Pareto-Pagerank algorithm

Olli Tuhkanen

MSc thesis
August 2022

Reviewers:
Ph.D. Yury Nikulin
Prof. Marko Mäkelä

DEPARTMENT OF MATHEMATICS AND STATISTICS

The originality of this thesis has been checked in accordance with the University of Turku quality assurance system using the Turnitin OriginalityCheck service

UNIVERSITY OF TURKU
Department of Mathematics and Statistics

TUHKANEN, OLLI: Pareto-PageRank algorithm
MSc Thesis, 52 pages, 21 appendix pages
Applied Mathematics
August 2022

PageRank algorithm is a powerfull graph analysis tools. This algorithm has been used in large variety of different applications. This thesis combines weighted PageRank with multiobjective optimization techniques producing a tool called Pareto-PageRank algorithm. This extension of the original PageRank allows it to be used in more demanding graph vulnerability analyses.

This thesis covers the theory behind PageRank and multiobjective optimization. After the theory, the Pareto-PageRank algorithm itself is introduced and it's Python implementation is tested on real world road network data.

Keywords: PageRank, Graph theory, Pareto-efficiency, Vulnerability analysis.

Contents

1 Introduction

Road networks are a vital aspect of modern societies. People and goods are transformed around cities and countries via roads. These roads connect intersections and cities together. This kind of connection network is usually called *a graph*. Roads do not form the only vital transportation graph. Electricity and water grids are also examples of such graphs. Maintaining and improving these graphs is expencive and some decisions must be made to choose which parts of the network are more important than others. This ranking can be hard and this is why vulnerability analysis algorithms have been used to rank vertices and arcs of the graph by their importance.

The vulnerability analysis of a graph analyses the structure of the graph and pinpoints the most important vertices or arcs in a graph. Classical vulnerability analysis methods take into account the structure of a graph and possibly some attribute but they lack the ability to utilize multible attributes at the same time. In this thesis graph analysis algorithm, *PageRank* is combined with ideas from multiobjective programming to produce a ranking algorithm that can take advantage of multible available attributes. The goal is to produce a set of solutions that give valuable information to the *decision maker*. The provided algorithm is named *Pareto-PageRank* as it combines elements of *Pareto-optimality* with weighted PageRank.

Data analysis and mathematical optimization are close but still different fields of study. While data analysis tries to reveal some useful information from data, mathematical optimization tries to find a solution that fullfills lots of restrictions and expectations defined by the optimization problem itself. The research objective of this thesis is to combine mathematical optimization and data analysis to provide a working tool for multiobjective analysis of graphs and along the way demonstrate that this kind of mixture of techniques is possible.

This thesis covers the idea and theory behind the Pareto-PageRank as well as some tests on real world road network data. After the theory behind the algorithm is covered, pseudocode implementation is provided to the reader. Different scalarization methods are compared and discussed and the performance of the algorithm is analysed. Even though the tests were carried out on road network data, the algorithm itself be used with any arbitrary attributes.

2 Markov chains

The focus of this chapter is to take a brief look at the theory behind PageRank which is the main method used in this thesis. *Markov chain* is a stochastic process containing a finite set of *states* S and a transition probability matrix M. The probability of each state depends only on the previous state. For stochastic matrix the element m_{ij} describes the probability of moving from state i to state j. From each state the probablities for each possible transition must sum up to 1, so $\sum_{j=1}^{|S|} m_{ij} = 1$ for all $i \in \{1, \ldots, |S|\}$.

2.1 Convergence of Markov chains

The convergence of a Markov chain depends on the properties of the transition probability matrix. Next we take a look at these properties and an example of a converging Markov chain.

Definition 2.1. Markov chain is *homogenous* if the transition matrix M does not depend on iteration t.

Definition 2.2. Markov chain is *irreducible* if it is possible to move from any state to any other state with a finite set of transitions.

Definition 2.3. Markov chain is *ergodic* if it is irreducible and its transition matrix M is stochastic.

Theorem 2.1. Markov chains that are homogenous and ergodic will converge to an equilibrium called *steady-state* when the transition matrix is applied $r < \infty$ times to the initial state x_0 [6].

Example 2.2. Suppose a system presenting the location of a person. In the system there are three locations or states the person can be in: *home*, *university* and *restaurant*. We define these states to have indexes 1, 2 and 3 respectively. We define the transition matrix as follows:

$$M = \begin{bmatrix} 0.7 & 0.2 & 0.1 \\ 0.4 & 0.3 & 0.3 \\ 0.4 & 0.4 & 0.2 \end{bmatrix}.$$

Each row and column represents the state the person can be in. The element m_{ij} represents the probability of moving from state i to state j. So we can see that probability from going from university to home is 0.4 and probability of staying in restaurant is 0.2. As matrix M is homogenous and ergodic, the steady-state of the location of the person can be reached by applying M to the initial state until the state converges.

With a given initial state we can calculate the probability of each possible state the person can be in after t iterations. Let the initial state be $x_0 = (1, 0, 0)$ which means that the person is at home. Table 1 shows the probability vector of the person being in any of the possible states with chosen initial state.

Iteration	Calculation	Result $\approx$
0	x_0	$(1, 0, 0)$
1	$x_0 \cdot M$	$(0.7, 0.2, 0.1)$
2	$x_0 \cdot M^2$	$(0.61, 0.24, 0.15)$
3	$x_0 \cdot M^3$	$(0.583, 0.254, 0.163)$
4	$x_0 \cdot M^4$	$(0.5749, 0.258, 0.1671)$
5	$x_0 \cdot M^5$	$(0.57247, 0.25922, 0.16831)$
$\vdots$	$\vdots$	$\vdots$
10	$x_0 \cdot M^{10}$	$(0.571431, 0.259739, 0.16883)$
$\vdots$	$\vdots$	$\vdots$
100	$x_0 \cdot M^{100}$	$(0.571429, 0.259740, 0.168831)$

Table 1: Iteration of Markov chain with initial value $(1, 0, 0)$

Next consider the same transitions but with different initial state. This time we know for certain that the person is not at home. There is 50% change that the person is at University and 50% change the person is in the restaurant. So the initial state vector is $x_0 = (0, 0.5, 0.5)$. Table 2 shows the results for this initial state.

Iteration	Calculation	Result $\approx$
0	x_0	$(0, 0.5, 0.5)$
1	$x_0 \cdot M$	$(0.4, 0.35, 0.25)$
2	$x_0 \cdot M^2$	$(0.52, 0.285, 0.195)$
3	$x_0 \cdot M^3$	$(0.556, 0.2675, 0.1765)$
4	$x_0 \cdot M^4$	$(0.5668, 0.26205, 0.17115)$
5	$x_0 \cdot M^5$	$(0.57004, 0.260435, 0.169525)$
$\vdots$	$\vdots$	$\vdots$
10	$x_0 \cdot M^{10}$	$(0.571425, 0.259742, 0.168833)$
$\vdots$	$\vdots$	$\vdots$
100	$x_0 \cdot M^{100}$	$(0.571429, 0.259740, 0.168831)$

Table 2: Iteration of Markov chain with initial value $(0, 0.5, 0.5)$

In this example both initial states converge to the same result. Convergence of Markov chains is a property of the process, not the initial state. If the initial state is a stable-state then no further convergence happens. But with other initial states converging happens. If we choose $x_0 = (0, 0, 0)$ then it does not matter how many times we apply the transition matrix to this state, we still get the same vector as a result. If we choose $x_0 = (1, 1, 1)$ then $x_{10} \approx (1.714, 0.779, 0.507)$. In this vector value of each element is 3 times the value of corresponding element after 10 iterations when using initial value $x_0 = (1, 0, 0)$. It should be noted that if the elements of x do not sum up to 1 then x is not a stochastic vector anymore and so it does not represent the location probabilities. However it is possible to use non-stochastic initial values on different use cases.

2.2 PageRank

PageRank is a Markov chain process originally used to rank web pages on search engines. PageRank ranks the importance of web pages by trying to predict on which page a random web surfer would be after t iterations. The process can be seen as follows. A graph $G = \{V, A\}$ consists of vertices V which are web pages and arcs A which are links pointing from a website to another one. On each web page there is some probability d for the surfer to click one of the links to move to another page and probability $1 - d$ to stay on the current page. The task is to rank all the pages $v_i \in V$ of the graph based on the probability that the user is in the page v_i after t iterations.

Simple PageRank of vertex v_i can be calculated as:

$$PR_{t+1}(v_i) = 1 - d + d \cdot \sum_{v_j \in M(v_i)} \frac{PR_t(v_j)}{|L(v_j)|}, \tag{1}$$

where $M(v_i)$ is the set of vertices that have an arc pointing to v_i, $L(v_j)$ is the set of outgoing arcs from vertex v_j and d is an adjustable parameter as described before. In [17] and [23], it is recommended to use value $d = 0.85$.

Even though PageRank was originally developed to rank web pages as the name suggests, the algorithm can in fact be used on also other types of graphs. In [18] PageRank is used to rank scientific articles based on references. PageRank was compared against a common *h-index* measurement and it was proposed that PageRank was a "fairer" metric than h-index. Key users in online social network were identified with Pagerank in [8]. In [4] and [24] PageRank was used for topological importance analysis. Similar to this thesis [4] focuses on road sensitivity analysis, identifying the most important nodes from the road network of Manila. In [24] the focus is on risk analysis of electric power, water supply, natural gas and oil, transportation and telecommunication networks in China. PageRank is even used to rank UFC-fighters [22].

To extend the usage of PageRank it is possible to implement a weighted version of the algorithm [23]. Weighted PageRank can be calculated by introducing a weight for each arc. This way the rank distributed is subject to the PageRank of the vertex and weight of the arc in the form

$$PR_{t+1}(v_i) = 1 - d + d \cdot \sum_{v_j \in M(v_i)} \frac{PR_t(v_j) \cdot c(a_{ji})}{|L(v_j)|}, \tag{2}$$

where $c(a_{ij})$ is the cost factor of arc a_{ij}. It should be noted that values of function $c(a_{ij})$ can not be chosen from an arbitrary interval. Formulation (2) reduces to (1) if we choose $c(a_{ij}) = 1$, for all $a_{ij} \in L(v_i)$ and 0 otherwise. If we modify this parameter without any other correction, our transition matrix M will not be a stochastic matrix anymore. This is where we introduce an artificial vertex v_0. If we choose $c : A \to [0, 1]$ we can distribute all of the "unused" PageRank to v_0. This vertex then distributes its PageRank uniformly to all other vertices.

The augmented transition matrix in this situation is

$$M = \begin{bmatrix} 0 & \frac{c(a_{12})}{|L(v_1)|} & \cdots & \frac{c(a_{1|V|})}{|L(v_1)|} & 1 - \frac{\sum\limits_{v_j \in L(v_1)} c(a_{1j})}{|L(v_1)|} \\ \frac{c(a_{21})}{|L(v_2)|} & 0 & \cdots & \frac{c(a_{2|V|})}{|L(v_2)|} & 1 - \frac{\sum\limits_{v_j \in L(v_2)} c(a_{2j})}{|L(v_2)|} \\ \vdots & & \ddots & & \vdots \\ \frac{c(a_{|V|1})}{|L(v_{|V|})|} & \frac{c(a_{|V|2})}{|L(v_{|V|})|} & \cdots & 0 & 1 - \frac{\sum\limits_{v_j \in L(v_{|V|})} c(a_{|V|j})}{|L(v_{|V|})|} \\ \frac{1}{|V|} & \frac{1}{|V|} & \cdots & \frac{1}{|V|} & 0 \end{bmatrix}, \tag{3}$$

where vertex v_0 is represented by the last row and column. So the corrected version of the PageRank can be calculated as

$$PR_{t+1}(v_i) = 1 - d + d \cdot \left(\sum_{v_j \in M(v_i)} \frac{PR_t(v_j) \cdot c(a_{ji})}{|L(v_j)|} + \frac{\sum\limits_{v_i \in V} 1 - \frac{\sum\limits_{v_k \in V} c(a_{ik})}{|L(v_i)|}}{|V|} \right). \tag{4}$$

From M it can be seen that the elements of each row all sum up to 1 so M is a stochastic matrix. With the above method we ensure that no rank "leaks" from the network and the total rank of the network does not approach infinity.

3 Multiobjective optimization

In this thesis the usage of PageRank is extended to a multiobjective case. The technique for this purpose is borrowed from multiobjective optimization. In this chapter the consept of multiobjective optimization as well as some classical methods of solving such problems are introduced.

Multiobjective optimization is a field of mathematical optimization consisting of solving optimization problems with multible objective functions.

These multible objective functions generally do not attain their optimum point with same decision variable values. This is why the consept of optimality must be extended. Multiobjective optimization problems are formally defined as

$$\min f_1(\boldsymbol{x}), \dots, f_{|F|}(\boldsymbol{x}) \tag{5}$$

$$\text{s.t. } \boldsymbol{x} \in X, \tag{6}$$

where X is the set of feasible solutions and $|F|$ is the number of objective functions.

3.1 Pareto efficiency

As there are multiple objective functions in (5), it is possible that each of these have a different optimal solution. An optimal solution for one objective might not be an optimal one for another. This is where *Pareto efficiency* is used to extend the definition of optimality.

Definition 3.1. A point $\boldsymbol{x} \in X$ is called *Pareto efficient* if there exists no solution $\boldsymbol{x}^* \in X$ that $f_i(\boldsymbol{x}^*) \leq f_i(\boldsymbol{x})$ for all i and $f_j(\boldsymbol{x}^*) < f_j(\boldsymbol{x})$ for some $j \in \{1, \dots, |F|\}$. Furthermore point $\boldsymbol{x} \in X$ is called *weakly Pareto efficient* if there exists no solution $\boldsymbol{x}^* \in X$ that $f_i(\boldsymbol{x}^*) < f_i(\boldsymbol{x})$ for all $i \in \{1, \dots, |F|\}$.

Definition 3.2. A point $\boldsymbol{x} \in X$ is said to *dominate* point $\boldsymbol{y} \in X$ if $f_i(\boldsymbol{x}) \leq f_i(\boldsymbol{y})$ for all $i \in \{1, \dots, |F|\}$ and $f_j(\boldsymbol{x}) < f_j(\boldsymbol{y})$ for some $j \in \{1, \dots, |F|\}$.

The set of Pareto efficient points is called *Pareto-set*, *Pareto-front* or *Pareto-frontier*. All of the points in Pareto-set are optimal by definition 3.1. However the points in decision space yielding to these solutions might differ a lot. This is why large multiobjective optimization problems are usually solved with a *decision maker*. Decision maker has preferences about the different Pareto-optimal solutions and is potentially able to direct the solver towards a desired solution. However the assessments of the decision maker might be imperfect, lacking the transitivity or being only ordinal and not cardinal [16]. Or there might not be a decision maker available at all and in these situations it is preferable to produce a set of solutions from different parts of the Pareto-set. The number of solutions even in a single objective optimization problem might be infinite. In multiobjective problems the number of Pareto-efficient solutions can be infinite even in a case where all of the individual objective functions have a unique optimal point. Finding a representative set of solutions from Pareto-set might be computationally exhausting. There are many different methods for exploring the Pareto-set and some of the most classical ones are presented next.

3.2 Methods for finding Pareto-efficient solutions

One way to find a Pareto-efficient point is to optimize each objective function one by one and ensuring that the achieved value of each objective function is not worsened. This is called *lexicographic optimization*. After each minimization iteration the achieved value of the objective function is set as a constraint for next iteration. Generally different Pareto-efficient solutions might be found by optimizing the objective functions in a different order [5] but in some cases the order of the objective functions do not influence the final solution [15]. Using lexicographic optimization do not usually explore the whole Pareto-front. Optimizing one function with expence of other ones does not generally produce similar solutions to a techinque where every function is optimized at the same time.

One way to handle the limited nature of lexicographic optimization is not to stick to the optimal values of individual objective functions but to control their degradations. Such a technique is called *ε-constraint optimization*, where only one objective function is optimized at a time and all the other functions are set as constraints similar to lexicographic optimization. What differs is that ε-constraint optimization allows the values of each non-currently-optimizable objective to be worsened by some $\varepsilon > 0$. This allows further exploration of the Pareto-front. The shape of the Pareto-front does not matter as ε-constrained method can find solutions from either convex or non-convex parts of the Pareto-front.

Both lexicographic and ε-constrained method optimize one objective function at a time while using others as constraints. *Linear scalarization* is a simple technique where all of the objective functions are combined under a single weighted sum-function. This way no extra constraints are needed and a single Pareto-efficient point can be found on each iteration. While linear scalarization is a simple and easy to implement technique it lacks the power to find solutions from non-convex part of Pareto-front [11]. Along with this restriction is the fact that time complexity of exploring the whole Pareto-frontier grows exponentially regarding to the number of objective functions.

Next we will take a closer look at linear scalarizations incompleteness and after that there are two simple examples on how linear scalarization works. Images of the examples are created with Geogebra Classic 5. The .ggb files of these examples can be found from [21].

Definition 3.3. The *cone of descent directions* at $\boldsymbol{x} \in X$ is

$$D_X(\boldsymbol{x}) := \{\boldsymbol{d} \in \mathbb{R}^n |\ \boldsymbol{d} = \boldsymbol{0} \text{ or there exists } \varepsilon > 0 \text{ such that } f(\boldsymbol{x} + t\boldsymbol{d}) < f(\boldsymbol{x}) \text{ for all } t \in (0, \varepsilon]\}. \text{ [1]}$$

Definition 3.4. The *cone of ascent directions* at $\boldsymbol{x} \in X$ is

$$A_X(\boldsymbol{x}) := \{\boldsymbol{d} \in \mathbb{R}^n |\ \boldsymbol{d} = \boldsymbol{0} \text{ or there exists } \varepsilon > 0 \text{ such that } f(\boldsymbol{x} + t\boldsymbol{d}) > f(\boldsymbol{x}) \text{ for all } t \in (0, \varepsilon]\}. \text{ [1]}$$

Definition 3.5. The set S is *convex* if all the points on the line segment between two points $x_a \in S$ and $x_b \in S$ also belong to S. [1]

Definition 3.6. *Convex hull* of set S is the the smallest convex set containing S. [1]

Lemma 3.1. If direction $\boldsymbol{d_1}$ at a point $\boldsymbol{x}$ on a linear function $f : \mathbb{R}^n \to \mathbb{R}$ can be formed as a linear combination of two ascent directions $\boldsymbol{d_2}$ and $\boldsymbol{d_3}$ with positive coefficients, then also $\boldsymbol{d_1}$ must be an ascent direction on f.

Proof. Let $t_i > 0$ for any i. The gradient of the linear function is same everywhere in the domain of the function. This means that an ascent direction $\boldsymbol{d}$ at point $\boldsymbol{x}$ is also an ascent direction at point $\boldsymbol{x} + \boldsymbol{d}$. If we take step to an ascent direction $t_1 \cdot \boldsymbol{d_2}$ at point $\boldsymbol{x}$ and after that another ascending step $t_2 \cdot \boldsymbol{d_3}$ we end up in the point $\boldsymbol{x} + t_1 \cdot \boldsymbol{d_2} + t_2 \cdot \boldsymbol{d_3} = \boldsymbol{x} + \boldsymbol{d_1}$ and as $\boldsymbol{d_2}$ and $\boldsymbol{d_3}$ are ascend direction, $f(\boldsymbol{x} + \boldsymbol{d_1}) > f(\boldsymbol{x})$. So $\boldsymbol{d_1}$ is an ascent direction. □

Lemma 3.2. The cone of descent direction at $\boldsymbol{x}$ for linear function is either an empty set or an open halfspace.

Proof. If the linear function is a constant function, then there exists no descent directions and $D_X(\boldsymbol{x}) = \emptyset$.

Now to the case where linear function is not a constant. This means that $D_X(\boldsymbol{x})$ and $A_X(\boldsymbol{x})$ are nonempty sets. We prove that the cone of descent directions must be at least an open halfspace.

The proof is done by contradiction. Let's suppose that vectors $\boldsymbol{d}$ and $\boldsymbol{-d}$ are both ascend directions. Using the additivity and homogeneity of linear function yields the following calculations

$$\begin{aligned} & f(\boldsymbol{x} + t\boldsymbol{d}) > f(\boldsymbol{x}) \\ \iff & f(\boldsymbol{x}) + f(t\boldsymbol{d}) > f(\boldsymbol{x}) \\ \iff & f(t\boldsymbol{d}) > 0 \end{aligned}$$

and

$$\begin{aligned} & f(\boldsymbol{x} - t\boldsymbol{d}) > f(\boldsymbol{x}) \\ \iff & f(\boldsymbol{x}) - f(t\boldsymbol{d}) > f(\boldsymbol{x}) \\ \iff & -f(t\boldsymbol{d}) > 0 \end{aligned}$$

which can not be true so one of $\boldsymbol{d}$ and $-\boldsymbol{d}$ must be a decending direction. This means that as two opposite directions can not be ascending and from lemma 3.1 we know that any direction "between" two ascending directions is also ascending, the cone of descending directions must be at least an open halfspace. A similar calculation can be done to show that the cone of ascending directions must be at least an open halfspace.

Contour lines on linear function are all straight and moving along this line is not an ascending or descending direction. All the values of the function have equal value on these lines. These lines are parallel to the gradient of the function and they separate cones $D_X(\boldsymbol{x})$ and $A_X(\boldsymbol{x})$.

So the cone of descending direction on linear function is either an empty set or an open halfspace. □

Theorem 3.3. Solutions from a non-convex region of Pareto-frontier can not be detected by means of linear scalarization.

Proof. Let $\boldsymbol{x'}$ be a Pareto-efficient point from a non-convex region of Pareto-front. Let $\boldsymbol{x_a}$ and $\boldsymbol{x_b}$ be some other efficient point from this same non-convex region such that $f_i(\boldsymbol{x_a}) < f_i(\boldsymbol{x'}) < f_i(\boldsymbol{x_b})$ for some i. By the definition of convexity we know that with some $c \in (0,1)$ there exists a point $\boldsymbol{x_c} = c \cdot \boldsymbol{x_a} + (1-c) \cdot \boldsymbol{x_b}$ that does not belong to the Pareto-front. As this point is not part of the Pareto-front but part of its convex hull, it is possible to choose the value of c that this point $\boldsymbol{x_c}$ dominates $\boldsymbol{x'}$. It should be thus noted that this artificial point $\boldsymbol{x_c}$ is not feasible.

The weighted sum function formed by linear scalarization is a linear function from $\mathbb{R}^n$ to $\mathbb{R}$. So by lemma 3.2 we know that the cone of descending directions is an open halfspace. This means that it is not possible that both $\boldsymbol{x_a}$ and $\boldsymbol{x_b}$ map to a higher value than $\boldsymbol{x_c}$.

As $\boldsymbol{x_c}$ dominates $\boldsymbol{x'}$ and at either $\boldsymbol{x_a}$ or $\boldsymbol{x_b}$ produce at least as good values for linear scalarization sum function as $\boldsymbol{x_c}$, $\boldsymbol{x'}$ can not be found by linear scalarization method. □

Example 3.4. Suppose we have an optimization problem as follows:

$$\begin{aligned} &\min\ f_1(x) = x \\ &\min\ f_2(x) = \frac{1}{x} \\ &\text{s.t. } x > 0. \end{aligned}$$

It can be seen that any value of $x > 0$ is a Pareto-optimal solution. Increasing x will make value f_2 smaller but also increase value of f_1. Similarly chosing smaller x makes value of f_1 smaller but increases value of f_2. Combining these two objective functions under a single sum function yealds the following optimization problem

$$\begin{aligned} \min\ & g(x) = \lambda f_1(x) + (1-\lambda) f_2(x) \\ \text{s.t. } & x > 0. \end{aligned}$$

Pareto-front search by linear scalarization is visualized in figures 1, 2, 3 and 4. Green curve is function $x \mapsto \frac{1}{x}$, black line is the contour line of weighted sum in objective space with different λ values. The arrow points to the better values. Red curve is the weighted sum function $g(x) = \lambda f_1(x) + (1-\lambda) f_2(x)$.

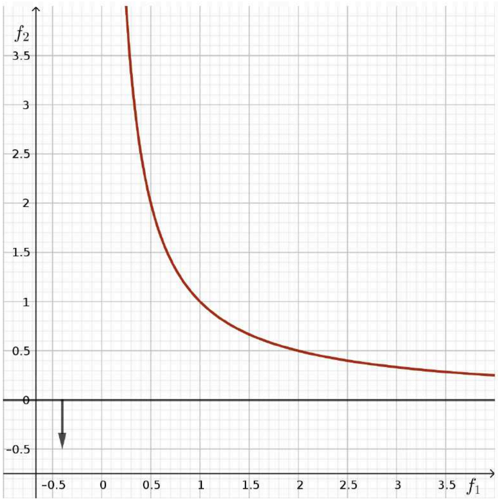

Figure 1: $\lambda = 0$

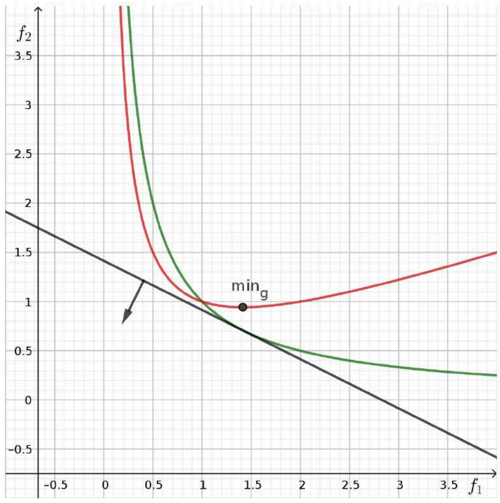

Figure 2: $\lambda = \frac{1}{3}$

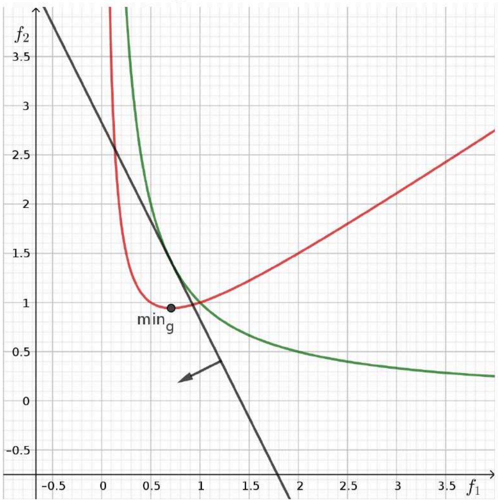

Figure 3: $\lambda = \frac{2}{3}$

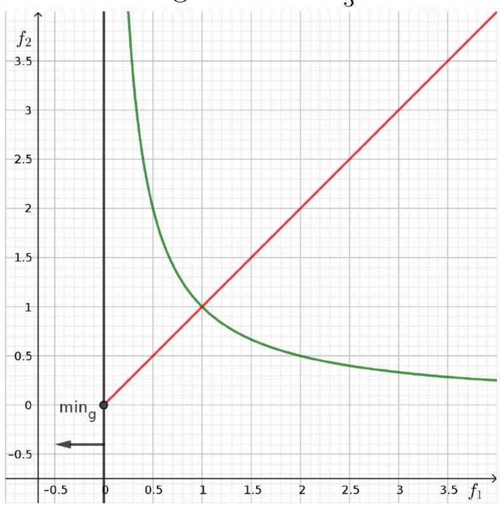

Figure 4: $\lambda = 1$

In previous example linear scalarization worked fine as our Pareto-frontier was convex. Next we take a look at a situation where linear scalarization does not work in a desirable way. Linear scalarization is able to produce some Pareto-efficient solutions but only extreme points of a non-convex region.

Example 3.5. Suppose an optimization problem as follows

$$\begin{aligned} &\min\ f_1(x) = x \\ &\min\ f_2(x) = -x^2 \\ &\text{s.t. } x \in [0, 1]\,. \end{aligned}$$

In this case the Pareto-frontier is not convex and so linear scalarization gets into trouble when searching for solutions from non-convex part of Pareto-front. Let $g(x) = \lambda f_1(x) + (1-\lambda) f_2(x)$. The following table shows the optimal solution for g with 6 different λ-values. Figures 5, 6, 7 and 8 illustrate the Pareto-front and sum function similar to example 3.4.

λ	$g(x)$	$arg\ min\ g(x)$
0	$-x^2$	1
0.2	$0.2x - 0.8x^2$	1
0.4	$0.4x - 0.6x^2$	1
0.6	$0.6x - 0.4x^2$	1
0.8	$0.8x - 0.2x^2$	0
1	x	0

Table 3

From the above table it can be seen that as the Pareto-front is not convex, linear scalarization is only able to find the individual minimas of the objective functions. For these kind of non-convex problems some other technique such as ε-constrained method might be a better choice [11].

Even though its simplistic nature, the possibility to use linear scalarizations without any extra constraint is the key benefit for purposes of data

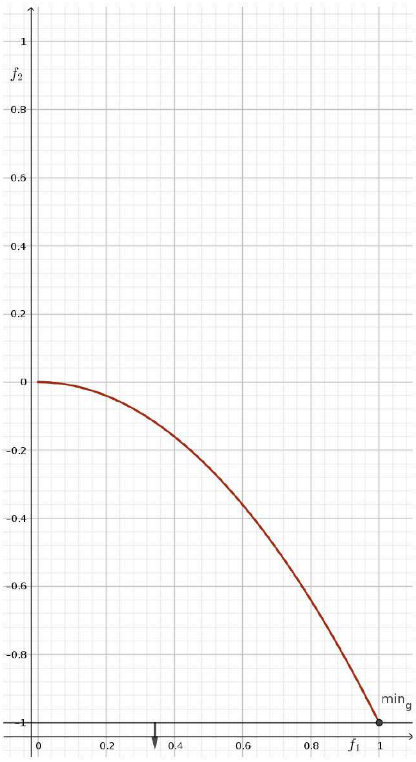

Figure 5: $\lambda = 0$

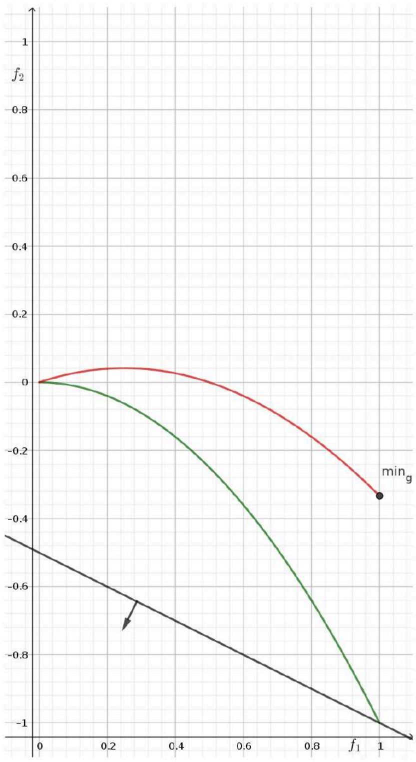

Figure 6: $\lambda = \frac{1}{3}$

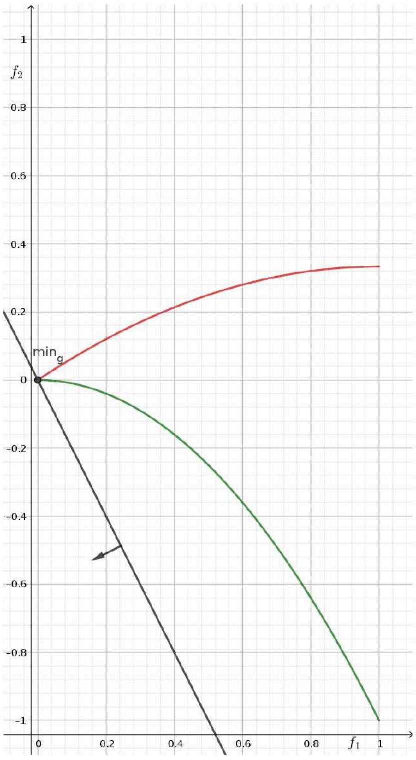

Figure 7: $\lambda = \frac{2}{3}$

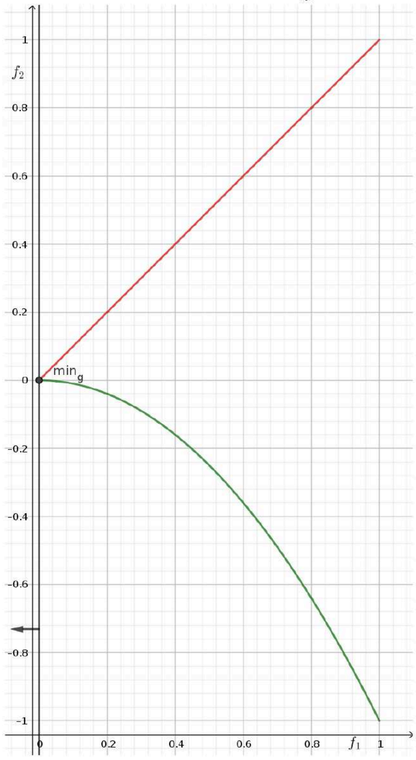

Figure 8: $\lambda = 1$

analysis. This makes it possible to use the idea of linear scalarization even in solutions where no constrains can be used in the first place. The focus of this thesis is on the PageRank algorithm which is not an optimization process but a data-analysis algorithm. Using linear scalarization with PageRank makes it possible to produce a *Pareto-PageRank* with only slight modifications to the original algorithm. The purpose of PageRank is not only to find the most important vertex but also second, third and nth most important vertices. PageRank also reveals the least important vertices and in some situation this might also be important for the decision maker. As the main idea of PageRank is the ranking or ordering of the vertices with iterative algorithm, adding constraints to matrix multiplication is not possible. Extending PageRank to multiobjective dimension is possible by using linear scalarization.

Linear scalarizations lack of ability to find solutions from a non-convex part of the Pareto-frontier is not a limiting factor in PageRank either. As the nature of PageRank's iterative process, the result calculated in (4) is a single real number, not a function to be minimized.

Let C be the set of cost attributes for arcs and $c_p(a_{ij})$ means the weight of attribute $p \in C$ of the arc a_{ij}. Pareto-PageRank solutions can be calculated as in (4) but $c(a_{ij})$ is redefined as

$$c(a_{ij}) = \sum_{p \in C} \lambda_p c_p(a_{ij}), \qquad (7)$$
$$\text{where } \sum_{p \in C} \lambda_p = 1.$$

With the above redefined cost, Pareto-PageRank points can be calculated with an artificial number of cost attributes. However as mentioned before the exploring of Pareto-frontier gets computationally exhausting as $|C|$ grows. In this thesis, only two dimensional ranks were calculated.

Another way to scalarize objective functions is to use *Chebyshev scalarization method*. Instead of sum, this scalarization technique uses weighted *max-norm* as the combined objective function. This way the resulting objective function is

$$g(\boldsymbol{x}) = \max(\lambda_1(f_1(\boldsymbol{x}) - \boldsymbol{I}_1), \lambda_2(f_2(\boldsymbol{x}) - \boldsymbol{I}_2), \ldots, \lambda_n(f_n(\boldsymbol{x}) - \boldsymbol{I}_n)), \qquad (8)$$

where I is an *ideal point* in objective space. This ideal point consists of individual minimas of each objective function.

One big advantage Chebyshev method has over linear scalarization is its ability to find solutions also from a non-convex part of the Pareto-frontier [13]. However, some additional calculations must be done to separate non-weak Pareto-efficient solutions from the weakly efficient solutions [13].

4 Road networks

In this chapter the structure of road networks is discussed as well as the need for data analysis algorithms for such graphs. Characteristics of a typical road network are covered. More detailed view of the data used in this thesis is provided in chapter 6.

4.1 Graph model

As discussed in chapter 2.2, a graph $G = \{V, A\}$ is a set of vertices and arcs connecting those vertices to each other. Topological networks such as road are intuitively modelled as graphs. Even graph theory itself can be considered to be born in 1736, when Euler published his paper about the Seven Bridges of Königsberg [2].

One way to produce a graph from a road network is to assign all the intersections as vertices and roads as arcs. This way the roads, or arcs, can have length and other natural attributes. This type of graph presentation is perhaps the most common for road networks. In this thesis this kind of representation is called a *primal graph*. Other possibility is to produce a *dual graph* from this primal graph, where vertices represent street segments and arcs connect those segments together, just like intersections connect roads [14]. In this thesis, primal graph presentation is used but it should be noted that the PageRank algorithm is not limited only to this kind of graphs.

Even though topological networks were the ones that gave birth to the graph theory, the structure of these networks is quite different to many more modern and abstract networks. The first use case of the PageRank, internet, has created possibilities for these kind of networks to exist. Social media and article citation networks are examples of those. While the average degree of citation network is around 10 (see e.g. [7] and [10]), the average degree of social network is measured in hundreds [7]. For road networks the average degree in urban environment is around 3 [3]. One limiting factor of street networks is that they are almost *planar*. This makes vertices with degree higher that 6 quite unusual [19]. Even with the structural differences between road networks and usual use cases of PageRank, it is shown that PageRank can be efficiently used also for road network graphs [4].

4.2 Reliability of the graphs

Road network risk analysis consists of analysing the structure of road network to degrations. These degrations may consist of vertex or arc complete failures or some degration of their attributes such as capasity or speed. The amount of different attributes available limits the usage of some methods. *Connectivity reliability* analysis can be done without any additional attributes. In connectivity reliability analysis the interest is in whether two links stay connected or not after the degration of the network. Other analysis techniques can take into account the travel times and capasity of the network. While *travel time reliability* focuses on the time difference when travelling between vertices in intact and degraded network, *capasity reliability* analyses how much travelling capasity the network can handle. This analysis can be done in either intact or degredad network. [20]

The limiting factor in road network risk analysis is the fact that the amount of attributes in public sources is limited [9]. The main datasource in this thesis is the open source database OpenStreetMap [25]. In [9] it was shown that the only two attributes available for Denmark road network were road category and speedlimit. From these two the latter attribute was available for only 13 % of the arcs.

Similar difficulties were faced in this thesis. As the OpenStreetMap data was missing many attributes and the only consistent one was the road type attribute, this road type was used to create two different new attributes that are used for road network importance analysis. These attributes were *bicycle safety* and *bicycle speed* -attributes. The reason these two were chosen was the fact that they can be produced from the only consistent attribute in the data. Also speed and safety are somewhat contrary to each other so comparing these attributes is interesting. A safe road usually is not fast. Altough it as it can be seen from the table 4 this is not true in every case. Safety and speed attributes were given a weight depending on the road type. Table 4 shows the weights of these two attributes for different OpenStreetMap road types.

Roadtype	Bicycle speed	Bicycle safety
motorway / motorway_link	1.0	0.1
trunk / trunk_link	1.0	0.1
primary / primary_link	1.0	0.1
secondary / secondary_link	1.0	0.2
tertiary / tertiary_link	1.0	0.4
unclassified	1.0	0.4
residential	1.0	0.6
living_street	0.8	0.8
service	0.5	0.8
cycleway	1.0	1.0
bus stop	0.2	0.8
services	0.2	0.8
platform	0.2	1.0
pedestrian	0.2	1.0
track	0.2	1.0
footway	0.2	1.0
bridleway	0.2	1.0
steps	0.2	1.0
corridor	0.2	1.0
path	0.2	1.0
crossing	0.2	1.0
rest_area	0.1	1.0

Table 4: Road type factors

One important note about the road types is that the type "unclassified" is actually a defined roadtype in OpenStreetMap and type "road" means that the road type attribute is unknown or missing [26].

The idea behind these artificial factors was that the speed and safety of a cyclist are depending on the road type. The speed of the cyclist is not limited by traditional speed limit but is dependant of the road type. Tarmac roads with good conditions yield higher speeds while smaller dirt roads reduce the speed. The safety of the cyclist is thought to depend on the other trafic on the route. Footways limit the speed of a cyclist heavily but they are relatively safe. While primary roads do not limit the speed, there are more cars and trucks travelling on that road which creates a safety hazard for the cyclist. The legislation aspects were not taken into account while creating these factors. For example it is forbidden to cycle on motorway in Finland but the motorways and trunks were still given a speedfactor of 1.0. All the factors were created by the author of this thesis and the main reason for this decision was the limitation of open source data resources.

5 Pareto-PageRank algorithm

In previous chapters the theory behind the PageRank and multiobjective optimization was covered as well as the structure of data and problem at hand. This chapter takes these elements and combines them into a pseudocode algorithm. Beside the algorithm itself, some data preparation and visualization methods are covered.

5.1 OpenStreetMap data preparation

The data of road network in OpenStreetMap database can be downloaded as an osm-file. This is a type of xml-fileformat used by OpenStreetMap. These files contain information about a given area on the globe and can be downloaded from *https://www.openstreetmap.org*. These files contain information about road network as well as buildings, railways, waterways, points of interest, parks, etc. An osm-file consists mainly on two different kind of xml-elements: *nodes* and *ways*. Nodes are fixed points that have coordinates and sometimes some attributes, if the node is a point of interest such as a bank or a shop for example. Ways combine multible nodes into a larger entity with more information. These ways can be an outline of a building, a road, a powerline, boundaries of cities, etc.

As this thesis focuses on the road network, all the unnecessary information is removed and the structure of road network is cleaned and saved as a csv-file for the main analysis. This cleaning of road network is done by removing all of the *go-through* vertices and combining the arcs connecting into this vertex into a single new arc. These go-through vertices are defined as vertices with degree of two and that the connecting arcs are of the same road type.

After this cleaning only the necessary information for the analysis is left and saved into two csv-files. One containing information about the vertices and one about edges. From vertices only id, longitude and latitude attributes are saved. From arcs saved attributes are: id, start vertex id, end vertex id, boolean value whether the arc is two directional or not, and road type of the way. Files used in this thesis can be found from [21].

5.2 Data normalization

Data normalization is an important element in many data analysis methods. The meaning of this process is to scale the values of different attributes into a common interval. In the case of linear scalarization this means that the changes on λ-value have effect on the whole $[0, 1]$ interval and not only near one end. As all the weights of arcs in PageRank calculation (4) need to be positive, min-max normalization was used in this thesis. With min-max normalization the values of different attributes can be brought into any $[a, b]$ interval with the following method:

$$x_{new} = a + (b - a) \cdot \frac{x_{orig} - \min(X)}{\max(X) - \min(X)}, \tag{9}$$

where x_{new} and x_{orig} are the new and original value for attribute x and $min(X)$ and $max(X)$ are the minimum and maximum values of the original attribute set X.

As the c function of (4) was defined to have values from interval $[0, 1]$, the a and b of (9) should be chosen accordingly. Choosing $a = 0$ means that some arcs will not distribute their rank when considering purely that attribute. This is why it is advisible to choose $a, b \in (0, 1]$. In this thesis values $a = 0.1$ and $b = 1$ were chosen.

5.3 Pseudocode algorithm

Once the data has been prepared from osm-file as in chapter 5.1 the data normalization and Pareto-PageRank algorithm can be excecuted. The pseudocode for the algorithm is presented next. Python implementation can be found from [21].

Algorithm 1 Pareto-PageRank algorithm for biobjective attributes

```
G ← (V, A)
P ← set of attributes
Λ ← set of preset values for λ
for all a ∈ A do
    for all p ∈ P do
        normalize c_p(a_ij) as in (9)
    end for
end for
for all λ ∈ Λ do
    λ_1 ← λ
    λ_2 ← 1 − λ
    t ← 0
    for all v ∈ V do
        initialize PR_0(v)
    end for
    repeat
        for all v ∈ V do
            calculate PR_{t+1}(v) as defined in (4) and (7).
        end for
        t ← t + 1
    until converges
    save the current PageRanks
end for
```

On each iteration of the PageRank calculation, the rank of each vertex is calculated. For this to be done, all the rank of each adjacent vertex must be taken into account. The number of dimensions and thus the number of different λ's also affects the overal time complexity. Let L be the set of usable attributes. This leads to time complexity of $O((|V|^2)^{|L|-1})$ for Pareto-PageRank. This means that the performance is heavily dependant on the used dimension. This is one reason why only two-dimensional Pareto-PageRank is used in this thesis.

Theorem 5.1. Algorithm 1 will converge and produce a set of Pareto-efficient PageRanks.

Proof. The calculation of PageRank is based on the transformation matrix M defined in (3). Because M homogenous and ergodic, by theorem 2.1 we know that the PageRank calculation converges.

Also theorem 3.3 states that Pareto efficient solutions can be found by the means of linear scalarization. □

After the algorithm has been executed a set of Pareto efficient solutions is produced. These solutions can then be analysed for example on spreadsheet calculator or in the case of road network, visualizing the solutions to a map. Even though PageRank is an attribute of the vertices, it is possible to produce PageRank of arcs by defining the PageRank of arc a_{ij} to be as

$$PR(a_{ij}) = \frac{PR(v_i) + PR(v_j)}{2}. \tag{10}$$

Using the rank of arcs rather than vertices makes it easier to visualize the solution to a map.

For visualization purposes on map, webpage *https://geojson.io* was used. Solutions were saved in a geojson-fileformat and arcs were colored relevant to their PageRank value. For this purpose the final ranks were min-max normalized and colors were chosen from predefined *perceptually uniform palettes* from Seaborn library [28]. Figures 9 and 10 are examples of perceptually uniform palettes.

Figure 9: Magma palette

Figure 10: Viridis palette

6 Test cases and results

This chapter focuses on the results of the algorithm on test cases. The structure of the test datasets and the results are covered in this chapter. All of the test are executed on Python 3.8.10 running on Linux Mint 20.3 x86_64. As a hardware, ThinkPad T440S laptop with Intel i7-4600U processor was used. The xml-format data was downloaded from *www.openstreetmap.org*. Before the execution of the Pareto PageRank algorithm, the data was preprosessed as described in chapter 5.1.

The results are covered and analyzed as well. The focus is on the linear scalarization but also some comparision between linear scalarization and Chebyshev scalarization is done.

6.1 Test Data

The test data used in this thesis consist of OpenStreetMap data around 10 largest cities in Finland: Helsinki, Espoo, Tampere, Vantaa, Oulu, Turku, Jyväskylä, Kuopio, Lahti and Pori [29]. The data areas used in the thesis are marked on the figure 11. The surface area of each dataset is different but the number of arcs and vertices is not only dependant on the surface area but also the density of the road network. Densily populated cities near Helsinki metropolitan area have more vertices per square kilometer compared to less populated areas for example around Oulu and Jyväskylä.

The structure of each dataset is tabulated in appendix A. The average degree of each dataset was quite consistant, rangeing from Espoo's and Jyväskylä's 2.62 to Lahti's 2.82. From the tables it can be seen that large part of the arcs on graph are residential road, service roads, cycleways, footways and paths. These types of roads usually consist of shorter segments than longer primary roads and motorways. The length of arcs is not used in the analysis.

Figure 11: Test areas

6.2 Results

Pareto-PageRank was calculated with λ values $0, \frac{1}{10}, \ldots, \frac{9}{10}, 1$. The calculations were done using linear scalarization and Chebyshev scalarization. Both results can be found from appendix B. The tables show the number of different roadtypes among the top 100 most important arcs on each city calculated with different λ values.

Alongside the table, the results for Pori are visualized on a map. Figures 12, 13 and 14 show the 10 most important vertices while figures 15, 16 and 17 visualize the ranks of all of the arcs with Magma palette with λ-values 0, $\frac{1}{2}$ and 1.

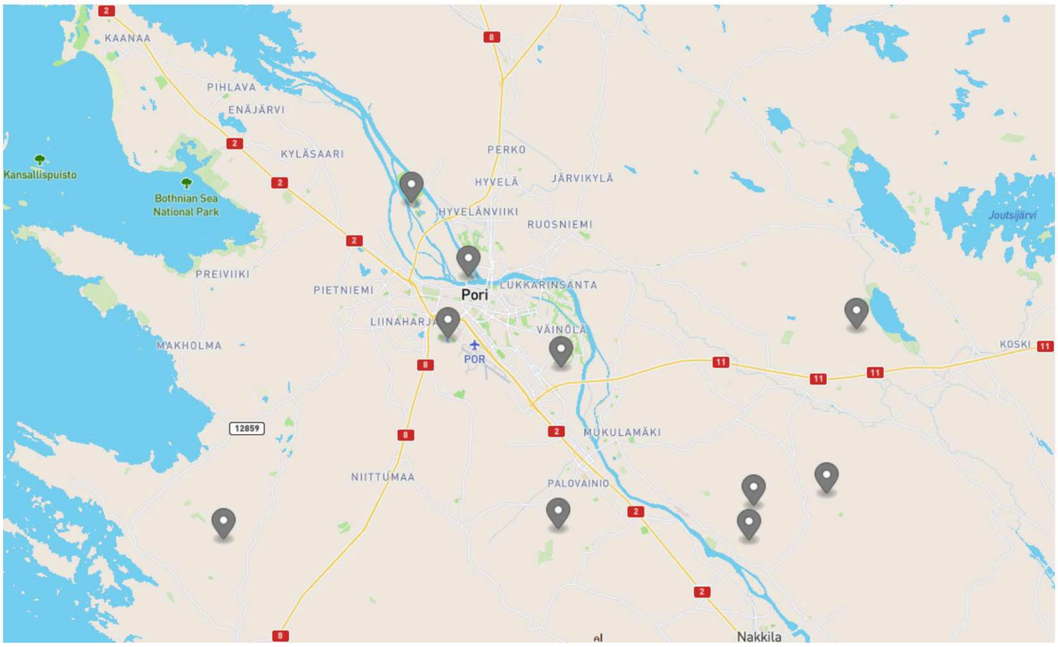

Figure 12: Linear scalarization: $\lambda = 0$

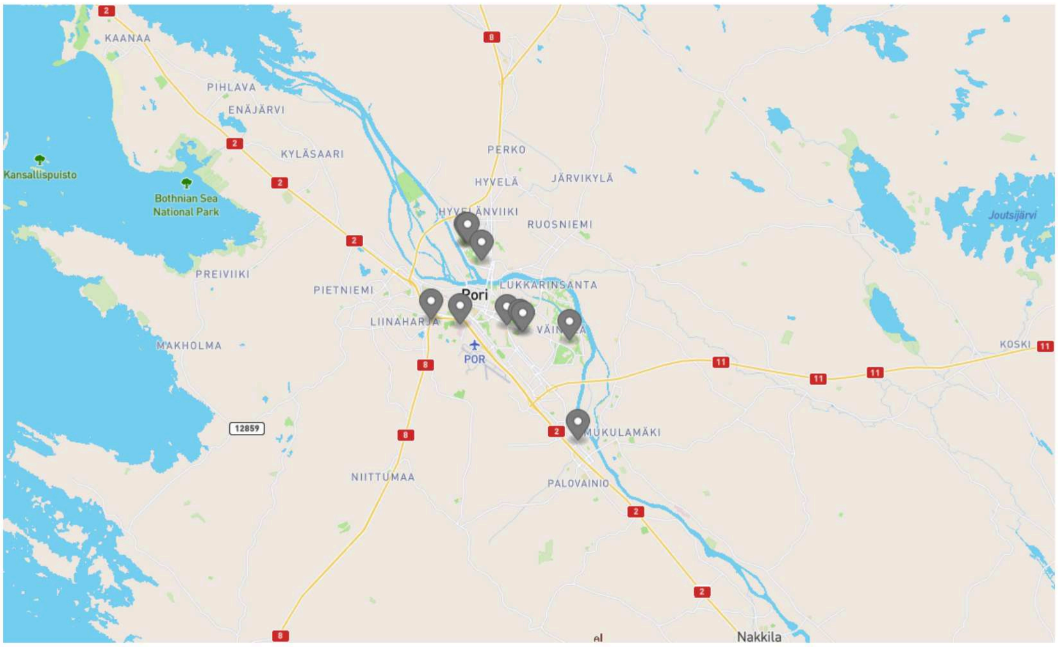

Figure 13: Linear scalarization: $\lambda = 0.5$

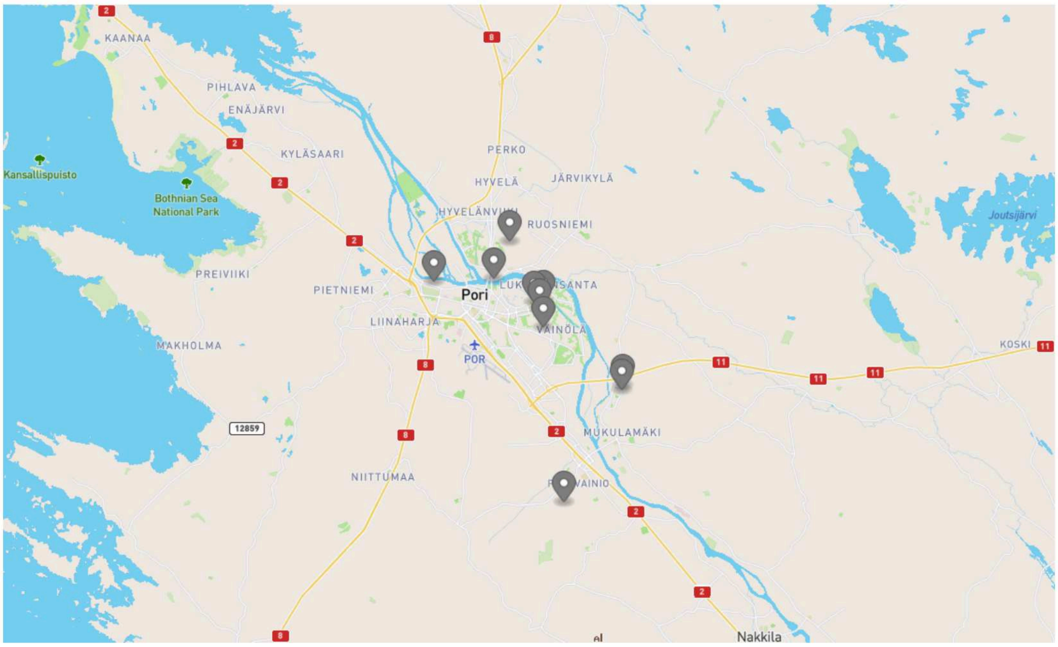

Figure 14: Linear scalarization: $\lambda = 1$

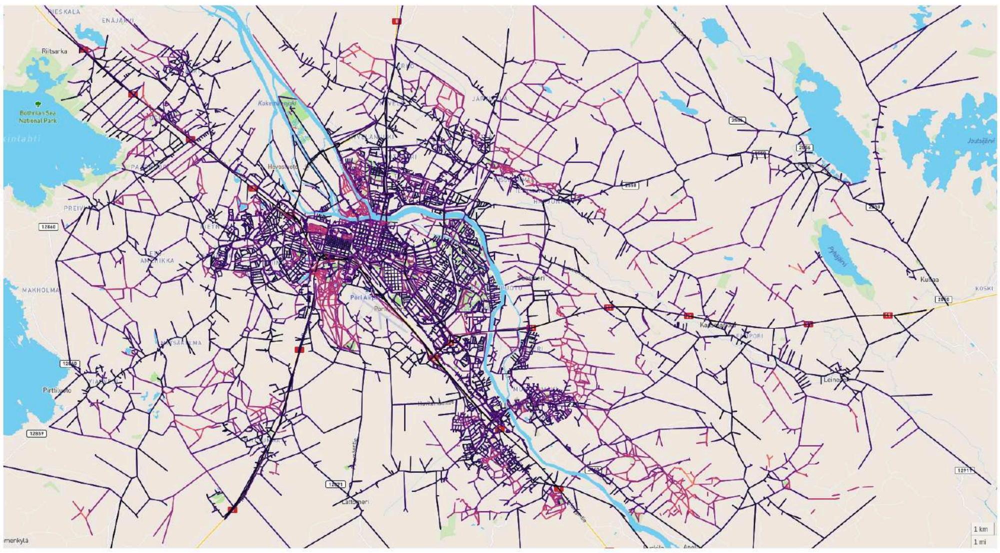

Figure 15: Linear scalarization: $\lambda = 0$

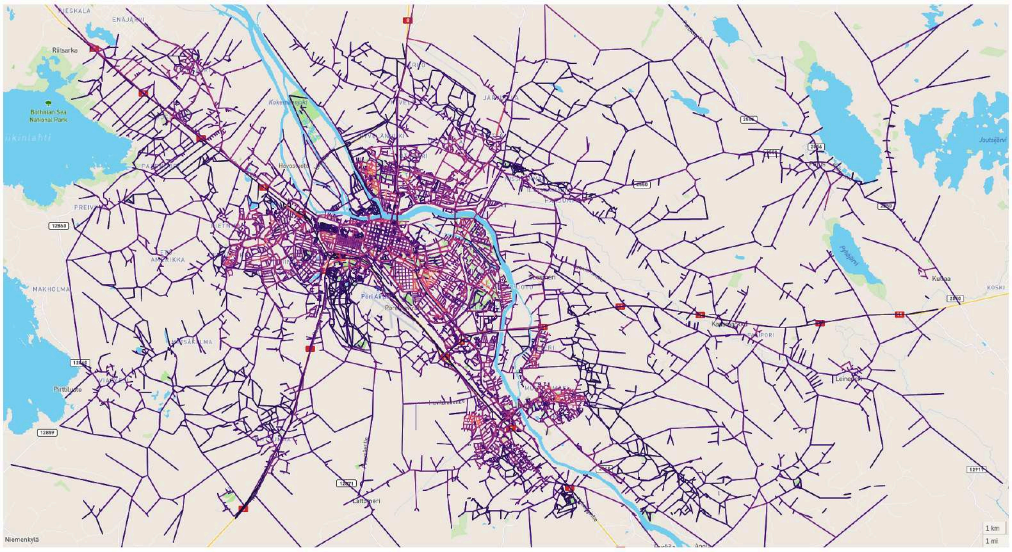

Figure 16: Linear scalarization: $\lambda = 0.5$

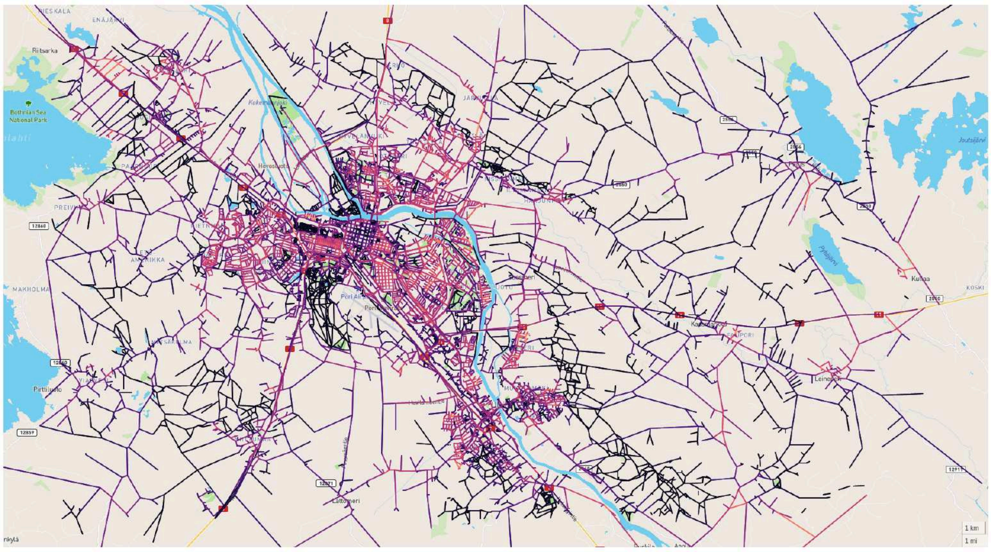

Figure 17: Linear scalarization: $\lambda = 1$

6.3 Analysis

Results from all datasets show that with low λ-values, there are many tracks, paths and footways among the most important arcs. These types all have the highest safety factor in the data: 1.0. With moderate λ-values, cycleways dominate the set of most important arcs. This correlates with the fact that these types of roads have the highest factor in both safety and speed. When focusing more on speed, some cycleways are still important but also other types of roads are present among the most important arcs, most notably residential roads. When the focus is purely on speed, some roads with very poor safety factor are also included such as secondary, tertiary and unclassified roads, but also primary roads, motorways and trunks.

Inspecting figures shows that the downtown of Pori seems to have parts that have high importance with all λ-values altough the importance of these areas seem to rise when focusing more on speed. When $\lambda = 0$ it seems that the most important vertices are spread out more widely compared to higher λ-values. With low λ-values there are some important vertices near the downtown, mainly in the Isomäki outdoor activities area, but the top 10 vertices are still spread out quite evenly. With higher λ-values the roads of the downtown seem to gain importance and dominate the set of most important vertices.

From the results it can be noticed that the Pareto-PageRank algorithm with bicycle speed and bicycle safety attributes produces a set of different solutions but with some consistency among them. The most important vertices and arcs seem to heavily dependent on the used λ-value but in the large part of the Pareto-set, most important arcs are cycleways, arcs that have good factor on both attributes. Visually the most important arcs seem to locate into higher degree areas, such as in the traditional non-weighted PageRank algorithm. Every solution with different λ-value is equally important but these kind of similarities in most of the solutions can give valuable information for the decision maker.

6.4 Chebyshev scalarization

The main scalarization method used in this thesis is linear scalarization. However also Chebyshev scalarization was used in comparison. The ranks calculated with Chebyshev scalarization are guaranteed to be only weakly Pareto efficient. This subsection is used for the difference analysis of the two used scalarization methods. The types of top 100 arcs are tabulated and can be found from appendix B. Similarly to linear scalarization, the results for Chebyshev scalarization are visualised in figures 18 - 23.

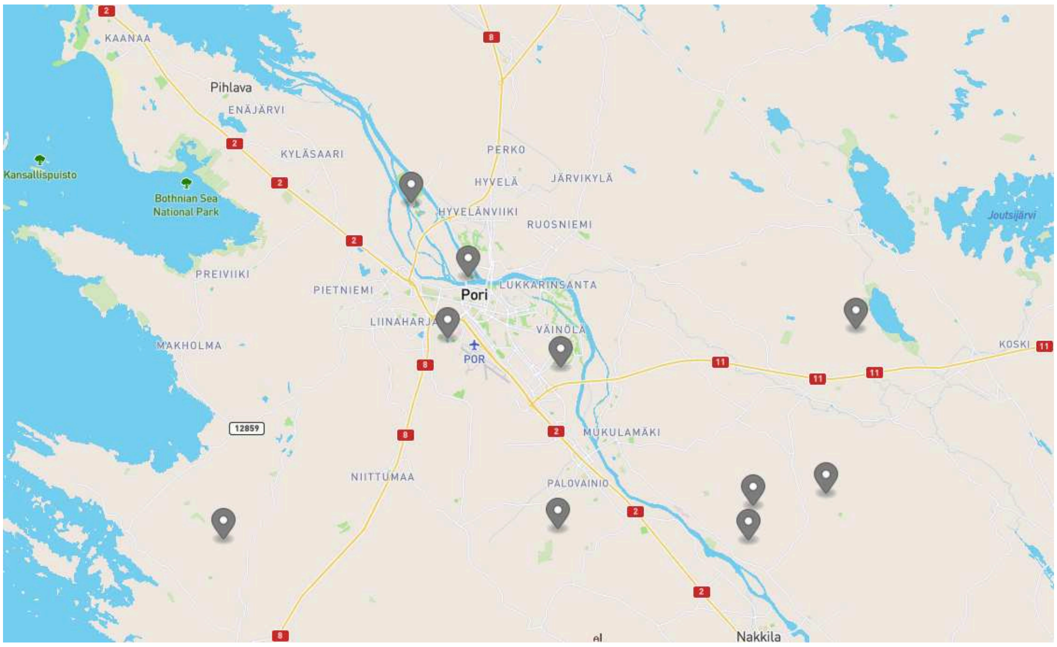

Figure 18: Chebyshev scalarization: $\lambda = 0$

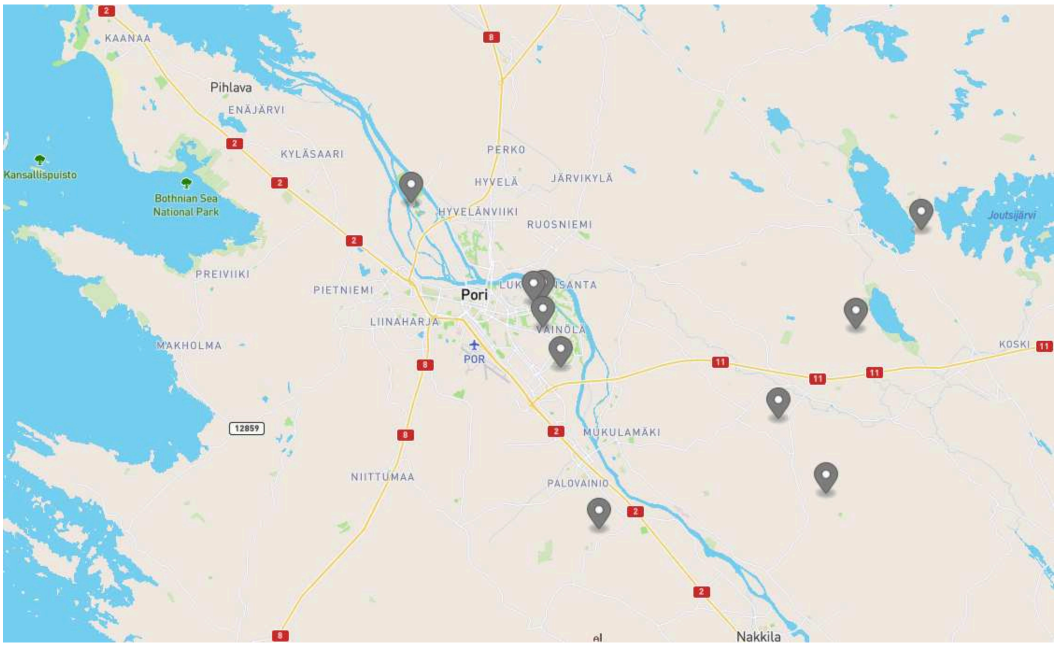

Figure 19: Chebyshev scalarization: $\lambda = 0.5$

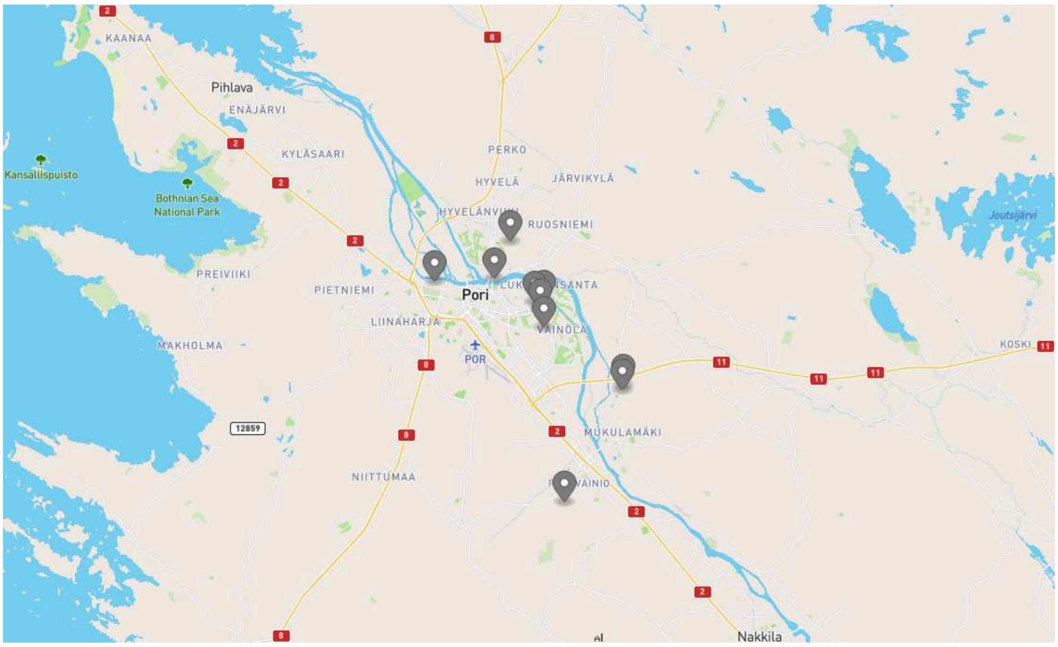

Figure 20: Chebyshev scalarization: $\lambda = 1$

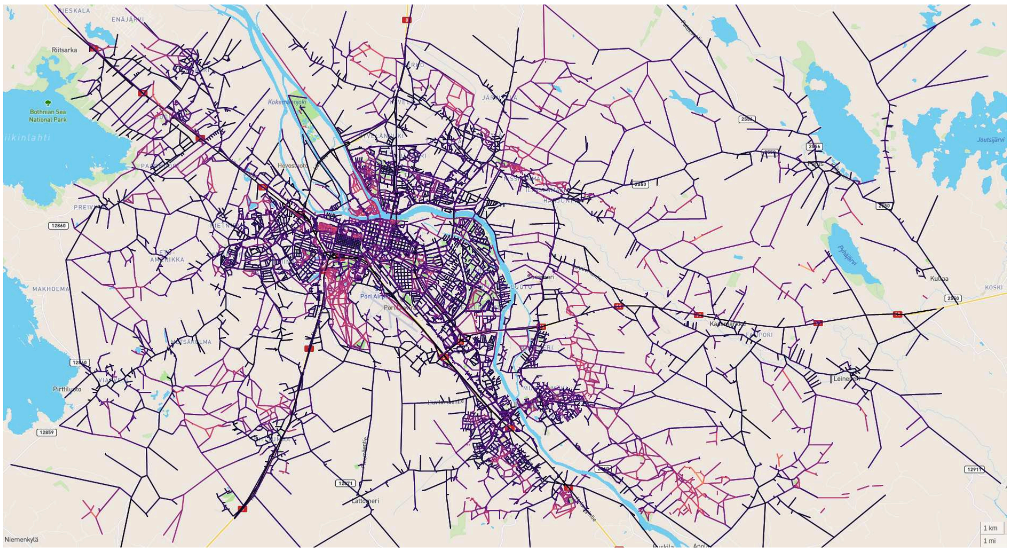

Figure 21: Chebyshev scalarization: $\lambda = 0$

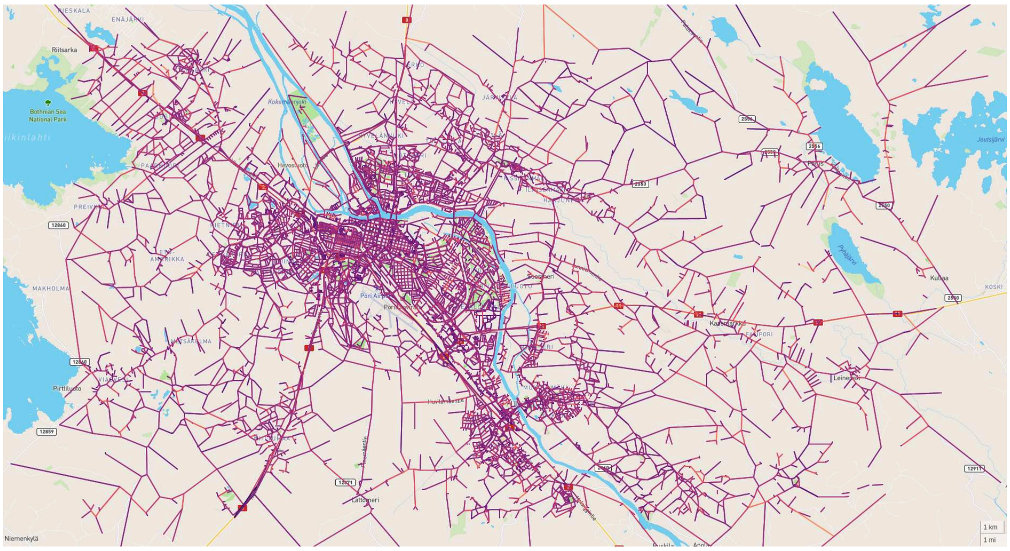

Figure 22: Chebyshev scalarization: $\lambda = 0.5$

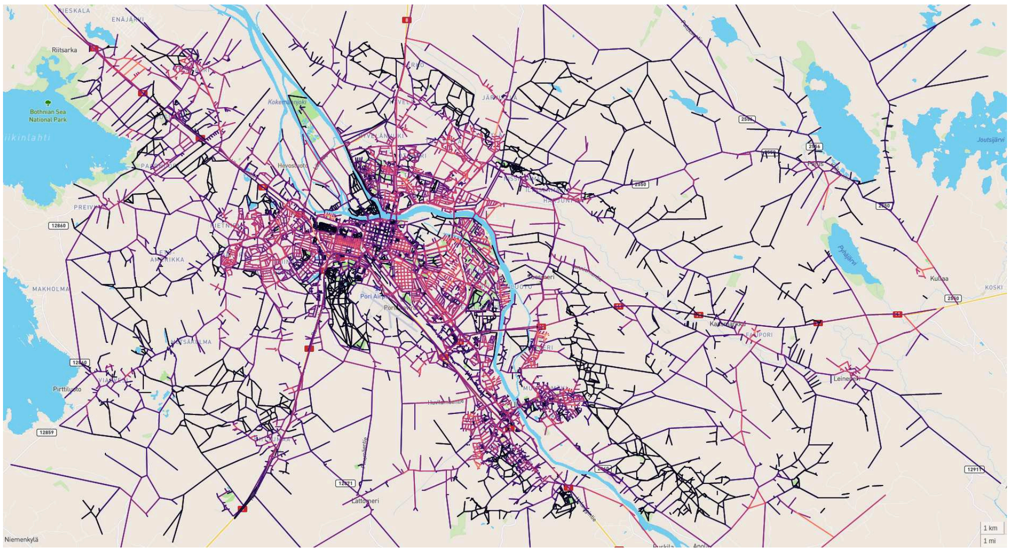

Figure 23: Chebyshev scalarization: $\lambda = 1$

The differences between linear scalarization and Chebyshev scalarization were inspected with statistical tests. For these test the structure of top 100 most important arcs was used to analyse whether the type of scalarization used has statistically significange impact on the results. To measure the difference of two distribution one can use either Student's t-test or Wilcoxon signed-rank test. Student's t-test needs normally distributed data to be used. As the used data consists of only 1000 arcs it is not possible to assume that the data was normally distributed. This is why Wilcoxon test was selected as the statistical test used. The null hypothesis in Wilcoxon test assumes that the median difference between pairs of datapoints is zero. [12]. The tests were performed with R statistical software.

The test were carried out by forming tables 37 and 38 that summed up the occurrence of each roadtype with each λ value. Wilcoxon test was executed for each λ to check if the difference of distribution of different road types is statitically significant or not. The significance level α is 0.05. Table 5 shows the results of these tests.

Variance and mean were calculated with R's build in functions. The sample variance s^2 is calculated as

$$s^2 = \frac{\sum_{i=1}^{n}(x_i - \overline{x})^2}{n-1}, \text{ where } \overline{x} \text{ is the mean of the datapoints.} \tag{11}$$

The mean $\overline{x}$ is calculated as

$$\overline{x} = \frac{\sum_{i=1}^{n} x_i}{n}. \tag{12}$$

λ	Sample Mean linear	Sample Variance linear	Sample Mean Chebyshev	Sample Variance Chebyshev	Wilcoxon-test p-value
0.0	50	14430.74	50	14430.74	NaN
0.1	50	19796.95	50	14823.16	0.1906
0.2	50	45204.11	50	14928.11	0.109
0.3	50	48023.89	50	14710.21	0.1258
0.4	50	48335.16	50	14306.21	0.1671
0.5	50	48337.26	50	7562.421	0.01142
0.6	50	46529.68	50	16552.42	0.1232
0.7	50	40128.42	50	15462.21	0.1232
0.8	50	26472.84	50	14425.68	0.1235
0.9	50	19822.84	50	14245.16	0.1232
1.0	50	15051.26	50	15051.26	NaN

Table 5: Statistical analysis

With $\lambda = 0$ and $\lambda = 1$ linear scalarization and Chebyshev scalarization produce identical arc weights and thus the set of top 100 arcs are identical. From other value only with $\lambda = 0.5$ the difference is statistically significant. With every other λ value the p-value is less than 0.2 this is still higher than the used significance level.

The visualization and Wilcoxon tests suggest that the scalarization method does not have a significant impact on the results. However there were less cycleways among the most important arcs when using Chebyshev scalarization. One reason for this might be the fact that in Chebyshev scalarization the lesser of the two attributes does not have as big impact on the total cost as in the linear scalarization. The arc factors with $\lambda \in [0, 1]$ are visualized in figures 24 and 25. The red curve is the factor of cycleways and all the other roadtypes are marked with gray. From these figures it can be seen that when using linear scalarization, cycleways have the highest arc factor with all the $\lambda \in (0, 1)$. However with Chebyshev scalarization multible road types share the same arc factor yielding to situation where no road type has superior arc factor with any λ value.

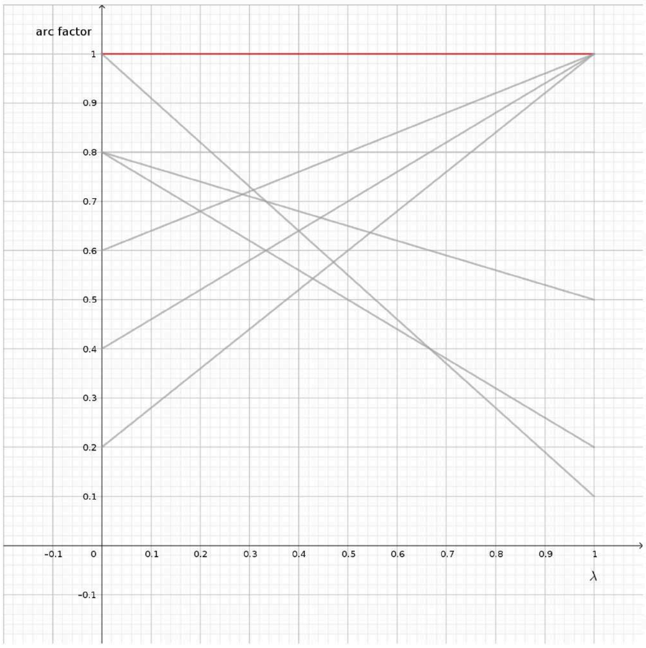

Figure 24: Linear scalarization

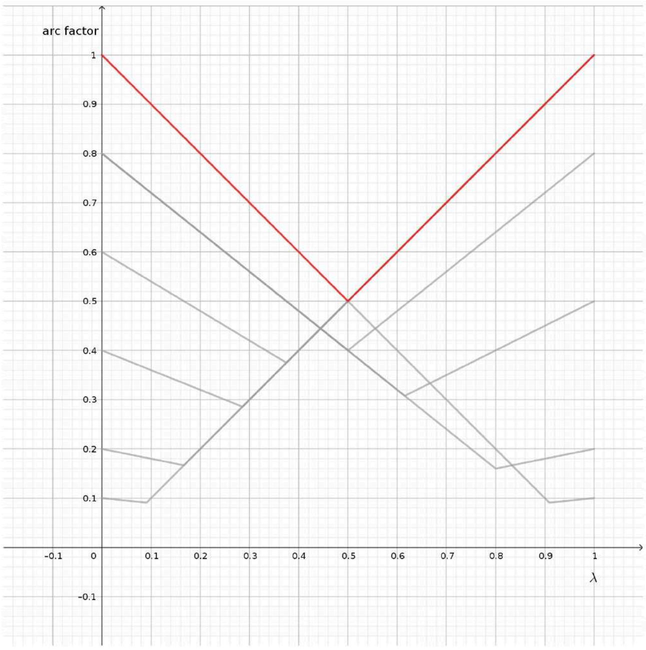

Figure 25: Chebyshev scalarization

6.5 Performance

The performance of the implementation was measured during calculations on each dataset when using linear scalarization. The number of iterations and execution time for each λ-value was saved. Table 6 shows the average number of iterations and average time over all λ values for each dataset.

Data	Average iterations	Average time (s)
Helsinki	147	81.5
Espoo	132	51.2
Tampere	149	62.9
Vantaa	136	100.3
Oulu	146	60.2
Turku	149	60.6
Jyväskylä	133	36.6
Kuopio	131	20.7
Lahti	131	21.3
Pori	144	14.2

Table 6: Performance with dataset

As a termination condition the total amount of PageRank difference between iterations was used. This was done by comparing the new rank to the one calculated on previous round. The difference was summed over all vertices and saved as a single floating number. If this total sum is smaller than ε, then the algorithm will terminate. In this thesis value $\varepsilon = 10^{-10}$ was used.

Analysing the perfomance of the implementation further reveals that the performance seems to depend on the used λ value. Figures 26 and 27 visualize the performance of the algorithm with different λ values. The thick line shows the median of the data. Borders of the box show first and third quartile and horizontal lines mark the minimum and maximum values of the data exluding outliers. Outliers are visualised with a circle. It seems that with λ close to one of the endpoints of the value range, the number of iterations and thus the execution time gets higher.

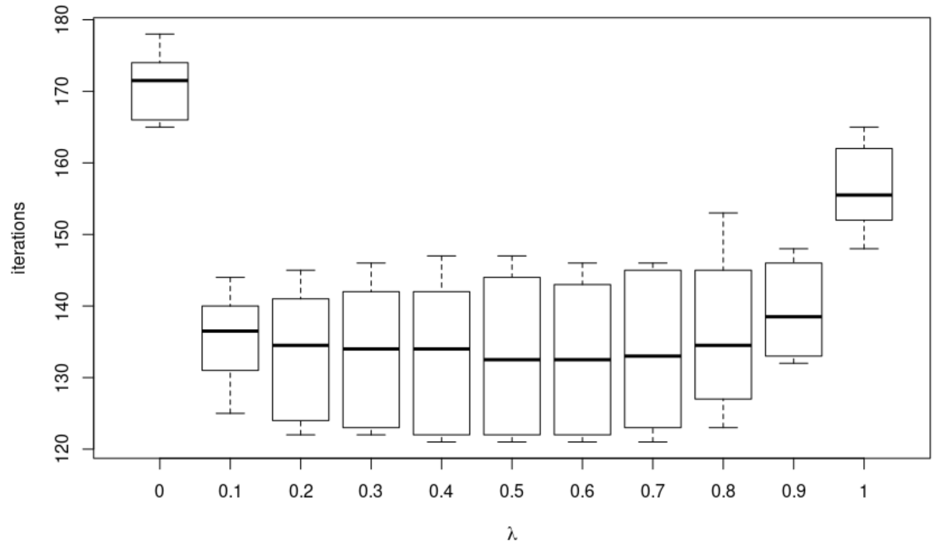

Figure 26: Performance - iterations

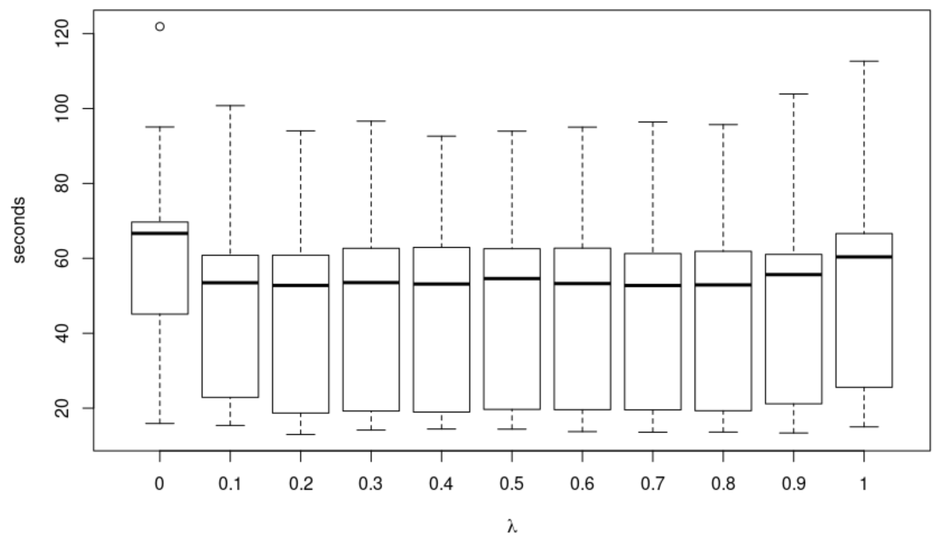

Figure 27: Performance - running time

7 Conclusion

In this thesis the theory behind Markov chains and multiobjective decision making was introduced and combined into an implementable Pareto-PageRank algorithm. The structure of this modified weighted PageRank algorithm was covered by introducing new elements to the traditional PageRank one by one. As a results, a multiobjective decision making tool was introduced as a pseudocode algorithm. The executable implementation was created with Python and the algorithm was tested on real world data. Even though the amount of openly available and suitable real world data was limited, the tests demonstrated the capacity of the algorithm on a bidimensional case.

The results of the tests were quite intuitive. Arcs with high value on both used attributes seemed to be important on large part of the Pareto-frontier while their position on graph also affected the importance through the nature of traditional PageRank. On the test cases the Pareto-PageRank algorithm did not seem to reveal anything extraordinaly information, regardless of that Pareto-PageRank is easy to implement and has a good enough performance with low number of attributes. Alongside linear scalarization, Chebyshev scalarization was used as a comparision. The results showed some differences but it is not possible no point out a superior method. It was shown that both methods are valid options and thus the used method can vary in different use cases.

For algorithm to be usable on real world scenarios the time complexity of the algorithm must be good enough. As discussed in chapter 5 the time complexity of the Pareto-PageRank is $O((|V|^2)^{|L|-1})$. The complexity of the real execution depends not only on the algorithm but also on the implementation. The author of this thesis has some experience on coding but the main field of studies have been applied mathematics. A more optimized code could lead to faster execution but the algorithm is simple enought that even with moderate experience it was possible to implement a working solution with decent performance.

References

[1] A. Bagirov, N. Karmitsa, M. Mäkelä: *Introduction to Nonsmooth Optimization: Theory, Practice and Software*. Springer. 2014.

[2] N. Biggs, E. Lloyd, R. Wilson: *Graph Theory 1736-1936*. Oxford University Press. 1976.

[3] A. Cardillo, S. Scellato, V. Latora, S. Porta: *Structural properties of planar graphs of urban street patterns*. Physical Revies E 73, 066107. 2006.

[4] J. Chan, K. Teknomo: *Hub Identification of the Metro Manila Road Network Using PageRank*. AYRF15-015, 2015.

[5] Y. Collette, P. Siarry: *Multiobjective Optimization: Principles and Case Studies*. Springer Science & Business Media. 2013.

[6] P. Gagniuc: *Markov Chains: From Theory to Implementation and Experimentation*. John Wiley & Sons, Incorporated. 2017.

[7] W. Hamilton, R. Ying, J. Leskovec: *Inductive Representation Learning on Large Graphs*. Department of Computer Science, Stanford University. 2018.

[8] J. Heidemann, M. Klier, F. Probst: *Identifying Key Users in Online Social Networks: A PageRank Based Approach*. ICIS 2010 Proceedings, 79. 2010.

[9] T. Jepsen, C. Jensen, T. Dyhre, K. Torp: *On Network Embedding for Machine Learning on Road Networks: A Case Study on the Danish Road Network*. Department of Computer Science, Aalborg University. 2019.

[10] M. Kas: *Structures and Statistics of Citation Networks*, Submitted in partial fulfillment of the requirements for the degree of Master of Science in Electrical and Computer Engineering. 2011.

[11] J. Koski: *Defectiveness of weighting method in multicriterion optimization of structures*. Communications in Applied Numerical Methods, Vol. 1, pp. 333-337. 1985.

[12] J. H. McDonald: *Handbook of Biological Statistics* (3rd ed.). Sperky House Publishing. 2014.

[13] K. Miettinen: *Nonlinear Multiobjective Optimization*. Springer. 1998.

[14] S. Porta, P. Crucitti, V. Latora: *The Network Analysis of Urban Streets: A Dual Approach* Physica A. Statistical Mechanics and its Applications., Vol. 369, Iss. 2, pp. 853-866. 2006.

[15] X. Prats, V. Puig, J. Quivedo, F. Nejjari: *Lexicographic optimization for optimal departure aircraft trajectories*. Aerospace Science and Technology Vol. 14, pp. 26-37. 2010.

[16] A. Punkka: *MS-E2134- Decision Making and Problem Solving D*. Lecture slides. Aalto University. 2021.

[17] F. Rimbach, M. Dannenberg, U. Bleimann: *Page ranking and topic-sensitive page ranking: micro-changes and macro-inpact*. Internet Research Vol. 17 No. 1. 2007.

[18] U. Senanayake, M. Piraveenan, A. Zomaya: *The Pagerank-Index: Going beyond Citation Counts in Quantifying Scientific Impact of Researchers*, PLoS ONE 10(8): e0134794. 2015.

[19] P. Sohouenou, P. Christidis, A. Christodoulou, L. Neves, D. Presti: *Using a rangom road graph model to understand road networks robustness to link failures*. International Journal of Critical Infrastructure Protection, Vol. 29. 2020.

[20] M. Taylor, S. Sekhar, G. D'Este: *Application of Accessibility Based Methods for Vulnerability Analysis of Strategic Road Networks*. Springer Science. 2006.

[21] O. Tuhkanen: *Pareto-PageRank thesis: files and codes on GitLab.* https://gitlab.utu.fi/oetuhk/gradu.[1]

[22] S. Wang: *Ranking The Best UFC Fighters Using PageRank and Neo4j.* https://towardsdatascience.com/ranking-the-best-ufc-fighters-using-pagerank-and-neo4j-5385805b4515.[1]

[23] W. Xing, A. Ghorbani: *Weighted PageRank Algorithm.* Second Annual Conference on Communication Networks and Services Research. pp. 305-314. 2004.

[24] C. Zhao, N. Li, D. Fang: *Criticality assessment of urban interdependent lifeline systems using biased PageRank algorithm and a multilayer weighted directed network model.* International Journal of Critical Infrastructure Protection, Vol. 22, pp. 100-112. 2018.

[25] https://www.openstreetmap.org.[1]

[26] https://wiki.openstreetmap.org/wiki/Key:highway.[1]

[27] https://geojson.io.[1]

[28] https://seaborn.pydata.org/tutorial/color_palettes.html.[1]

[29] https://www.kuntaliitto.fi/tietotuotteet-ja-palvelut/kaupunkien-ja-kuntien-lukumaarat-ja-vaestotiedot.[1]

[1] Visited 13.7.2022

A Data tables

A.1 Data - Helsinki

There are 91542 vertices and 248211 arcs in the Helsinki -test data. The average degree of a vertex is 2.71.

Roadtype	**Arcs in graph**
motorway / motorway_link	139
trunk / trunk_link	179
primary / primary_link	1799
secondary / secondary_link	3001
tertiary / tertiary_link	5049
unclassified	4300
residential	18855
living_street	761
service	35814
services	6
cycleway	28091
pedestrian	7517
track	1186
footway	117035
bridleway	162
steps	9605
corridor	252
path	12882
platform	1578

Table 7: Data - Helsinki

A.2 Data - Espoo

There are 64323 vertices and 168461 arcs in the Espoo -test data. The average degree is 2.62.

Roadtype	**Arcs in graph**
motorway / motorway_link	212
trunk / trunk_link	162
primary / primary_link	68
secondary / secondary_link	3243
tertiary / tertiary_link	3861
unclassified	1456
residential	25134
living_street	356
service	49737
services	8
cycleway	33242
pedestrian	836
track	1856
footway	31239
bridleway	132
steps	2155
corridor	126
path	13900
platform	724
bus stop	14

Table 8: Data - Espoo

A.3 Data - Tampere

There are 66064 vertices and 183957 arcs in the Tampere -test data. The average degree is 2.78.

Roadtype	**Arcs in graph**
motorway / motorway_link	272
trunk / trunk_link	464
primary / primary_link	597
secondary / secondary_link	2649
tertiary / tertiary_link	6020
unclassified	3425
residential	28595
living_street	454
service	32804
services	4
cycleway	34517
pedestrian	792
track	6046
footway	23495
bridleway	20
steps	1018
corridor	246
path	42487
platform	44
bus stop	6
rest area	2

Table 9: Data - Tampere

A.4 Data - Vantaa

There are 118078 vertices and 313326 arcs in the Vantaa -test data. The average degree is 2.65.

Roadtype	**Arcs in graph**
motorway / motorway_link	403
trunk / trunk_link	527
primary / primary_link	70
secondary / secondary_link	8523
tertiary / tertiary_link	9582
unclassified	7821
residential	44513
living_street	536
service	75933
services	4
cycleway	63684
pedestrian	1960
track	5624
footway	60542
bridleway	426
steps	2023
corridor	2
path	29245
platform	1888
bus stop	0
rest area	20

Table 10: Data - Vantaa

A.5 Data - Oulu

There are 66889 vertices and 179750 arcs in the Oulu -test data. The average degree is 2.69.

Roadtype	Arcs in graph
motorway / motorway_link	219
trunk / trunk_link	288
primary / primary_link	0
secondary / secondary_link	2864
tertiary / tertiary_link	6135
unclassified	2714
residential	29902
living_street	716
service	36068
services	2
cycleway	35807
pedestrian	736
track	12950
footway	30767
bridleway	316
steps	608
corridor	180
path	19236
platform	222
bus stop	14
rest area	6

Table 11: Data - Oulu

A.6 Data - Turku

There are 64392 vertices and 174180 arcs in the Turku -test data. The average degree is 2.70.

Roadtype	**Arcs in graph**
motorway / motorway_link	195
trunk / trunk_link	530
primary / primary_link	750
secondary / secondary_link	2563
tertiary / tertiary_link	8421
unclassified	5173
residential	31386
living_street	1000
service	40520
services	24
cycleway	26113
pedestrian	774
track	2862
footway	36422
bridleway	48
steps	1690
corridor	232
path	14719
platform	748
bus stop	0
rest area	10

Table 12: Data - Turku

A.7 Data - Jyväskylä

There are 46711 vertices and 122363 arcs in the Jyväskylä -test data. The average degree is 2.62.

Roadtype	**Arcs in graph**
motorway / motorway_link	95
trunk / trunk_link	657
primary / primary_link	0
secondary / secondary_link	3665
tertiary / tertiary_link	6160
unclassified	7263
residential	20310
living_street	500
service	23609
services	4
cycleway	18238
pedestrian	1050
track	8934
footway	13314
bridleway	0
steps	650
corridor	46
path	17774
platform	8
bus stop	0
rest area	86

Table 13: Data - Jyväskylä

A.8 Data - Kuopio

There are 27145 vertices and 74127 arcs in the Kuopio -test data. The average degree is 2.73.

Roadtype	**Arcs in graph**
motorway / motorway_link	171
trunk / trunk_link	88
primary / primary_link	0
secondary / secondary_link	1283
tertiary / tertiary_link	3738
unclassified	4121
residential	12021
living_street	198
service	13054
services	6
cycleway	12015
pedestrian	246
track	4346
footway	12376
bridleway	2
steps	384
corridor	228
path	9850
platform	0
bus stop	0
rest area	0

Table 14: Data - Kuopio

A.9 Data - Lahti

There are 20536 vertices and 57870 arcs in the Lahti -test data. The average degree is 2.82.

Roadtype	**Arcs in graph**
motorway / motorway_link	117
trunk / trunk_link	116
primary / primary_link	0
secondary / secondary_link	1138
tertiary / tertiary_link	2719
unclassified	1173
residential	10350
living_street	50
service	8847
services	0
cycleway	14374
pedestrian	334
track	982
footway	7734
bridleway	0
steps	154
corridor	0
path	9722
platform	60
bus stop	0
rest area	0

Table 15: Data - Lahti

A.10 Data - Pori

There are 17005 vertices and 46622 arcs in the Pori -test data. The average degree is 2.74.

Roadtype	**Number of arcs in graph**
motorway / motorway_link	0
trunk / trunk_link	466
primary / primary_link	43
secondary / secondary_link	2056
tertiary / tertiary_link	3332
unclassified	1915
residential	11100
living_street	94
service	7460
services	8
cycleway	10226
pedestrian	36
track	2992
footway	2550
bridleway	54
steps	36
corridor	0
path	4218
platform	0
bus stop	0
rest area	0

Table 16: Data - Pori

B Result tables

B.1 Helsinki

	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
motorway / motorway_link											
trunk / trunk_link											2
primary / primary_link										2	6
secondary / secondary_link										1	2
tertiary / tertiary_link									2	14	21
unclassified											7
residential								4	8	10	13
living_street											
service											2
cycleway	6	48	92	100	100	100	100	96	90	73	47
pedestrian	2	2									
track	4	2									
footway	42	30	4								
bridleway											
steps	2										
corridor	2										
path	42	18	4								
crossing											

Table 17: Results with linear scalarization - Helsinki

	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
motorway / motorway_link							1	1	1	1	
trunk / trunk_link							5	4	4	5	2
primary / primary_link							4	4	6	6	6
secondary / secondary_link							3	3	3	3	2
tertiary / tertiary_link							4	8	12	15	21
unclassified							6	7	5	5	7
residential							20	17	16	16	13
living_street											
service			2	2	6	6	4	2	2	2	2
cycleway	6	6	6	6	6	6	53	54	51	47	47
pedestrian	2	2	2	2	2	2					
track	4	6	6	6	6	4					
footway	42	40	42	42	42	42					
bridleway											
steps	2	2	2	2	2	2					
corridor	2	2									
path	42	40	42	42	42	42					
platform		2	2	2	2	2					
crossing											

Table 18: Results with Chebyshev scalarization - Helsinki

B.2 Espoo

	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
motorway / motorway_link											7
trunk / trunk_link										4	12
primary / primary_link											
secondary / secondary_link											10
tertiary / tertiary_link											6
unclassified									2	2	2
residential								4	10	16	26
living_street											
service											
cycleway	2	72	98	100	100	100	100	96	88	78	37
pedestrian											
track	2	2									
footway	82	22									
bridleway											
steps											
corridor	2										
path	12	4	2								
crossing											

Table 19: Results with linear scalarization - Espoo

	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
motorway / motorway_link							5	9	10	10	7
trunk / trunk_link						2	9	11	13	13	12
primary / primary_link											
secondary / secondary_link							8	8	9	11	10
tertiary / tertiary_link								2	4	4	6
unclassified							8	6	6	2	2
residential						12	28	24	26	26	26
living_street											
service			4	6	6	8	2	2	2	2	
cycleway	2	2	2	2		2	40	38	30	32	37
pedestrian											
track	2	4	8	10	8	10					
footway	82	80	70	64	66	46					
bridleway											
steps											
corridor	2										
path	12	14	16	18	20	20					
crossing											

Table 20: Results with Chebyshev scalarization - Espoo

B.3 Tampere

	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
motorway / motorway_link											
trunk / trunk_link											4
primary / primary_link											
secondary / secondary_link											
tertiary / tertiary_link											4
unclassified										4	12
residential								6	14	42	62
living_street											
service											
cycleway		60	92	100	100	100	100	94	86	54	18
pedestrian											
track	26	10	6								
footway	50	26	2								
bridleway											
steps											
corridor	4										
path	20	4									
crossing											

Table 21: Results with linear scalarization - Tampere

	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
motorway / motorway_link											
trunk / trunk_link						2		2	3	3	4
primary / primary_link											
secondary / secondary_link							2	2			
tertiary / tertiary_link							6	4	10	6	4
unclassified							6	6	8	12	12
residential						18	64	60	56	57	62
living_street											
service					4	4					
cycleway				2	2	4	22	26	23	22	18
pedestrian											
track	26	24	22	22	22	16					
footway	50	52	52	52	50	44					
bridleway											
steps											
corridor	4	2	2	2	2	2					
path	20	22	24	22	20	10					
crossing											

Table 22: Results with Chebyshev scalarization - Tampere

B.4 Vantaa

	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
motorway / motorway_link											3
trunk / trunk_link										2	6
primary / primary_link											
secondary / secondary_link										1	3
tertiary / tertiary_link								2	2	2	
unclassified											6
residential							2	4	34	63	70
living_street											
service											
cycleway	12	57	93	98	100	100	98	94	64	32	12
pedestrian											
track	4	2									
footway	46	22									
bridleway	8										
steps											
corridor											
path	30	19	7	2							
crossing											

Table 23: Results with linear scalarization - Vantaa

	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
motorway / motorway_link							3	4	6	5	3
trunk / trunk_link						2	7	8	9	9	6
primary / primary_link											
secondary / secondary_link						1	2	2	4	3	3
tertiary / tertiary_link							5	4	5	5	
unclassified						2	4	4	4	6	6
residential						4	58	60	56	60	70
living_street											
service				4	7	8					
cycleway	12	8	10	6	6	4	21	18	16	12	12
pedestrian											
track	4	4	4	4	4	4					
footway	46	50	50	52	52	46					
bridleway	8	8	8	4	2						
steps											
corridor											
path	30	30	28	30	29	29					
crossing											

Table 24: Results with Chebyshev scalarization - Vantaa

B.5 Oulu

	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
motorway / motorway_link											5
trunk / trunk_link											
primary / primary_link											
secondary / secondary_link											
tertiary / tertiary_link				1	2	2	2	2	6	6	6
unclassified											
residential							4	6	24	38	58
living_street											
service											
cycleway	10	66	96	95	98	98	94	92	70	56	31
pedestrian											
track	28	10									
footway	34	12	2	2							
bridleway											
steps											
corridor											
path	28	12	2	2							
crossing											

Table 25: Results with linear scalarization - Oulu

	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
motorway / motorway_link							1	4	6	6	5
trunk / trunk_link											
primary / primary_link											
secondary / secondary_link											
tertiary / tertiary_link					2	2	2	4	9	7	6
unclassified						2					
residential						22	68	67	58	54	58
living_street											
service			2	6	6	6					
cycleway	10	14	11	9	7	7	29	25	27	33	31
pedestrian											
track	28	26	26	19	20	20					
footway	34	36	42	46	46	28					
bridleway											
steps											
corridor											
path	28	24	19	20	19	13					
crossing											

Table 26: Results with Chebyshev scalarization - Oulu

B.6 Turku

	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
motorway / motorway_link											
trunk / trunk_link											
primary / primary_link											
secondary / secondary_link											
tertiary / tertiary_link											
unclassified											
residential							4	24	52	70	82
living_street											
service											
cycleway		48	100	100	100	100	96	76	48	30	18
pedestrian											
track	6										
footway	78	42									
bridleway											
steps											
corridor	4	2									
path	12	8									
crossing											

Table 27: Results with linear scalarization - Turku

	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
motorway / motorway_link											
trunk / trunk_link											
primary / primary_link							2	2			
secondary / secondary_link									2	2	
tertiary / tertiary_link										2	
unclassified						8					
residential						44	96	92	92	88	82
living_street											
service											
cycleway			2	2		2	2	6	6	8	18
pedestrian											
track	6	2	4	4	4	2					
footway	78	78	74	76	76	28					
bridleway											
steps					2						
corridor	4	4	2	2	2	2					
path	12	16	18	16	16	14					
crossing											

Table 28: Results with Chebyshev scalarization - Turku

B.7 Jyväskylä

	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
motorway / motorway_link											
trunk / trunk_link											
primary / primary_link											
secondary / secondary_link											
tertiary / tertiary_link										4	10
unclassified									2	12	44
residential			4	4	6	6	10	16	32	60	42
living_street											
service		2	2	2	2	2	2	2			
cycleway	8	66	88	94	92	92	88	82	66	24	4
pedestrian	10	6									
track	28	6									
footway	34	14	4								
bridleway											
steps											
corridor											
path	20	6	2								
crossing											

Table 29: Results with linear scalarization - Jyväskylä

	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
motorway / motorway_link											
trunk / trunk_link											
primary / primary_link											
secondary / secondary_link											
tertiary / tertiary_link						2	12	12	12	12	10
unclassified						12	38	34	36	40	44
residential						28	48	50	48	44	42
living_street											
service				2	2	2					
cycleway	8	8	8	6	8	8	2	4	4	4	4
pedestrian	10	6	6	6	2						
track	28	26	26	26	24	12					
footway	34	32	32	28	28	14					
bridleway											
steps											
corridor											
path	20	28	28	32	36	22					
crossing											

Table 30: Results with Chebyshev scalarization - Jyväskylä

B.8 Kuopio

	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
motorway / motorway_link											
trunk / trunk_link											
primary / primary_link											
secondary / secondary_link										2	8
tertiary / tertiary_link										6	8
unclassified									6	16	54
residential				2	4	6	6	16	68	72	30
living_street											
service											
cycleway	2	50	96	94	94	94	94	84	26	4	
pedestrian											
track	54	18									
footway	32	22	4	4	2						
bridleway											
steps											
corridor											
path	12	10									
crossing											

Table 31: Results with linear scalarization - Kuopio

	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
motorway / motorway_link							1	1	1	1	
trunk / trunk_link											
primary / primary_link											
secondary / secondary_link						4	4	3	4	6	8
tertiary / tertiary_link						8	6	8	10	10	8
unclassified						12	45	52	53	52	54
residential						44	44	36	32	31	30
living_street											
service											
cycleway	2	4	4	4	4						
pedestrian											
track	54	44	36	36	36	12					
footway	32	36	42	42	42	16					
bridleway											
steps											
corridor		2									
path	12	14	18	18	18	4					
crossing											

Table 32: Results with Chebyshev scalarization - Kuopio

B.9 Lahti

	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
motorway / motorway_link											
trunk / trunk_link											1
primary / primary_link											
secondary / secondary_link											2
tertiary / tertiary_link											
unclassified									2	2	2
residential							4	18	42	60	71
living_street											
service											2
cycleway	2	76	98	100	100	100	96	82	56	38	22
pedestrian											
track	22	6									
footway	36	12	2								
bridleway											
steps											
corridor											
path	40	6									
crossing											

Table 33: Results with linear scalarization - Lahti

	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
motorway / motorway_link						1	1	2	2	2	
trunk / trunk_link							1	1	1	1	1
primary / primary_link											
secondary / secondary_link										2	2
tertiary / tertiary_link											
unclassified							6	4	4	4	2
residential						46	64	64	68	65	71
living_street											
service						2	2	4	4	4	2
cycleway	2	6	8	10	14	11	26	25	21	22	22
pedestrian											
track	22	22	20	20	20	8					
footway	36	42	46	44	42	18					
bridleway											
steps											
corridor											
path	40	30	26	26	24	14					
crossing											

Table 34: Results with Chebyshev scalarization - Lahti

B.10 Pori

	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
motorway / motorway_link											
trunk / trunk_link											4
primary / primary_link											
secondary / secondary_link											4
tertiary / tertiary_link											8
unclassified											4
residential									12	52	68
living_street											
service											
cycleway		62	100	100	100	100	100	100	88	48	12
pedestrian											
track	34	14									
footway	30	10									
bridleway											
steps											
corridor											
path	36	14									
crossing											

Table 35: Results with linear scalarization - Pori

	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
motorway / motorway_link											
trunk / trunk_link						2	2	2	4	4	4
primary / primary_link											
secondary / secondary_link								2	2	2	4
tertiary / tertiary_link						2	8	10	8	8	8
unclassified						26	16	12	12	8	4
residential						30	58	58	62	70	68
living_street											
service					2						
cycleway				2	4	4	16	16	12	8	12
pedestrian											
track	34	32	34	34	32	22					
footway	30	32	36	36	32	8					
bridleway											
steps											
corridor											
path	36	36	30	28	30	6					
crossing											

Table 36: Results with Chebyshev scalarization - Pori

B.11 Combined

	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
motorway / motorway_link											15
trunk / trunk_link										6	29
primary / primary_link										2	6
secondary / secondary_link										4	29
tertiary / tertiary_link				1	2	2	2	4	10	32	63
unclassified									12	36	131
residential			4	6	10	12	30	98	296	483	522
living_street											
service		2	2	2	2	2	2	2			4
cycleway	42	605	953	981	984	984	966	896	682	437	201
pedestrian	12	8									
track	208	70	6								
footway	464	212	18	6	2						
bridleway	8										
steps	2										
corridor	12	2									
path	252	101	17	4							
platform											
crossing											

Table 37: Combined results - Linear scalarization

	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
motorway / motorway_link						1	12	21	26	25	15
trunk / trunk_link						8	24	28	34	35	29
primary / primary_link							6	6	6	6	6
secondary / secondary_link						5	19	20	24	29	29
tertiary / tertiary_link					2	14	43	52	70	69	63
unclassified						62	129	125	128	129	131
residential						248	548	528	514	511	522
living_street											
service			8	20	33	36	8	8	8	8	4
cycleway	42	48	51	49	51	48	211	212	190	188	201
pedestrian	12	8	8	8	4	2					
track	208	190	186	181	176	110					
footway	464	478	486	482	476	290					
bridleway	8	8	8	4	2						
steps	2	2	2	2	4	2					
corridor	12	10	4	4	4	4					
path	252	254	245	248	246	168					
platform		2	2	2	2	2					
crossing											

Table 38: Combined results - Chebyshev scalarization