................................................. i
Sammanfattning ................................................................................................................... ii
Acknowledgements .......................................................................................................... iii
List of figures .................................................................................................................... vi
List of tables .................................................................................................................... vii
List of formulas ............................................................................................................... vii

1 Introduction .................................................................................................................... 1
  1.1 Background ................................................................................................................. 1
    1.1.1 Nature Index .......................................................................................................... 2
    1.1.2 Slope Index .......................................................................................................... 2
  1.2 Objectives .................................................................................................................. 2

2 Related work .................................................................................................................. 2
  2.1 The shortest path to happiness.................................................................................. 3
  2.2 Bicycle route planner – Fietsersbond route planner .............................................. 3
  2.3 EcoTour: Reducing the Environmental Footprint of Vehicles Using Eco-Routes ...... 5
  2.4 Happiness is greater in natural environments ......................................................... 6
  2.5 Accommodating user preferences in the optimization of public transport travel ...... 6

3 Study area and collection of data ................................................................................... 8
  3.1 Network data .............................................................................................................. 8
  3.2 Land cover map ......................................................................................................... 9
  3.3 Digital elevation model ............................................................................................ 9
  3.4 Orthophoto ................................................................................................................ 9

4 Methodology .................................................................................................................. 9
  4.1 Summary of the algorithm ....................................................................................... 9
    4.1.1 Generic summary of the whole algorithm ......................................................... 10
    4.1.2 Summary of each step of the algorithm ............................................................ 10
  4.2 Network data preparation ....................................................................................... 14
  4.3 Unique IDs ............................................................................................................... 15
  4.4 Nature index ............................................................................................................. 15
    4.4.1 Land cover map modification ............................................................................ 15
    4.4.2 Weighting each environment type .................................................................... 18
    4.4.3 Nature index calculation .................................................................................. 18
  4.5 Slope index ............................................................................................................... 19
  4.6 Normalizing indexes and assigning net type weight ............................................... 21
    4.6.1 Nature index normalization .............................................................................. 21
    4.6.2 Slope index normalization ............................................................................... 21
    4.6.3 Net type weighting ........................................................................................... 22
  4.7 Cost calculation ....................................................................................................... 22
  4.8 Cost examples .......................................................................................................... 24
  4.9 Applying the costs .................................................................................................... 24
  4.10 Network building .................................................................................................... 24
List of figures

Figure 1: Screenshot taken from the online editor showing the network used by the Fitserbond route planner. Source: Kamminga, (2016). ................................................................. 5

Figure 2: On the left a simple transportation network connected only by buses. On the right a more complex network with walking connections between all bus stations added. Source: Hartley & Wu (2004). .................................................................................................................. 7

Figure 3: The study area and the road coverage of the chosen municipalities. The green lines represent the network data used in this paper. Coordinates of the upper left corner (north western corner): (660800.00, 6592900.00) Coordinates of the lower right corner (south eastern corner): 686500.00, 6566000.00) (SWEREF99 TM) .................................................................................................................. 8

Figure 4: A generic summary of the complete algorithm. ................................................................................. 10

Figure 5: Step A of the data preparation. ........................................................................................................... 11

Figure 6: Step B of the data preparation. ........................................................................................................... 11

Figure 7: Modification of the land cover map. .................................................................................................. 12

Figure 8: Nature index calculation. ................................................................................................................ 12

Figure 9: Slope index calculation. .................................................................................................................... 13

Figure 10: Step A of the cost calculation and network building. ....................................................................... 13

Figure 11: Step B of the cost calculation and network building. ....................................................................... 14

Figure 12: The same area displayed with different amounts of information using. The left is stripped of information and the right contains all the original information stored by GSD-Terrängkartan .................. 16

Figure 13: Major traffic solutions receive the same classification as parks. ................................................. 16

Figure 14: The major road Klarastrandsleden is buffered but the road segment going through the tunnel Bleholmstunneln is not buffered ............................................................................. 17

Figure 15: The values used to reclassify the environment classification map. The left column contains the old classes and their values and the right column contains the new classes and their values. The arrows point out which original class is reclassified into which new class ........................................................................ 17

Figure 16: The assigned weight for each class in the environment classification map. ................................. 18

Figure 17: The difference in the calculated slope index for a road going up and down a hill. On the left the slope index is calculated using the default length of the polyline and on the right the length of the polyline is lower ................................................................................................................. 20

Figure 18: A road initially going uphill but then level out. On the left the slope index calculated using the original polyline length and on the right the slope index calculated for polylines along the same road with lower length. ......................................................................................................................... 20

Figure 19: Normalization of the nature index plot ............................................................................................. 21

Figure 20: Normalization of the slope index plot ............................................................................................. 22

Figure 21: Extract of the attribute table belonging to the unbuilt network. ..................................................... 25

Figure 22: Visualization of the network using the Nature Index to determine the display color. ................. 26

Figure 23: Visualization of the network using the Slope Index to determine the display color ........................ 27

Figure 24: Loss of connectivity due to removing polylines of a certain road class. ....................................... 28

Figure 25: Routes between Fridhemsplan and Rålambshovsparken using differently weighted networks. ............................................................................................................................................... 29

Figure 26: Routes between Bondegatan, close to Medborgarplatsen on Södermalm and Hornstull using differently weighted networks. ........................................................................................................ 30

Figure 27: Paths between Ålvsjö south of Stockholm and Skärsbrink close to Gullmarsplan. This figure shows a larger deviation between different paths than the other figures which are at smaller distances. 31

Figure 28: Two overlapping service areas with the center in Gamla Stan, central Stockholm. The blue polygons represent the range using cost2 and the pink represent the range using cost1 with a simulated constant nature index of 50. ...................................................................................... 32

Figure 29: Two overlapping service areas with the center at Norra Stationsgatan, the outer boundary of central Stockholm. The blue polygons represent the range using cost2 and the pink represent the range using cost1 with a simulated constant Nature Index of 50 ...................................................................................... 33

Figure 30: Two overlapping service areas with the center in Östermalm, central Stockholm. The blue polygons represent the range using cost2 and the pink represent the range using cost1 with a simulated constant Nature Index of 50 ...................................................................................... 34
List of tables

Table 1: Network datasets downloaded and the attributes used from each dataset. ............................................14
Table 2: The parameters used to generate a buffer for each polyline.................................................................19
Table 3: Cost of two polylines given different cost functions and the percentage difference between them. .................................................................24
Table 4: The attributes of the finished shapefile explained. .................................................................................26

List of formulas

Formula 1: Nature index formula ........................................................................................................................19
Formula 2: Slope index formula .........................................................................................................................20
Formula 3: Normalization formula for the nature index .....................................................................................21
Formula 4: Normalization of the slope index ......................................................................................................21
Formula 5: Net type weighting formula ...........................................................................................................22
Formula 6: Cost1 formula ...................................................................................................................................22
Formula 7: Cost2 formula ...................................................................................................................................23
Formula 8: Cost3 formula ...................................................................................................................................23
Formula 9: Cost4 formula ...................................................................................................................................23
Formula 10: Cost5 formula ..................................................................................................................................23
Formula 11: Cost6 formula ..................................................................................................................................23
Formula 12: Cost7 formula ..................................................................................................................................23
1 Introduction

An increasing number of people all over the world are moving into cities, a phenomenon commonly called urbanization. The majority of the world’s urbanization is happening in Asia and Africa but Stockholm is no exception (United Nations, 2014). Stockholm’s population is growing by 35 000 – 40 000 inhabitants per year, which is corresponds to an annual growth of about 2 percent (Trafikverket, 2016). This threatens sustainable development when the expansion is progressing so fast that there is not enough time to develop the necessary infrastructure to support the population. Inadequately managed urban expansion leads to increasing congestion, pollution and environmental degradation (United Nations, 2014).

To support this huge growth, billions of Swedish kronor are annually invested towards infrastructure (Trafikverket, 2016), Stockholm has further more decided to invest an additional billion kronor over a six-year period dedicated to bicycle transportation alone (Stockholm, 2012). These investments are not enough and congestion is still an increasing problem. Resorting to other means of transportation than cars is an important step towards a more sustainable society and an increasing number of cities see the importance of adjusting taxes and fees so that the negative consequences of driving into cities are reflected in the cost of car use (OECD, 2015). This has also been seen in Stockholm where a congestion fee has been introduced to discourage people from traveling by car in the central parts of Stockholm. As of January 1 2016 the congestion fee has also been applied on Essingeleden which is Sweden’s most trafficked road with upwards of 170 000 vehicles passing every day (Trafikverket, 2016). The inclusion of Essingeleden in the congesting fee system is projected to reduce the number of cars passing each day by 10% (Trafikverket, 2016). With the hope that these will resort to other means of transport, such as public transport or bicycles.

Aside from discouraging car users it is equally important to encourage bicycle users. One of Stockholm´s goals in its action plan for climate and energy is to increase the proportion of cyclists and pedestrians. This is progressing well and the bicycle traffic has increased by ten percent each year for five consecutive years. Today 18% of travelers using a personal vehicle to and from the inner city use a bicycle. Even though this a huge increase Stockholm still has a lot of potential to further increase this number. Around 80% of work-related journeys are below ten kilometers which is considered as an interval where bicycle is a competitive mean of transport and can replace the car (Stockholm, 2012). This means that the potential number of candidates which could use a bicycle is far greater than the number of cyclists we see today.

Stockholm has listed several actions which are supposed contribute to an increasing number of bicyclists, one of which is “improving information about bicycle paths, for example informing about the network of bicycle roads, cycle travel planners, current re-direction of traffic in connection with traffic works, as well as making this information accessible via mobile phones” (Stockholm, 2012). Stockholm themselves have introduced a bicycle route planning service which is in line with the goal. Despite this goal the service has been discontinued during the course of this project with the motivation that Stockholm wants to prioritize its public transport planner. The public transport planner does not allow route planning using a bicycle, for bicycle route planning the website instead refers to Google Maps (Trafiken, 2016). The existence of a bicycle planner is an important feature that is needed to be able to promote bicycling as a mean of transport which can compete with cars.

1.1 Background

Network analysis is commonly used to study flows or movement along a network. An everyday technology which makes use of network analysis is GPS navigation systems which use distance, speed and traveling time as cost parameters to find the optimal route between two locations (Harrie, 2012).
A network used in network analysis consists of nodes and edges which connects the nodes. A node can typically be an intersection in a road network, an edge connected to that node would in this case be one of the roads leading to the intersection (Harrie, 2012). Edges are referred to as polylines though out this paper.

Every edge in a network is associated with some sort of cost. Common examples of costs can be Euclidian distance or driving time. The cost of each edge is used by an algorithm to find the lowest cost path through the network. The most commonly used algorithm is Dijkstra’s algorithm (Harrie, 2012).

Ordinary route planners commonly use distance as traveling time as cost parameters when finding the optimum route, but there are services which takes other aspects into account. As mentioned above the algorithm in this paper aims to find the optimum route using slope and amount of nature along the route as additional aspects. To enable route planning using other aspects than length it is necessary to assign additional attributes aside from the length which will be taken into consideration when calculating the cost of each polyline. In this paper two additional attributes are considered sufficient given the limited amount of time and data available. These are described in the following sub sections.

1.1.1 Nature Index
The first additional attribute is the nature index. It is gives a numerical representation of the environment surrounding the polyline and is calculated using a land cover map as basis. The nature index is used to make a route planner prefer more beautiful paths that go along water, parks or forests over shorter ones which for instance passes through high density development. A higher nature index is considered better when planning a route.

1.1.2 Slope Index
The second attribute which is added is the slope index. The purpose of the slope index is to enable the route planner to find paths which are as flat as possible. During calculations the absolute value of the slope index is used which means that a slope index as close to zero as possible is the most preferable according to the planner. This leads to the route planner ranking polylines with down or uphill slopes lower than flat ones. What is down or up is of course relative and depends on which way you are traveling.

1.2 Objectives
This report aims to create a concept for a route planner which will increase the appeal of using a bicycle instead of traveling by car. The planner is supposed to highlight and use the aspects of bicycling that are some of its biggest upsides compared to traveling by car. Such as the outdoor experience and contact with natural environments. This can then be used as a way of promoting biking.

2 Related work
In this section four scientific reports and one service that is very similar to what is done in this paper are presented. The first report presented introduces a route planning service which uses crowd sourcing to determine the beauty of locations which then is taken into account when planning a route. The second sub section contains a presentation of the Fietsersbond route planner which is a very advanced route planner that is able to accommodate user preferences using a network that is crowdsourced on a small scale. The network contains highly detailed information. It uses a weighting system much like the one used in this paper to determine the best path depending on the criteria of choice. After the Fietsersbond a Danish route planning service called EcoTour which finds the eco-friendliest route is presented. It also uses a weight based system to find the most appropriate route in combination with crowdsourced data to assign the additional attributes. The third report presents the strong connection between being in natural environments and wellbeing. The last report discuses different methods of taking several user preferences into account when planning a single route.
2.1 The shortest path to happiness

Aiello et al. (2014) have created an algorithm which finds routes for pedestrians using four different criteria. Their algorithm can find the shortest, the most beautiful, the quietest or the happiest route between two points.

Aiello et al. (2014) use crowdsourcing as a basis for determining how beautiful a certain location is. This is done using a website which displays two photographs of random urban scenes and let the visitor select which one of the two is more beautiful, quiet, or happy. The photographs are taken from Google street view. They use images from Google street view because they are very consistent when it comes to image quality whereas pictures taken by individuals and uploaded to Flikr or similar pages vary a lot. This information is then plotted on a map which they use to determine the beauty score of each polyline.

The algorithm finds the best fitting path by running the k-shortest path algorithm which finds the k shortest paths between two points. The beauty score for each route is then added and the highest ranked route is the selected. They use a very large number for k to ensure that they are exhaustive enough. Then a comparison is performed between the first m paths found and the best ranked path among those is stored. This is then repeated for the next m paths and until the rank between the best path overall and the best path from \( m_i \) do not change, this is used to interrupt the algorithm early. This improves performance and allows the algorithm to be interrupted early before all \( k \) paths are calculated. The process is repeated for all four criteria resulting in one beautiful, one quiet and one happy route between the points in addition to the shortest one, called the baseline. According to their tests these paths turn out to be on average just 12 percent longer than the shortest path which makes the additional distance surmountable for a pedestrian.

To evaluate the routes generated by their algorithm they use people familiar to the area and allowed them to determine if the beautiful route actually were more scenic than the shortest route etc. The results were very convincing and proved that their algorithm did indeed find more beautiful routes.

The usage of a map to store the underlying information about which locations are beautiful etc. is very similar to this paper's approach. The difference is that the map used by Aiello et al. is generated via crowdsourcing and the map used by this paper is created by an authority and stores land cover class which is then assumed to have a certain level of beauty or nature.

2.2 Bicycle route planner – Fietsersbond route planner

A route planner which accommodates for very detailed user preferences is the bicycle route planner provided by the Dutch cyclist union, the Fietsersbond. It provides complete coverage of the Netherlands and is able to handle a lot of user inputs.

It comes with the following predefined profiles for route planning:

1. **Limited Stops**: this route uses cycling paths alongside roads and main cycling routes as much as possible. It also avoids traffic lights when there is a good alternative. Routes of this type are therefore not always the shortest but are relatively quick to cycle.

2. **Racer Route**: this route uses roads with a good surface, preferably asphalt. It avoids narrow, crushed shell paths, and doesn’t use unpaved roads. The route prefers roads with a more or less recreational character and roads outside urban areas.

3. **Scenic Route**: the scenic route stays as much as possible outside urban areas. Outside nature areas it will lead you as much as possible along open countryside, and within urban areas through parks or via bike paths or roads with lots of green and few cars.
4. **Shortest route**: the shortest route is the route that covers the shortest distance.

A few more profiles are included but these are more linked to the cycle network in the Netherlands and therefore not of interest in this paper. Aside from this predefined profiles it is also possible to create a custom weighting which allows for full accommodation of the user’s preferences.

The profiles described above are very similar to what the algorithm in this paper is trying to accomplish. The big difference is that the network data used by this planner is maintained and edited by volunteers from the Fietsersbond. This allows for very comprehensive data with a lot of different attributes. Data of this quality and amount of information is unavailable in Sweden. The Fietsersbond use 20 attributes for each road, six of which are obligatory and the rest being extra information that is optional for the contributor to include (Kamminga, 2016).

The network itself is also modifiable by the users. They can edit the stretch of each road, add new roads and remove roads. This gives a huge freedom and allows for a more realistic network and a better representation of roads that are available. It is not uncommon with roads that are not officially considered as a bike trail and therefore not included in the downloadable datasets. The user edited network allows for a workaround for this problem (Kamminga, 2016). A sample of the Fietsersbond network is shown in Figure 1.

Using all this data they have created a planner that caters different types of cyclists with different kinds of route types. The general shortest route just takes the length of each path and finds the path with the shortest summarized length. Aside from this the planner has two different mechanisms that can change route-choice:

- **Weight**: the route planner assigns a value that the length of the road will be multiplied by to modify its cost based on each attribute. For instance, a stretch of road with a concrete surface might get a weight of 0.9. This will make it come up sooner in the route-advice. But if that same stretch of road is ill-maintained that will give it an additional weight of 1.2, making the total cost during route-calculation equal to $0.9 \cdot 1.2 \cdot \text{length}$ (Kamminga, 2016).

- **Point-penalties**: Some places in a route network can really be an annoyance to cyclists, without being easy to translate as a weight or attribute for a road. This can for instance be bollards or traffic lights. The Fietsersbond use a system of penalties that can be given to these kinds of hindrance. This penalty basically just adds an extra amount of cost to the route for each instance of a point penalty passed along the route (Kamminga, 2016).

Overall the similarity of the Fietsersbonds approach to route planning and the algorithm in this paper is very similar. Both use a weighting system to calculate the cost of each polyline. The difference is that the network used by Fietsersbond is on a small scale crowdsourced which allows the set of attributes to be a lot more comprehensive and take even more aspects into account. It also enables a more realistic topography with the possibility to add additional unofficial bicycle paths to the network.
2.3 EcoTour: Reducing the Environmental Footprint of Vehicles Using Eco-Routes

Andersen et al. (2013) have developed a route planning service for cars in Denmark which generates the route with the lowest environmental impact. They use an OpenStreetMap representation of the road network on which an additional attribute call eco-weight is stored together with the distance and traveling time. The eco-weight is based previous journeys along the road and the data originates from GNSS data in combination with fuel consumption data from busses. Much like the Fietsersbond Andersen et al. (2013) use the extra attribute in a weighting formula which calculates the fuel consumption for each polyline.

Since fuel consumption is very dependent of the amount of traffic at the time of travel the EcoTour uses different weights depending on the time of day. They have divided each day into 96 15-minute intervals which stores separate time and eco-weights. The division of a day into smaller segments cause some polylines to have insufficient amount of data to be assigned with a time or eco-weight. This issue is solved by combining data from the correct time of day with data from the rest of the day until a sufficient amount of data to calculate the weights is reached. Time dependent weighting increase the accuracy of the suggested routes.

In an example shown by Andersen et al. (2013) the eco-route use 22 percent less fuel and is only 3 seconds slower than the fastest route. In the same example it uses 2 percent less fuel than the shortest route whist being 3 minutes and 38 seconds faster. Earlier studies examined by Andersen et al. show that on average the eco-route takes 9 percent longer than the fastest route but uses 9 percent less fuel.

Andersen et al. (2013) also base their additional attributes on crowed sourced data, only a different form of acquiring is used. Instead of using voluntary data input or contribution like the Fietsersbond and Aiello et al. (2014) they use the GNSS data which is stored passively for all users.
2.4 Happiness is greater in natural environments

There are few studies on the topic of wellbeing in combination with environmental factors. MacKerron and Mourato (2013) introduce a method using GPS positioning and a smartphone that examines the relation between momentary subjective wellbeing and the environment type that individual currently is resided in. Questionnaires are sent out to the participants at random moments in time asking about their current state of wellbeing. The environment type linked to the answers is gathered using a land cover map in combination with the GPS position of the smartphone. Linking each answer to an environment type enables MacKerron and Mourato (2013) to study the correlation between wellbeing and environment type.

The questionnaire consists of three parts. The first one involving feelings, here the participants are asked to state how happy, relaxed and awake they are. Each of these questions allowed the participant to select where between for instance “not at all happy” and “extremely happy” they currently are. The second question stored if the participants were indoors, outdoors or in a vehicle. The last question involved what activities the participant was doing at the time of the answer.

In total they collected more than 1 000 000 answers from over 20 000 individuals. Analyzing this data showed that on average the participants were substantially happier outdoors in natural environments types than they were in urban areas. This relationship was highly statistically significant.

The study provides strong evidence of the link between natural environments and wellbeing, strengthening the existing evidence of the positive effects which nature has on our daily wellbeing. This motivates route planning using additional aspects such as a nature index even more since there is a strong link between productivity in the workplace and wellbeing (Oswald et al., 2014).

2.5 Accommodating user preferences in the optimization of public transport travel

Hartley and Wu (2004) use network analysis in combination with user preferences to find the public transportation route that accommodates three different user preferences the best. The preferences taken into account are minimum travel time, minimum number of bus-changes, and minimum walking distance.

Their cost-calculations between bus stations are not based on coordinates or network distance. Instead they are based on time tables and use traveling time as cost. The walking distance uses to coordinates of each bus stop together with Pythagoras theorem and a fixed average walking speed of 5km/h to determine the travel time by foot. This makes the network bi-modal, which is more realistic but also causes a fundamental topological change to the network and significantly increases the complexity of it. The difference between the network with only bus stops and the bi-modal network is shown in Figure 2.

Their usage of Pythagoras theorem is a very weak form of measurement in a city since the actual distance required to get from one point to another almost always differ quite a lot from the distance calculated by Pythagoras which is the shortest possible. The best results are achieved by using the network distance which is the real distance. Another approach is to use the distance retrieved by Pythagoras multiplied by a factor which represent the average additional length required when traveling through the network.
Hartley and Wu (2004) use two different methods to accomplish the route finding. The first method is using three different single-purpose shortest path algorithms to evaluate each preference separately. The second method they are using is the \( k \)-shortest paths algorithm. It computes a number of ranked shortest paths which are evaluated with respect to each other and consideration to the preferences. The optimum path among those is then selected.

The single-purpose shortest path algorithms used is based on Dijkstra’s algorithm. Each algorithm handles one user preference and they are very sensitive to how the constraints (preferences) are defined if the correct route is to be found. The constraints are added onto the standard shortest path algorithms or various basic shortest path algorithms are combined together. The algorithms compute one shortest path each and are capable of generating routes which fulfill the user’s preferences.

Different algorithms use different parameters and sometimes involve a completely new parameter that is not required or used by other algorithms. For instance, the algorithm which finds the route with the smallest number of transfers needs to store information about how many transfers it took to get to the node and all arrivals with the same number of transfers must be stored at the node as well.

The second approach of accommodating public transportation users’ preferences used by Hartley and Wu (2004) is the \( k \)-shortest paths algorithm. The \( k \)-algorithm is based on multipurpose network search. The algorithm repeats itself until the \( k \)-th ranked path has been outputted. \( k \) is specified by the user.

The \( k \)-algorithm can handle the user preferences in two ways. The first way is done by comparing each route with the preferences and choosing the first route which satisfies the preferences. The second approach regards the preferences as constraints and embeds them into the algorithm. Hartley and Wu (2004) used the first option but made efforts to implement some constraints. The second approach was only partly implemented due to the fact that the \( k \)-algorithm is consumes a lot of computational power and memory together with the problem that the constraints must be defined carefully to get the correct result. One of the constraints made was to remove all walking links which had a traveling time exceeding 10 minutes. This constraint proved helpful to increase the efficiency of the algorithm since the amount of connections in the network is heavily reduced. Constraints of this type are very subjective and a potential weakness. For instance, if someone is willing to walk more than 10 minutes the optimal path for that person might not be found.
Hartley and Wu (2004) came to the conclusion that the single purpose shortest path algorithms worked efficiently but could only accommodate one user preference whilst the K-shortest path algorithm generated multiple routes in one time which could satisfy several preferences but required a longer computational execution time.

3 Study area and collection of data

The area of interest in this study is Stockholm and its neighboring municipalities. To reduce the amount of data required the municipalities of interest were narrowed down to Danderyd, Huddinge, Solna, Stockholm, Sundbyberg and Lindingö. A map of the study area is shown in Figure 3.

---

**Figure 3**: The study area and the road coverage of the chosen municipalities. The green lines represent the network data used in this paper.

Coordinates of the upper left corner (north western corner): (660800.00, 6592900.00)
Coordinates of the lower right corner (south eastern corner): 686500.00, 6566000.00 (SWEREF99 TM)

3.1 Network data

The network data used was downloaded from Trafikverkets (the Swedish transport department) geo-data distribution service, Lastkajen 5.2. Three datasets are used, these are further described in section 4.2.
Aside from the three main network dataset used road data containing attributes of which polylines that are tunnels or bridges is also used to modify the land cover map. See section 4.1 for a description of how this data is used.

3.2 Land cover map
The map used to determine the environment type around each line is GSD-Terrängkartan in vector-format. It consists of administrative division, streets, public and private roads, railways, power lines, lakes, rivers, degree of development, buildings, different types of forest, open land, marshes, exposed rock surfaces, restrictions on land use, mapping text and contours. The map was downloaded from Lantmäteriets (the Swedish land survey agency) geodata extraction tool (GET) maps.slu.se. The original land cover map is modified to better fit its purpose in this paper, see section 4.4.1.

3.3 Digital elevation model
To determine the slope of polylines a digital elevation model is used. The highest precision free of charge digital elevation model in Sweden is the 2-meter resolution laser data provided by Lantmäteriet. It covers the whole nation and comes in the form of a regular grid of laser points scanned by airplane. The grid covers both land and water and the water points coincide with the water surface.

3.4 Orthophoto
To provide a way of visually inspecting the results of calculations and the routes suggested by the planner an orthophoto is needed. Lantmäteriets orthophotos are usually very expensive but as of 1 January 2016 the government has decided to remove the fee if the purpose of use is educational or part of research.

The orthophotos from Lantmäteriet are available in several resolutions, the highest being 0.25 m. For this project a resolution of 2 m was considered sufficiently accurate its purpose. This resolution was selected because an orthophoto with a resolution of 0.25 meter would have been 16 times larger than one with a 2-meter resolution. This would make the rendering a lot slower and reduce the performance as well as requiring larger storage space.

The image was delivered split into separate images containing only one band each with the pixel values 0 – 255. Each of the bands was combined using band 1 = red, band 2 = green, band 3 = blue which resulted in a true color image suitable for visual inspection of the results.

4 Methodology
Modifying a basic shapefile containing a road network into a multi attribute network on which route planning can be performed is a process that demands several steps and a lot of different data. Each step and the required data to perform them are described in the following sections.

4.1 Summary of the algorithm
The following chapter is meant to visualize the algorithm and give a brief overview of its entirety using flowcharts. Each step showed in a flowchart is further explained in its corresponding sub section following after section 4.1.

The following coloring is used in the flowcharts:

Blue = shapefile or similar ArcMap file format
Green = operation of some sort
Yellow = decision that affects the outcome
Pink = input raster file
Orange = temporarily stored information
4.1.1 Generic summary of the whole algorithm
A very generic summary of the whole algorithm is shown in Figure 4. Each of the steps in Figure 4 is then summarized once again in another flowchart.

Figure 4: A generic summary of the complete algorithm.

4.1.2 Summary of each step of the algorithm
The additional attributes road class and net type are overwritten to shapefile containing the network geometry using dictionaries. Each polyline is assigned with the corresponding attribute from each shapefile. This is shown in Figure 5.
Figure 5: Step A of the data preparation.

All roads which are not suitable for bicycle usage are removed from the network. A unique AltID is then created for each polyline in the shapefile. This is shown in Figure 6.

Figure 6: Step B of the data preparation.

The initial land cover map in vector format provided by Lantmäteriet is modified by removing layers containing unwanted information such as buildings etc. A new land cover class which represents the area in close proximity to large roads is then added. The land cover map is the reclassified into raster where appropriate weights for each environment type is chosen. This is shown in Figure 7.
Figure 7: Modification of the land cover map.

A buffer is created around each polyline in the shapefile following the chosen parameters, it is linked to the corresponding polyline using the AltID. The buffers are then used to calculate zonal statistics of the area inside the buffer using the land cover map as basis. This generates a table with statistics (the nature index) linked to each buffer AltID which is then overwritten to the shapefile containing the polylines. This is shown in Figure 8.

Figure 8: Nature index calculation.
The slope of each polyline is calculated using the digital elevation map. The coordinates of the starting and ending point of each polyline is used to extract the corresponding z-value which is then used to calculate the slope. This is shown in Figure 9.

![Figure 9: Slope index calculation.](image)

This part of the algorithm constructs costs using the nature and slope index. To use the indexes as weights in cost functions they first have to be normalized. After being normalized the indexes can be used in different cost functions resulting in a list of attributes which can be applied to each polyline. This is shown in Figure 10.

![Figure 10: Step A of the cost calculation and network building.](image)
After the cost attributes have been applied to all polylines the network dataset is built using that. The corresponding cost attribute is used as the only cost in the network, this is shown in Figure 11. A network dataset can only use the cost function it was built with.

![Diagram](image)

**Figure 11: Step B of the cost calculation and network building.**

### 4.2 Network data preparation

The road network data downloaded covered the entire study area. Three separate shapefiles sharing the majority of information was downloaded. Each contained one key element of information which the others did not. The information not included in all shapefiles was an attribute containing road class, another one storing net type and the last one which contained the network geometry. These shapefiles and their corresponding unique information are described in Table 1.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Information used from the dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td>Funktionell vägklass</td>
<td><strong>Road class</strong>: an attribute containing a classification value of how important the road is according to Trafikverket. This corresponds to size and type of road quite well.</td>
</tr>
<tr>
<td>Vägtrafiknät</td>
<td><strong>Net type</strong>: an attribute containing information of which roads are dedicated bicycle paths and which are car roads.</td>
</tr>
<tr>
<td>Transportnät vald</td>
<td><strong>Network geometry</strong>: information needed to build a network on which route planning can be performed.</td>
</tr>
</tbody>
</table>

*Table 1: Network datasets downloaded and the attributes used from each dataset.*

Some information was considered redundant and therefore removed from the attribute table of the shapefiles. After this information had been removed the attributes road class and net type needed to be transferred to the shapefile containing the network geometry. The shapefiles contain an attribute called RLID which identifies each road but this ID was not unique, at some places several RLIDs existed because
the polylines corresponding to the same road in reality had been split in two inside the dataset. This resulted in multiple instances of RLIDs which made joining of the tables to transfer attributes in ArcMap impossible.

To solve this three fields (AltID, KLASS, NÄTTYP) were added to the attribute table of the shape file containing the network geometry. The values of these fields are then set to their corresponding value acquired form the corresponding shapefile using the following method. First the shapefile containing the information about the road class is iterated through and the road class is stored as a value with the RLID as key inside a Python dictionary. The same thing is done with the shapefile containing the net type but the values are stored in a new separate dictionary. This method works even though the join does not. This is because of the fact that polylines sharing the same RLID are the same road in reality but it has been split into separate polylines in the dataset. Thus both lines sharing the RLID have the same road class in reality as well and all roads with the same RLID are assigned with the same value from the dictionary since they use the same key.

Now when the shapefile containing the network geometry also stores all necessary attributes the roads where you cannot ride a bicycle can be removed from the shapefile. This is done by selecting all polylines with road class 0 to 4 using an SQL query. The selection is the switched and a new shape file is created using the now selected features.

After this a visual inspection of several areas know by the author is preformed to verify that only roads that cannot be by bicyclists are removed. The inspection showed that in most cases the resulting network is correct but there are a few issues. For instance, some roads inside industrial areas are being removed together with a few roads of class 4 that in some cases are fine to bike along despite their class. This is due to the fact that the road class is determined by the roads importance, not its size or characteristics (Trafikverketverket, 2015). Roads inside an industrial area might be classified higher than other roads of the same size due to their logistical importance and if some roads are the only possible path between two locations they might receive a higher class as well.

4.3 Unique IDs

After the data preparation in section 4.2 has been performed the issue with non-unique IDs needs to be solved. Since all polylines now contain the correct attribute data they now only need an identifier which is not shared with any other polyline. The new ID is called AltID in the attribute list.

4.4 Nature index

4.4.1 Land cover map modification

The land cover map is used for the calculation of the nature index. The land cover map is based on GSD-Terrängkartan in vector format. It contains a lot of unnecessary information and to improve results it is necessary to strip away some of this information. Since the map is in vector format it is easy to remove redundant information such as buildings, roads, text etc. which are not used by the algorithm and only contributes to a noisy map. The information is removed by deleting all layers containing unwanted features. In the end the only wanted layer is the one containing land use. Figure 12 illustrates the difference between the map containing the initial information and the map containing only the desired information about land use which is used to calculate environment type.
A problem with GSD-Terrängkarten is that the classification "Open land" is given both to parks and areas near large roads, highways and public buildings such as hospitals, schools etc. This is shown in Figure 13 which contains (1.) the major traffic solution at the intersection of E4 (highway) and road number 275 (major road) and (2.) the city park Rålambshovsparken which both have the same classification even though their environment type is completely different to each other.

This is a major issue since biking along a highway is not very pleasant nor something you desire if you are looking for a route with a high level of nature. To solve this road data containing information about road class is used. An SQL Query is used to select all roads with class 0 to 3 which corresponds to roads of large enough size that they impair the bicycling experience. But this resulted in roads inside tunnels which do not affect bicycles traveling above being selected anyway. A new layer was created containing only tunnels. The new layer is then used to perform a spatial query in combination with a SQL-query which selects all lines of class 0 to 3 which do not overlap with a tunnel. All of the selected features from these queries are used to create a new layer. A 30-meter buffer is created as a single polygon around all features in this newly created layer. Figure 14 illustrates that Klarastrandsleden is buffered whilst the area above Belholmstunneln is not buffered which is the wanted result.
Figure 14: The major road Klarastrandsleden is buffered but the road segment going through the tunnel Blekholmstunneln is not buffered.

The newly created buffer is then used to overwrite all areas it covers in the land cover map. Which are given a new class, “In close proximity of a large road/highway”. As a result of adding this new land cover class twelve million square meters is reclassified.

To be able to calculate the surrounding area of a polyline the land cover map is reclassified into raster data. In this process a new distribution of classes is selected to reduce the number of redundant classes and make the class-values more logical. Figure 15 illustrates which of the old classes that are classified into which new class. 14 classes are reduced to 9 classes in the end.

Figure 15: The values used to reclassify the environment classification map. The left column contains the old classes and their values and the right column contains the new classes and their values. The arrows point out which original class that is reclassified into which new class.
All classes merged into another are considered redundant or only occurs rarely in the map. For instance, recreational buildings are not very common in the study area and the experience of cycling through one is quite similar to the experience of cycling though a low density development area, because of this the reclassification is well-founded.

4.4.2 Weighting each environment type

The weights of each environment type are determined via a two-step process. First each environment type is ranked from most preferable to least preferable when it comes to how nice it is to travel through it by bike. The most preferable environment class is assigned with the nature index of 100, in this case water. Each of the following environments is then evaluated in terms of how many times nicer it is to bike along the one with highest rank. For instance, it is considered five times nicer to bike along water than to bike through high density development, hence the assigned nature index weight of high density development is 20. Each assigned value is used as the pixel value for that category in the raster land cover map. The weighting used is shown in Figure 16.

![Figure 16: The assigned weight for each class in the environment classification map.](image)

4.4.3 Nature index calculation

The first step of calculating the nature index is to generate a buffer for each individual polyline which shares the same AltID to enable linking the correct buffer to the corresponding polyline during calculations. A new shapefile containing the buffers is created during the operation. The buffers are generated with the parameters shown in Table 2.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Used value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Distance</strong> - the distance from the generated polygon border to the line.</td>
<td>30 meters</td>
</tr>
<tr>
<td><strong>Side Type</strong> - determines how the polygon is generated in relation to the input feature.</td>
<td>Full - buffers will be generated on both sides of the line.</td>
</tr>
<tr>
<td><strong>End Type</strong> - the shape of the buffer at the end of line input features.</td>
<td>Round - the ends of the buffer will be round, in the shape of a half circle.</td>
</tr>
</tbody>
</table>
**Dissolve Type** - specifies how overlapping buffers should be handled. | **None** - an individual buffer is created and maintained for each feature, regardless of overlap.

<table>
<thead>
<tr>
<th>Table 2: The parameters used to generate a buffer for each polyline.</th>
</tr>
</thead>
</table>

These values are selected to generate a buffer which represents the environment around a polyline in the most correct way. The 30-meter radius is considered to include the area around a polyline which directly affects the cycling experience. A larger radius could be motivated depending on which level of impact is desirable. The side type full is used since you are affected by the environment on both sides of the road and therefore a buffer needs to be generated on both sides. The end type round is chosen so that the whole area at intersecting lines is considered. This creates some extra overlap among the polygons but it was not considered to affect the calculations negatively in any way. Using the dissolve type none is necessary since each polylines needs to be linked to one unique buffer sharing the same AltID as the polyline. If dissolve was used all overlapping polygons would be merged into one.

The idea is to calculate statistics based on the pixel values of all pixels in the land cover map contained within each buffer and use the mean value of the pixels as the nature index corresponding to that line. The nature index is calculated using the following formula:

\[
\text{Nature index} = \frac{\sum \text{raster values of individual pixels inside buffer}}{\text{number of pixels inside buffer}}
\]

**Formula 1: Nature index formula**

Initially Esri’s own tool Zonal statistics as table was used for this but this tool did not work as expected. On a test area containing 1367 polylines with 1367 matching buffers the output table only contained 821 entities. The reason was that the default Zonal statistics as table cannot handle overlapping polygons. This is a big problem since the dataset in question contains overlapping polygons at all intersections between lines etc. An add-on tool package called Spatial Analyst Supplemental Tools was downloaded to solve this problem. It includes the tool Zonal Statistics as Table 2 which can handle overlapping polygons.

Running Zonal Statistics as Table 2 still only resulted in 868 entities even though the issue with overlapping polygons had been removed. This is what led to the discovery of RLIDs not being unique. At this time of development, the algorithm did not generate individual unique IDs for each polyline. By iterating through the data it was discovered that 499 of the polyline RLIDs existed more than once. Some of the polylines in the dataset were split into two lines sharing the same RLID and Zonal Statistics as Table 2 only handled each ID once which resulted in 868 lines being processed since 1367 – 499 = 868. The solution is to manually generate IDs that were unique for each polyline (see section 4.3). With the algorithm now generating its own AltIDs and the buffers generated once again, based on the AltID, the output table now contains the correct amount of entities. Each entity receives the same AltID as the buffer to enable linking it to the corresponding polyline. The nature index is then added to the correct polylines attribute list using a dictionary in ArcPy.

4.5 **Slope index**

The initial approach of calculating the slope index for a polyline was to evaluate all line segments belonging to that polyline individually and calculate a combined index for the polyline based on those calculations. This method proved to be way to slow because the method arcpy.GetCellValue_management used a huge amount of time for each call. The method takes the coordinates from a point and returns the elevation at the same coordinates in the digital elevation model but it can only handle 10 calls per second resulting in 5 line segments being processed per second which made this method not applicable due to the size of the data set.
Instead each line is evaluated using only the elevation at the starting point and ending point. This method has lower accuracy, especially for longer polylines but is a lot faster. With this method 5 polylines per second are being processed instead of 5 line segments. The method still causes very long calculation times when using extensive datasets but is more manageable.

The slope of each polyline is calculated using the following formula:

\[
\text{Slope Index [\%]} = \frac{\text{elevation}_2 - \text{elevation}_1}{\text{length}} \cdot 100
\]

*Formula 2: Slope index formula*

Where \( \text{elevation}_1 \) is the altitude of the starting point and \( \text{elevation}_2 \) is the elevation at the ending point of each polyline. \( \text{Length} \) is taken from an attribute called SHAPE_LEN which is included in the original data and not calculated during any step of the algorithm.

Since this method only evaluates the elevation in the beginning and the end of each line it assesses roads that go up and then down from a hill incorrectly since the \( \text{elevation}_1 \) and \( \text{elevation}_2 \) will be very similar even though the actual path contains a lot of slopes, this is illustrated in Figure 17. This issue is big with long polylines and almost no issue with shorter ones.

![Figure 17: The difference in the calculated slope index for a road going up and down a hill. On the left the slope index is calculated using the default length of the polyline and on the right the length of the polyline is lower.](image)

The method is also vulnerable to roads that for instance start with a steep slope and then level out. This is because it is the average slope of the polyline that is calculated and if the flat part is long enough in comparison to the slope the calculated average slope will not reflect the reality. This is shown in Figure 18.

![Figure 18: A road initially going uphill but then level out. On the left the slope index calculated using the original polyline length and on the right the slope index calculated for polylines along the same road with lower length.](image)

Another issue causing miscalculations is due to bridges not being included in the digital elevation model. Instead it is the level of the water surface at those coordinates which is given. This causes some of the calculations for polylines passing over bridges to be misleading. Especially polylines that are very short and ends in the middle of a bridge will be calculated as being very steep even though that is not the case in reality.
4.6 Normalizing indexes and assigning net type weight

When both the nature and the slope index has been determined and added to each polyline they need to be adjusted to allow easy weighting between several different attributes at once. This is done via two different normalizations which modifies the indexes so that they range from 0 to 1.

4.6.1 Nature index normalization

The formula used to normalize the nature index (nature index = NI):

\[ NNI_i = \frac{|NI_i - \max(NI)|}{\max(NI) - \min(NI)} \]

*Formula 3: Normalization formula for the nature index*

![Figure 19: Normalization of the nature index plot.](image)

This formula is constructed as an inverted normalization since a high nature index needs to result in a low weight factor for the cost calculations. This means that a nature index that is close to max will be close to 0 and a low nature index that is far away from max will receive a weight close to 1 when they are normalized. This can be seen in Figure 19.

4.6.2 Slope index normalization

The formula used to normalize the slope index (slope index = SI), note that the slope index is used as an absolute value in all aspects of this formula:

\[ NSI_i = \begin{cases} 0, & |SI_i| < 1 \\ \frac{|SI_i| - 1}{10 - 1}, & 1 \leq |SI_i| \leq 10 \\ 1, & 10 < |SI_i| \end{cases} \]

*Formula 4: Normalization of the slope index*
The slope index is ignored if the absolute slope is lower than 1%. If the absolute slope index is between 1 and 10%, the slope index is normalized using the standard formula resulting in a value between 0 and 1. Absolute slope indexes above 10% are assigned with the weight 1 independent of how large the value is. This can be seen in Figure 20.

4.6.3 Net type weighting
The final cost weighting attribute is used to make the route planner regard different road types and is supposed to promote the usage of dedicated bicycle roads. It uses the net type attribute to ensure that car roads are weighted more heavily than bike roads.

Formula for the weighted net type (net type = NT):

\[
WNT_i = \begin{cases} 
1.5, & NT_i = 1 \\
1, & NT_i > 1 
\end{cases}
\]

Formula 5: Net type weighting formula

A NT\textsubscript{i} value equal to 1 corresponds to a car road and value above 1 corresponds to a walkway or bicycle road in the formula above.

4.7 Cost calculation
The normalized indexes are then weighted in seven different ways resulting in seven cost attributes. The different weighting formulas are shown below.

Cost1 represents the overall shortest path but prioritizes bicycle roads above car roads. Cost1 should be used by a user that at all cost want to travel the shortest path between two points.

\[
cost1 = length \cdot WNT_i
\]

Formula 6: Cost1 formula

Cost2 uses the nature index as a very heavy factor when determining the cost for each polyline. This results in a bigger percentage difference in length than most other costs using nature index. Cost2 should be used by someone who do not care very much about the length of the route or travel time and instead
focuses on the setting of the route. Cost2 is mostly geared towards recreational riding but can be used by commuters with a lot of time.

\[ cost2 = \text{length} \cdot \text{WNT}_i \cdot \text{NNI}_i \]

**Formula 7: Cost2 formula**

Cost3 only takes the length and slope index into account. The maximum additional cost due to slope is one additional length which means that a steep slope has twice the cost of the equivalent flat road. Cost3 should for instance be used by someone who are commuting to work but do not take a shower afterwards and therefore want to avoid straining uphill’s.

\[ cost3 = \text{length} \cdot \text{WNT}_i + \text{length} \cdot \text{NSI}_i \]

**Formula 8: Cost3 formula**

Cost4 takes a combination of the slope and nature index as factor when calculating the cost of a polyline. Cost4 should be used by someone who have the same priorities as the user of cost3 but want to follow a route with a little more recreational character.

\[ cost4 = \text{length} \cdot \text{WNT}_i + \text{length} \cdot \text{NNI}_i + \text{length} \cdot \text{NSI}_i \]

**Formula 9: Cost4 formula**

Cost5 calculates a cost depending on the nature index but its influence is much lower than in for instance cost2. A user of cost5 should prioritize having a short route highly but if the additional detour required to travel along a more natural path is only slightly longer the user should be willing to go the extra distance. Cost5 is suitable for commuters.

\[ cost5 = \text{length} \cdot \text{WNT}_i + \text{length} \cdot \text{NNI}_i \cdot \frac{1}{2} \]

**Formula 10: Cost5 formula**

Cost6 uses the nature index as a quadratic factor when determining the cost, this means that the difference between a high and low normalized index will deviate much faster. Much like the user of cost2 a user of cost6 should be willing to take extensive detours to follow a more natural route. In most cases, cost6 will require an even bigger detour than cost2. Cost6 is very much geared towards recreational riding.

\[ cost6 = \text{length} \cdot \text{WNT}_i \cdot \text{NNI}_i^2 \]

**Formula 11: Cost6 formula**

Cost7 takes no regard to the length of the path at all. It will find the sum of normalized nature indexes that is the lowest. This means that it will prefer as few polylines as possible with the highest nature index, and in the end it is more likely to select polylines with a higher length. A user of cost7 should have the same preferences as a user of cost6.

\[ cost7 = \text{NNI}_i \]

**Formula 12: Cost7 formula**
4.8 Cost examples

To illustrate the difference between these cost functions a table containing the calculated cost of each function for two example polylines and their percentage difference is presented, see Table 3. The percentage difference column presents how many times larger the cost of the polyline in example 1 is in comparison to the cost of the polyline in example 2.

The two example polylines are:

- Example 1: 100 meter bicycle road with nature index 40 and slope index 5.
- Example 2: 200 meter bicycle road with nature index 85 and slope index 2.

<table>
<thead>
<tr>
<th>Cost function</th>
<th>Cost example 1</th>
<th>Cost example 2</th>
<th>Percentage difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>cost1 = length \cdot WNT_i</td>
<td>100</td>
<td>200</td>
<td>50%</td>
</tr>
<tr>
<td>cost2 = length \cdot WNT_i \cdot NNI_i</td>
<td>70.59</td>
<td>35.29</td>
<td>200%</td>
</tr>
<tr>
<td>cost3 = length \cdot WNT_i + length \cdot NSI_i</td>
<td>144.44</td>
<td>222.22</td>
<td>65%</td>
</tr>
<tr>
<td>cost4 = length \cdot WNT_i + length \cdot NNI_i + length \cdot NSI_i</td>
<td>215.03</td>
<td>257.52</td>
<td>84%</td>
</tr>
<tr>
<td>cost5 = length \cdot WNT_i + length \cdot NNI_i \cdot \frac{1}{2}</td>
<td>135.29</td>
<td>217.65</td>
<td>62%</td>
</tr>
<tr>
<td>cost6 = length \cdot WNT_i \cdot NNI_i^2</td>
<td>49.83</td>
<td>6.23</td>
<td>800%</td>
</tr>
<tr>
<td>cost7 = NNI_i</td>
<td>0.71</td>
<td>0.18</td>
<td>400%</td>
</tr>
</tbody>
</table>

Table 3: Cost of two polylines given different cost functions and the percentage difference between them.

4.9 Applying the costs

The cursor containing all the polylines is iterated over and all of the costs are calculated for each polyline. The algorithm then applies each of these costs as a unique attribute for that polyline.

4.10 Network building

In ArcMap a network has to be built for each different cost weighting configuration presented in section 4.7. The networks are built using the Network Analyst extension in ArcMap 10.2. CostX is then chosen as the default cost attribute for the corresponding network instead of length. The built network is ready to perform route calculations between points of choice in the map using the costX as cost parameter. To use different weights, the user has to change the network on which the calculations are performed to the network built with the corresponding cost function.

5 Results

5.1 Attribute table of the unbuilt network

An extract of the attribute table belonging to the unbuilt network can be seen in Figure 21. The meaning of each field is explained in Table 4.
**Figure 21: Extract of the attribute table belonging to the unbuilt network.**

<table>
<thead>
<tr>
<th>Field</th>
<th>Data type</th>
<th>Value</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>FID</td>
<td>Integer</td>
<td></td>
<td>Not used, created by ArcMap.</td>
</tr>
<tr>
<td>Shape</td>
<td>Explains the geometry type of the feature</td>
<td>Polyline ZM, indicates that it is a 3D polyline feature.</td>
<td>Used by ArcMap internally.</td>
</tr>
<tr>
<td>RLID</td>
<td>String</td>
<td></td>
<td>Used to map dictionary values to the corresponding line before AltID is assigned.</td>
</tr>
<tr>
<td>SHAPE_LEN</td>
<td>Double</td>
<td>0 – maximum length of a polyline</td>
<td>Displays the length of the polyline. Used to calculate traveling cost in a built network.</td>
</tr>
<tr>
<td>KLASS</td>
<td>Integer</td>
<td>0 – 9, 0 is a road of highest importance and 9 is a road of lowest importance. 99 = NoData</td>
<td>Used to determine the road class of a polyline. This is done to remove roads too large to travel by bike form the network.</td>
</tr>
<tr>
<td>NÄTTYP</td>
<td>Integer</td>
<td>1 = Car road 2 = Bicycle road 4 = Walkway</td>
<td>Used to prioritize bike roads above car roads by giving them a reduced cost in the network. Referred to as net type in this paper.</td>
</tr>
<tr>
<td>AltID</td>
<td>Integer</td>
<td>Unique values ranging from 0 – number of features</td>
<td>Used to connect several tables and shapefiles containing different data and enable linking information to the correct feature.</td>
</tr>
<tr>
<td>NatureIndx</td>
<td>Float</td>
<td>Values ranging from 15 – 100, dependent on the pixel values used in the environment classification map.</td>
<td>Used to represent the average environment around a road. The value is later taken into consideration when determining the cost of traversing that polyline.</td>
</tr>
</tbody>
</table>
Table 4: The attributes of the finished shapefile explained.

5.2 Visualization of the calculated Nature and Slope Index
Figure 22 shows a representation of the unbuilt network using the value of the nature index as display factor. A green line represents a higher nature index which is better.

Figure 22: Visualization of the network using the Nature Index to determine the display color.
Figure 23 shows a representation of the same network in the same location but visualized using the slope index instead. The algorithmic color ramp is mirrored at the value zero since the cost calculation formulas uses the absolute value of the slope index. A white line is a flat line with a slope index close to zero which most preferred.

Figure 23: Visualization of the network using the Slope Index to determine the display color.
5.3 Issues with the finished network

Since a lot of polylines have been removed from the network based on their road class to ensure that only roads where you can ride a bicycle remain some gaps in the network have occurred. One example of this is shown in Figure 24 which contains an image of the network in Hornstull, central Stockholm. The area of interest is marked in red. This loss of connectivity occurs at places where a polyline with road class 4 has been removed. In reality a bicycle path does exist at this particular location but it is not included in the dataset for some reason. This error is not a major issue and do not occur very often but it does affect routes in some manner.

![Figure 24: Loss of connectivity due to removing polylines of a certain road class.](image)
5.4 Routes between the same locations using different cost networks

Figure 25, Figure 26 and Figure 27 show all least cost paths between two locations. Each path is using one of the seven cost functions described in this paper. It is clear that different weightings result in different routes.

Cost functions with low regard to the nature or slope index tend to follow the shortest path in most cases but sometimes deviate a little. On the other hand, cost functions that weight those indexes heavily almost never follow the same route as the shortest path. The resulting route is more than twice as long in some cases. This is not unexpected behavior since these cost functions are created to satisfy recreational riding where the path leading to the destination is more important than the distance it takes to get there.

In Figure 25 all seven costs follow two routes with minor deviations in some places. Cost1, 3, 4 and 5 all go along an almost straight path traveling through the high density development with highly trafficked streets. Cost2, 6 and 7 all choose the path that runs along the water.

Figure 25: Routes between Fridhemsplan and Rålamshovsparken using differently weighted networks.
In Figure 26 the amount of different paths is more diverse than in Figure 25. Here all paths divert from the shortest one (cost1), which in this case pretty much only passes through heavily trafficked high density development. The paths selected using cost3, 4 and 5 passes a park along the way and is in fact shorter when it comes to distance only but parts of the route are not following a bicycle path which is why the cost1 path goes along Hornsgatan and Götgatan.

The path selected by cost6 might seem odd at first but the choice is made due to two factors, the first one being that Södersjukhuset, which is the hospital the route is passing in the middle, is classified as open land in the land cover map which gives it a high nature index. The other being that since the normalized nature index is squared in this cost-function polylines with a low nature index are very unfavorable to travel along. That is why it converges with the cost3, 4 and 5 path which goes along a mid-low nature index polyline whilst the cost2 and 7 path follow the water and then turn towards the destination following Götgatan which has a very low nature index.

Figure 26: Routes between Bondegatan, close to Medborgarplatsen on Södermalm and Hornstull using differently weighted networks.
Routes at smaller distances tend to not deviate very much but they are the better used when visualizing the differences since the details in the orthophoto are easier to see at lower scales. Figure 27 shows the seven routes between two locations at a larger distance. It is clear that the different cost algorithms follow different paths to a larger extent at this level, even though the differences might be small they all deviate from each other at least once along the way. It is also evident that costs which prioritizes the nature index tend to follow green patches.

![Legend](image)

Figure 27: Paths between Älvsjö south of Stockholm and Skärmar brink close to Gullmarsplan. This figure shows a larger deviation between different paths than the other figures which are at smaller distances.

5.5 Range difference between cost1 network and cost2 network

The difference between different network traveling costs is displayed best using service areas. Figure 28, Figure 29 and Figure 30 all contain two service areas each, one showing the range 0-1000 cost units from the center when traversing the cost1 network and the other the same range traversing the cost2 network with the equivalent range of 0-590. The blue polygons represent the cost1 network which is only based on network distance. The pink polygons represent the cost2 network which is based on weighting the distance using the nature index.

To be able to fairly compare the range of two different networks using service areas one has to be mathematically converted to match the other. In this case the mathematical relation between cost1 and cost2 was used with a simulated constant nature index of 50 in the following relation formula.

\[
\text{cost2} = \frac{\text{cost1}}{\text{NNI}_i}
\]

Given a Nature Index of 50, which represents an environment type in-between mid and low density development, we get \(\frac{\text{NNI}_i}{\text{NNI}_i} = 0.59\), see Formula 3. This means that we should use cost limits that are 1.69
times larger for the polygons. This will result in a fair representation of their range using the actual equivalent costs. The Resulting differences between cost limits for the service areas using this ratio can be seen in the legend of Figure 28.

Using this simulated value as the nature index the following costs are calculated for the same stretch of road, in this case a 100 meter car road with Nature Index 80:

\[
\begin{align*}
\text{cost}_1 \text{ (simulated)} &= 100 \cdot 1 \cdot 0.59 = 59.00 \\
\text{cost}_2 &= 100 \cdot 1 \cdot \left(\frac{180 - 100}{100 - 15}\right) = 23.52
\end{align*}
\]

A 100 meter car road with Nature Index 25:

\[
\begin{align*}
\text{cost}_1 \text{ (simulated)} &= 100 \cdot 1 \cdot 0.59 = 59.00 \\
\text{cost}_2 &= 100 \cdot 1 \cdot \left(\frac{125 - 100}{100 - 15}\right) = 88.24
\end{align*}
\]

Figure 28 show a clear difference between the ranges using these two networks at Gamla stan in Stockholm. This is an area where paths along the shore have a high nature index which means that the blue polygon will range a lot further at these locations. It is clear that the algorithm using the nature index heavily prioritizes paths along the water. Where the polygons go through urban environment their range is pretty equal which indicates that the average nature index along this path is around 50.

---

**Figure 28:** Two overlapping service areas with the center in Gamla Stan, central Stockholm. The blue polygons represent the range using cost2 and the pink represent the range using cost1 with a simulated constant nature index of 50.
Figure 29 shows two service areas using the same cost restrictions as above but at a location at the very edge of Stockholm. In this image it is very clear that the pink polygon which uses a simulated constant nature index ranges further inside the city whilst the blue polygons which regards the nature index ranges around one kilometer further when passing through Haga parken in the northern part of the image.

Figure 29: Two overlapping service areas with the center at Norra Stationsgatan, the outer boundary of central Stockholm. The blue polygons represent the range using cost2 and the pink represent the range using cost1 with a simulated constant Nature Index of 50.
Figure 30 uses the same settings for service areas as the two images above but the center is located in central Östermalm which is an area consisting almost entirely of high density development with a low nature index. In this case the pink polygon which does not take the nature index into account ranges further than the blue one in almost all directions.

![Image of Figure 30](image)

**Figure 30** Two overlapping service areas with the center in Östermalm, central Stockholm. The blue polygons represent the range using cost2 and the pink represent the range using cost1 with a simulated constant Nature Index of 50.

## 6 Discussion

The data used in this paper do not hold perfect quality. Since the network dataset lacks information about which car roads that can be used by bicyclists a manual transfer of information and filtering of the data has to be performed which causes errors in the network making it less reliable and cause reduced connectivity at some points. Results would be better if something else than road class had been used to remove roads where bicycles cannot go. This is error arises because the road class is not determined by the actual size or type of road, instead it is chosen based on the roads importance which causes some roads that are usable by bicyclists to be removed.

There is also an issue with big difference in polyline lengths. Since most of the polylines in the dataset used have their actual length between intersections as geometrical length they tend to vary a lot. This causes issues with cost functions that do not regard length, such as cost7 which because of this will for instance choose a 2 km polyline with nature index 40 above four polylines with length 200 m and nature index 80. The length of polylines is also a problem when calculating the slope index since it is divided by the shape length which will remove a lot of detail if the polyline is long.
Because the nature index is based solely on the land cover map provided by Lantmäteriets GSD-Terängkartan any quality defects in the map are directly transmitted onto the nature index. The most obvious issues with the land cover map is that parks are assigned with the same classification as areas around large roads. This class is also shared by public buildings do not have their own class which means that hospitals, schools etc. are also classified the same way. To improve this a more detailed land cover map has to be developed. It would be preferable if this map would have more environment classes to better be able to separate different environment types and then manually merge classes if so is wished. The level of detail in the map is fine but more distinction between environments types is needed. It could be possible to combine that map with a classified satellite image which would give a broader spectrum of information to solve this issue. For instance, green avenues are often quite pleasant but in the land cover map used they are classified the same as the development type that is adjacent. A combination like the one mentioned above would result in this type of environment getting a bit better score.

Both Aiello et al. (2014) and the Fietsersbond route planner use some sort of crowdsourcing to generate data of which roads are more pleasant etc. and as of today this seems to be the best way of acquiring data of this kind. An implementation of crowdsourcing in this algorithm would require a bit of a structural change and some way of collecting the data. It is possible to implement if the collection of data is successful and as long as the results are in form of a table with a value connected to an ID matching the corresponding polyline or in the form of a map such as the one used by Aiello et al. This biggest challenge would be to setup a way of collecting a sufficient amount of data with good quality.

After the land cover map is finished it is weighted so that each environment type receives its own corresponding nature index. No scientific evidence is used during this process due to lack of research in the area which means that the weighting of each environment class is very subjective and can possibly differ a lot between individuals. It would have been possible to make a small survey with in the course of this project to get some data which would allow a better weighting, but since the objective was to create a concept this were not considered important enough to perform the actual survey.

To generate an area which is supposed to represent the neighborhood which affects the bicyclist a buffer is created for each polyline. In this paper a radius of 30 meters is selected but this selection is made without using any scientific basis which makes it questionable if this really is the best fitting radius to represent the area. The radius which affects the experience is also very different depending on the environment type. For instance, a highway contributes to noise and air pollution well over 30 meters away, whilst a suburban low density development area has a very low radius which affects the bicyclist. During the calculation of the nature index the mean of all pixels inside the corresponding buffer is used to determine its value. It is possible that other statistical measurements would have given a better result but this was not investigated in this paper and would probably require more careful weighting of environment class nature indexes if measurements like standard deviation were to be used.

The biggest issues with calculating the slope index are related to the length of each polyline. If the dataset had been divided into more polylines with lower length most of its inaccuracies could have been solved. If the dataset had included Z-geometry the problem with bridges getting the water level as height would also be removed or if a digital surface model containing the bridges were used to extract the Z-value.

The cost functions are one aspect that can be developed a lot more to achieve results more in line with what is desired. The formulas used in this report all work but more careful selection of ratios and composition of the formulas would enable more controlled route planning. Another aspect that could be regarded is the percentage difference in length between the calculated shortest path and the path selected by the algorithm. Something like a 40 % length difference limit could give better results and remove some of the ridiculously long detours created in some cases using certain cost functions. It would not be possible to implement on the networks used in this report but could give more reasonable results if implemented in a similar service.
Due to a low knowledge of the network analyst extension the slope index is regarded as an absolute value during cost calculations. The network analyst tool is able take the direction of the line into account using a custom script which means that the slope index actual value could have been used and thus given negative slopes a lower cost but this was discovered too late during the project to implement a solution like this. A better and more realistic result would be achieved if the direction of the slope would be taken into consideration.

7 Conclusion and further work

7.1 Conclusion
The resulting cost functions do find routes that fulfil the criteria they are supposed to. Routes that weight the nature index highly do actually go thorough green areas and along water in a much larger extent than the shortest path. Sometimes they tend to be quite a lot longer and other times they are about the same length but follow different paths. The slope index works good as well and prefers flat roads but if it is weighted highly the resulting routes were very winding and follow an unnatural path. This is why the slope index is weighted quite low in all the cost functions.

Some of the cost functions used are better suited for recreational riding but the algorithm can be tweaked toward commuters as well. More work on the cost functions is required if a service like this would be released targeting commuters. This is because the routes created need more reliable with a smaller length difference when used by commuters.

To get the best networks with the most accurate and detailed data some sort of crowdsourcing seems to be the most suitable approach. As long as the response is sufficient and the data is on the correct form it is possible to implement crowdsourced data in an algorithm like this one. Data of this kind could help increase the accuracy of which locations should receive a higher nature index. It might even be possible to implement a different index storing beauty, much like Aiello et al.

The findings showed that Stockholm is not the best city to try this kind of algorithms due to its topography with a lot of water and a few bridges which works like bottlenecks and all routes has to go via the bridge in order to reach its destination. This means that the routes prioritizing nature will be forced to go through areas with low nature index in order to go over the bridge. A more thorough testing of the algorithm is better accomplished in a city with a more even distribution of roads and a smoother topology.

7.2 Further work
This concept can be further developed in many ways which are described in the following sections. Apart from this a quantitative analysis of different cost functions is an interesting addition that would result in statistics of range differences etc. between different cost functions. A service area analysis like the one in section 5.5 where all cost functions are included is an interesting way of approaching a comparison as well.

7.2.1 Implementing the network in a web-based service
An important step in order to make a service like this actually usable is to launch a web-based service which allows the information to be easily accessed and enables everyday usage. There are several viable options to accomplish this which are described in the following sub-chapters.

7.2.1.1 ArcMap-web
The easiest way om implementing this algorithm online is to just export the built networks to an ArcMap-web client and let the users preform route planning through that interface. It would enable access for some people but the user base would probably be quite low.
This approach would require someone to acquire the data and perform the same operations as described in this paper for each of the regions where publishing this service is of interest. Which is far from an automated process at this time. This would also mean that all the limitations of the Network Analyst tool inside ArcMap would persist. For instance, a unique network would have to be built for every different attribute weighting configuration.

7.2.1.2 Google Maps
An implementation in Google Maps would with no doubt generate the biggest possible user base and with Google already in possession of a very well-functioning bicycle network with good coverage the implementation would only require adding additional attributes to all lines in their network, a process which is possible to automate, together with a modification of their path finding algorithm so that it could take the additional attributes into account.

Google are also in possession of all the data required to calculate the equivalent of a nature and slope index. The nature index could be calculated using their satellite images with some sort of classification to create the equivalent to the land cover map used in this algorithm all over the globe. Height data is already visualized in Google Maps but not used during route calculation.

7.2.1.3 Standalone web-client
It is also possible to develop a standalone web-client, much like the Fietsersbond, but this would require a large amount of work. A standalone web-client introduces a large amount of freedom when choosing how to implement path finding algorithms, networks etc. but it is a really demanding task.

7.2.2 Live weighting of the attributes
The Network Analyst extension inside ArcMap only supports pathfinding using a cost assigned to a polyline when the network was built. It is possible to perform some scripting to create more complicated cost functions but it is not possible for the user to make live inputs to change the way the cost of traversing the network is calculated which is a desirable functionality in this kind of route planner.

Live weighting of the attributes could be implemented using a pathfinding algorithm which uses a mathematical formula taking several attributes into account to determine the cost of each polyline when finding the route. Instead of only summarizing the fixed cost assigned when the network was built. If such an algorithm is used it is possible to change the weighting configuration of the formula uniquely each time for every user.

7.2.3 Adding additional attributes
There are a lot more possible attributes which could be taken into consideration when determining the optimal route. For instance, an attribute who differs between tarmac and gravel bicycle paths would make the life of racer cyclists a lot easier. Another attribute that is more along the line of finding a more pleasant route is an attribute with traffic data. Traffic data would allow the algorithm to steer clear of heavy trafficked roads and prefer smaller ones which are more safe and relaxing. The traffic data could be stored as an attribute for each polyline and then be used by the cost functions to weigh heavily trafficked roads higher. Data of traffic accidents involving bicyclists could also be implemented, for instance using a system of point penalties like the one used by the Fietsersbond to avoid traffic lights.
References


Kamminga, J. (2016). Email interview regarding Fietsersbond routeplanner. [Interview – 6/5 2016] U


Trafiken. (2016). Reseplanerare. [Online – used 16/5 2016]
Available at: http://www.trafiken.nu/stockholm/reseplanerare/

Available at: http://www.trafikverket.se/TrvSeFiler/Foretag/Bygga_och_underhalla/Vag/Dataproductspecifikationer/Trafiknat/Vag/vagnat.pdf

Available at: http://www.trafikverket.se/resa-och-trafik/vag/Trangselskatt--infrastrukturavgifter/trangselskatt-i-stockholm/