



Oliver Unterlercher, BSc

**Bewertung und Analyse von Wanderwegen im
Zusammenhang mit multikriterieller Routenplanung**

Masterarbeit

zur Erlangung des akademischen Grades

Diplom-Ingenieur

Masterstudium Geomatics Science

eingereicht an der

Technischen Universität Graz

Betreuer

Ao.Univ.-Prof. Dipl.-Ing. Dr.techn. Manfred Wieser

Institut für Geodäsie

Graz, März 2017

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Oliver Unterlercher
Graz, März 2017

Danksagung

Ich möchte mich bei meinem Betreuer Herrn Prof. Wieser für die Zusammenarbeit bedanken. Ein Danke, dass ich die Möglichkeit bekommen habe ein Thema meiner Wahl bei ihm und dem Institut für Geodäsie zu schreiben.

Weiteres möchte ich mich beim Tourismusverband Osttirol für die bereitgestellten Daten, die mir die Durchführung erheblich erleichtert haben, bedanken.

Darüber hinaus möchte ich mich vor allem bei meinen Eltern, meiner Familie und meinen Freunden für die Unterstützung bedanken.

Oliver Unterlercher
Graz, März 2017

Kurzfassung

Routenplanung gibt es mittlerweile für verschiedenste Anwendungen, sei es die Routenplanung für Verkehrsmittel, für Roboter oder für Fußgänger. Ziel dieser Diplomarbeit ist es, eine multikriterielle Routenplanung für Fußgänger auf Wanderwegen in Osttirol zu entwickeln. Als Datenbasis dienen geometrisch definierte Knoten entlang verschiedener Wanderwege in der Tourismusregion Osttirol. Ausgehend davon sollen Wegenetze erstellt werden, wobei ein wichtiger Aspekt die Bewertung der Knoten-Kanten Struktur darstellt. Es gibt verschiedene Algorithmen, die die Routenplanung je nach Problemstellung und Optimierung einsetzt, sei es der Dijkstra-Algorithmus oder verschiedenste heuristische Ansätze, um Aufgabenstellungen, wie zum Beispiel das „shortest path problem“, zu lösen. Allerdings ist dies in den meisten Fällen eine „single-criterion“ Optimierung.

Ziel dieser Arbeit ist es, mehrere Kriterien in die Routenplanung einfließen zu lassen und verschiedene Wege miteinander zu vergleichen. Es soll die Möglichkeit bestehen, vordefinierte Einstellungskriterien auszuwählen, um so benutzerspezifisch den geeignetsten Wanderweg zu finden. Kriterien, die auf Wanderwegen eine besondere Rolle spielen, sind zum Beispiel die Distanz, die Steigung, die Untergrundbeschaffenheit und die Höhe in der man sich bewegt, um einige zu nennen.

Als Ergebnis soll ein Anwendungsprogramm entstehen, das eine benutzerspezifische Routenplanung für Wanderwege ermöglicht. Die optimalen Wanderrouten können somit vor der Begehung der Wege berechnet und dargestellt werden, wobei auch die genaue Bestimmung der Gehzeit einen wichtigen Aspekt darstellt.

Abstract

Route planning exists for various applications such as route planning for vehicles, for robots or for pedestrians. The aim of this master thesis is the development and implementation of a multi-criteria route planner for hikers in an established hiking area in Osttirol. The databasis consists of points containing coordinates and height values along different hiking paths. Based on these datapoints, a network consisting of nodes and edges will be generated. An important part is the valuation of the edges in the hiking network. There are many different algorithms that are used in route planning, depending on the optimization and the problem statement. To solve problems like the shortest path problem, algorithms like the Dijkstra algorithm or different heuristic methods can be used. However these algorithms deal only with one criterion most of the time.

The aim of this thesis is to develop a route planner that includes more than one criterion simultaneously and to compare different routes. The possibility to select the criteria and different optimization methods should enable the user to find user-specific the best path as a pre-trip calculation. Relating to hiking paths, the criteria that play a major role are for example the distance, the inclination, the ground composition and the height in which the path is located.

The outcome of this master thesis will be a user-specific route planner for hiking paths. The optimal routes can be calculated and visualised. Moreover also the determination of the walking time is an important aspect.

Abkürzungsverzeichnis

AJAX	Asynchronous JavaScript und XML
CEN	European Committee for Standardization
CGI	Common Gateway Interface
DGM	Digitales Geländemodell
DIN	Deutsches Institut für Normung
DSPA	Dijkstra's Shortest Path Algorithmus
EN	Europäische Norm
GIS	Geographisches Informationssystem
GK	Gauß-Krüger
GM	Geländehöhenmodell
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
ISO	International Organization for Standardization
KM	Kartographisches Modell
LM	Landschaftsmodell
MOLSA	Multiobjective Label-Setting Algorithmus
MOSP	Multi-Objective Shortest Path
MSPP	Multikriterielles Shortest Path Problem
OGC	Open Geospatial Consortium
ON	Österreichisches Normungsinstitut
SA	Spatial Analysis
SP	Shortest Path
SPP	Shortest Path Problem
SSPP	Single Criterion Shortest Path Problem
TSP	Travelling Salesman Problem
WGS-84	World Geodetic System 1984
WMS	Web Map Server/Web Map Service
WSM	Weighted Sum Method

Abbildungsverzeichnis

Abbildung 1: Brücken von Königsberg.....	7
Abbildung 2: Graph von Brücken von Königsberg.....	7
Abbildung 3: Wichtige Graphenelemente.....	8
Abbildung 4: Vollständiger Graph	10
Abbildung 5: Planarer Graph (links) - nicht-planarer Graph (rechts).....	11
Abbildung 6: Gerichteter Graph	12
Abbildung 7: Zyklischer Graph (links) - azyklischer Graph (rechts).....	12
Abbildung 8: Bewerteter Graph	13
Abbildung 9: Dreiecksungleichung	13
Abbildung 10: Beispielgraph mit Knotennummern und Kantenkosten	22
Abbildung 11: Prozedere des Dijkstra Algorithmus	24
Abbildung 12: Kriterienwerte - Autobeispiel.....	29
Abbildung 13: Pareto-optimale Lösungen.....	32
Abbildung 14: Kostenvektoren für die Kanten.....	34
Abbildung 15: Alle Pareto-optimale Lösungen des Beispielgraphen von Abbildung 14.....	35
Abbildung 16: Illustration von Pareto-optimale Lösungen vom Knoten 3 von Abbildung 15.....	36
Abbildung 17: Initialisierung und erster Schritt des erweiterten Dijkstra Algorithmus, um alle Pareto-optimale Wege zu finden, mit 2 Kriterien.....	41
Abbildung 18: Zweiter und dritter Schritt des erweiterten Dijkstra Algorithmus, um alle Pareto-optimale Wege zu finden, mit 2 Kriterien.....	42
Abbildung 19: Vierter und fünfter Schritt des erweiterten Dijkstra Algorithmus, um alle Pareto-optimale Wege zu finden, mit 2 Kriterien.....	42
Abbildung 20: Sechster und siebter Schritt des erweiterten Dijkstra Algorithmus, um alle Pareto-optimale Wege zu finden, mit 2 Kriterien.....	43
Abbildung 21: Achter und neunter Schritt des erweiterten Dijkstra Algorithmus, um alle Pareto-optimale Wege zu finden, mit 2 Kriterien.....	43
Abbildung 22: Zehnter und elfter Schritt des erweiterten Dijkstra Algorithmus, um alle Pareto-optimale Wege zu finden, mit 2 Kriterien.....	44
Abbildung 23: Approximationsmethoden der multikriteriellen Optimierung für die Routenplanung.	46
Abbildung 24: Ablauf des erweiterten Dijkstra Algorithmus um eine Pareto-optimale Lösung zu berechnen (Lexikographisch mit f_1 ="Gehzeit" und f_2 ="Schöne Aussicht").....	49
Abbildung 25: Ablauf des erweiterten Dijkstra Algorithmus um eine Pareto-optimale Lösung zu berechnen (Lexikographisch mit f_1 ="Schöne Aussicht" und f_2 ="Gehzeit")	50
Abbildung 26: Lexikographische Minimumslösungen.....	51
Abbildung 27: Flowchart eines Datenmodellierungsprozesses	54
Abbildung 28: Geographische Vektordaten und deren Attribute.....	55
Abbildung 29: Geographische Daten für die Routenplanung inklusive Attributinformationen	56
Abbildung 30: Ausschnitt aus "www.geoland.at" - Digitalisierung eines Weges (1)	59
Abbildung 31: Ausschnitt aus "www.geoland.at" - Digitalisierung eines Weges (2)	60
Abbildung 32: Ausschnitt aus "www.geoland.at" - Digitalisierung eines Weges (3)	60
Abbildung 33: Ausschnitt aus "www.geoland.at" - Digitalisierung eines Weges (4)	61

Abbildung 34: Ausschnitt aus "www.geoland.at" - Digitalisierung eines Weges (Höhenwerte)	61
Abbildung 35: Übersichtskarte des Wandergebietes Defereggental	78
Abbildung 36: Übersichtskarte des Wandergebietes Defereggental (Orthofotoausschnitt als Hintergrund)	78
Abbildung 37: Übersichtskarte des Wandergebietes Defereggental mit Gemeinden (Orthofotoausschnitt als Hintergrund)	79
Abbildung 38: Lage des Wandergebietes innerhalb Osttirols	79
Abbildung 39: Übersichtskarte des Wandergebietes Defereggental mit Knotennummern	80
Abbildung 40: Ablauf eines WMS	81
Abbildung 41: Ausschnitt aus Matlab – Anzahl effizienter Wege (Startknoten 100, Gehzeit & Weglänge, Approximation)	96
Abbildung 42: Ausschnitt aus Matlab – Anzahl effizienter Wege (Startknoten 100, Weglänge & Gehzeit, Approximation)	96
Abbildung 43: Ausschnitt aus Matlab – Anzahl effizienter Wege (Startknoten 100, Gehzeit & Weglänge, alle Lösungen)	97
Abbildung 44: Ausschnitt aus Matlab – Anzahl effizienter Wege (Startknoten 100, Gehzeit & Höhenunterschied, alle Lösungen)	98
Abbildung 45: Ausschnitt aus Matlab – Anzahl effizienter Wege (Startknoten 100, Höhenunterschied & Weglänge, alle Lösungen)	98
Abbildung 46: Ausschnitt aus Matlab – Anzahl effizienter Wege (Startknoten 100, Höhenunterschied & Weglänge & Gehzeit, alle Lösungen)	99
Abbildung 47: Ausschnitt aus Matlab – Anzahl effizienter Wege (Startknoten 100, Standardpfad & Weglänge & Gehzeit, alle Lösungen)	99
Abbildung 48: Übersichtskarte - optimale Route Nr.1 (St.Veit-Speikboden)	106
Abbildung 49: Ausschnitt aus Übersichtskarte - optimale Route Nr.1 (St.Veit-Speikboden)	107
Abbildung 50: Web Map Display (World Street Map) - optimale Route Nr.1 (St.Veit-Speikboden) ..	107
Abbildung 51: Web Map Display (World Topographic Map) - optimale Route Nr.1 (St.Veit-Speikboden)	108
Abbildung 52: Web Map Display (Open Street Map) - Hütte entlang des Weges (Speikbodenhütte) ..	109
Abbildung 53: Übersichtskarte - optimale Route Nr.2 (St.Veit-Speikboden)	110
Abbildung 54: Ausschnitt aus Übersichtskarte - optimale Route Nr.2 (St.Veit-Speikboden)	110
Abbildung 55: Web Map Display (World Street Map) - optimale Route Nr.2 (St.Veit-Speikboden) ..	111
Abbildung 56: Übersichtskarte - optimale Route (Leppetalalm-Mooseralm)	113
Abbildung 57: Ausschnitt aus Übersichtskarte - optimale Route (Leppetalalm-Mooseralm)	113
Abbildung 58: Web Map Display (World Street Map) - optimale Route (Leppetalalm-Mooseralm) .	114
Abbildung 59: Web Map Display (World Imagery) - optimale Route (Leppetalalm-Mooseralm)	114
Abbildung 60: Übersichtskarte - optimale Route Nr.1 (Start Zwenewaldweg-Leisacher Alm)	116
Abbildung 61: Übersichtskarte - optimale Route Nr.2 (Start Zwenewaldweg-Leisacher Alm)	116
Abbildung 62: Übersichtskarte - optimale Route Nr.3 (Start Zwenewaldweg-Leisacher Alm)	117
Abbildung 63: Übersichtskarte - optimale Route Nr.4 (Start Zwenewaldweg-Leisacher Alm)	117
Abbildung 64: optimale Route Nr.1	118
Abbildung 65: optimale Route Nr.2	118
Abbildung 66: optimale Route Nr.3	119
Abbildung 67: optimale Route Nr.4	119
Abbildung 68: WMS Ausschnitt (Open Street Map) - optimale Route Nr.1	120
Abbildung 69: WMS Ausschnitt (Open Street Map) - optimale Route Nr.2	120

Abbildung 70: WMS Ausschnitt (Open Street Map) - optimale Route Nr.3.....	121
Abbildung 71: WMS Ausschnitt (Open Street Map) - optimale Route Nr.4.....	121
Abbildung 72: Übersichtskarte - optimale Route Nr.1 (St. Jakob Schule-Weiße Spitze)	125
Abbildung 73: Ausschnitt aus Übersichtskarte - optimale Route Nr.1 (St. Jakob Schule-Weiße Spitze)	126
Abbildung 74: Übersichtskarte - optimale Route Nr.2 (St. Jakob Schule-Weiße Spitze)	127
Abbildung 75: Ausschnitt aus Übersichtskarte - optimale Route Nr.2 (St. Jakob Schule-Weiße Spitze)	127
Abbildung 76: WMS Ausschnitt (World Street Map) - Vergleich der optimalen Routen.....	129
Abbildung 77: WMS Ausschnitt (Open Street Map) - Vergleich der optimalen Routen	129
Abbildung 78: WMS Ausschnitt (World Imagery) - Vergleich der optimalen Routen	130
Abbildung 79: WMS Ausschnitt (World Topographic Map) - Vergleich der optimalen Routen	130
Abbildung 80: Übersichtskarte - optimale Route Nr.1 (Brunnalm Talstation-Knoten Großer Leppleskofl)	131
Abbildung 81: Ausschnitt aus Übersichtskarte - optimale Route Nr.1 (Brunnalm Talstation-Knoten Großer Leppleskofl)	132
Abbildung 82: Web Map Display (World Imagery) - optimale Route Nr.1 (Brunnalm Talstation-Knoten Großer Leppleskofl)	133
Abbildung 83: Übersichtskarte - optimale Route Nr.2 (Brunnalm Talstation-Langschneid).....	134
Abbildung 84: Ausschnitt aus Übersichtskarte - optimale Route Nr.2 (Brunnalm Talstation- Langschneid).....	134
Abbildung 85: Web Map Display (World Imagery) - optimale Route Nr.2 (Brunnalm Talstation- Langschneid).....	136
Abbildung 86: Software Anforderungen für Matlab R2016a als 64-Bit Version (für Windows)	137
Abbildung 87: Software Anforderungen für Matlab R2016a als 64-Bit Version (für Mac)	137
Abbildung 88: Software Anforderungen für Matlab R2016a als 64-Bit Version (für Linux).....	138
Abbildung 89: GUI - Benutzereingabe in Matlab.....	138
Abbildung 90: GUI - beispielhafte Benutzereingabe (1).....	141
Abbildung 91: GUI - beispielhafte Benutzereingabe (2).....	141
Abbildung 92: Benutzerausgabe - Textdatei (1).....	142
Abbildung 93: Benutzerausgabe - Textdatei (2).....	142
Abbildung 94: Benutzerausgabe - Textdatei (3).....	143
Abbildung 95: Benutzerausgabe - Übersichtskarte (Ausschnitt).....	143
Abbildung 96: Benutzerausgabe - Routenverlauf im Wandernetz (grün).....	144
Abbildung 97: Benutzerausgabe - Routenverlauf im WMS (Vergleich zweier unterschiedlicher Baselayer (1)).....	144
Abbildung 98: Benutzerausgabe - Routenverlauf im WMS (Vergleich zweier unterschiedlicher Baselayer (2)).....	145
Abbildung 99: Benutzerausgabe - Ausschnitt aus dem WMS (Startknoten).....	145
Abbildung 100: Benutzerausgabe - Ausschnitt aus dem WMS (Zielknoten).....	146

Tabellenverzeichnis

Tabelle 1: Adjazenzliste/Bogenliste.....	16
Tabelle 2: Indizierte Adjazenzliste / Knotenliste	17
Tabelle 3: Indizierte Adjazenzliste / Bogenliste.....	17
Tabelle 4: Pseudocode Symbole.....	25
Tabelle 5: Kriterien und Alternativen des Autobespiels.....	28
Tabelle 6: Pareto-optimalen Wege vom Knoten 1 bis zum Knoten 4	41
Tabelle 7: Knoten- und Kantenanzahl des Wandernetzes	62
Tabelle 8: Parameter der Untergrundbeschaffenheit.....	70
Tabelle 9: Beispiel zur Berechnung der Gehzeit aus dem Wegehandbuch der Alpenvereine.....	71
Tabelle 10: Bestimmung des Höhenfaktors für die Berechnung der Gehzeit.....	72
Tabelle 11: Bestimmung des Steigungsfaktors für die Berechnung der Gehzeit.....	72
Tabelle 12: Bestimmung des Untergrundfaktors für die Berechnung der Gehzeit.....	72
Tabelle 13: Datenstruktur (indizierte Adjazenzliste / Knotenliste) - Ausgangsdaten für den Algorithmus	74
Tabelle 14: Datenstruktur (indizierte Adjazenzliste / Bogenliste) - Ausgangsdaten für den Algorithmus mit Gesamtstatistik	74
Tabelle 15: Datenstruktur - Ausgangsdaten für den Algorithmus mit Einzelstatistik	74
Tabelle 16: Eintrag aus Statistikspalte in Ausgangsdatenstruktur (Zellen 1-5).....	75
Tabelle 17: Eintrag aus Statistikspalte in Ausgangsdatenstruktur (Zellen 6-10).....	75
Tabelle 18: ID der POI's	75
Tabelle 19: ID des Untergrunds.....	76
Tabelle 20: Aufbau eines Labelvektors des erstellten Algorithmus zur Routenplanung mit multikriterieller Optimierung	84
Tabelle 21: Aufbau eines Beispiel-Labelvektors des Knoten 150 während der Berechnung des Algorithmus	85
Tabelle 22: Ergebnis Einzellösung	87
Tabelle 23: Ergebnis einer Teilmenge (1).....	88
Tabelle 24: Ergebnis einer Teilmenge (2).....	88
Tabelle 25: Ergebnis einer Teilmenge (3).....	89
Tabelle 26: Ergebnis aller Pareto-optimalen Lösungen.....	89
Tabelle 27: Ergebnis (Labelvektor) einer Einzellösung (Gehzeit, Weglänge (horizontal)) - 103 bis 34.	91
Tabelle 28: Ergebnis (Labelvektor) einer Einzellösung (Weglänge (schräg), Weglänge (horizontal)) - 103 bis 34	91
Tabelle 29: Kombinationen von Kriterien, ob alle Pareto-optimalen Wege berechnet werden können	93
Tabelle 30: Auswirkung der Rundung der Gehzeit auf die Ergebnisse	94
Tabelle 31: Auswirkung der Rundung der Weglänge auf die Ergebnisse.....	94
Tabelle 32: Anzahl der Möglichkeiten, die mehr als eine effiziente Lösung aufweisen - Gehzeit als Hauptkriterium.....	100
Tabelle 33: Anzahl der Möglichkeiten, die mehr als eine effiziente Lösung aufweisen - Weglänge als Hauptkriterium	102

Tabelle 34: Ergebnisse (Gehzeit & geometrische Kriterien) der optimalen Route Nr.1 (St.Veit-Speikboden).....	108
Tabelle 35: Ergebnisse (Untergrund & POI`s) der optimalen Route Nr.1 (St.Veit-Speikboden)	109
Tabelle 36: Ergebnisse (Gehzeit & geometrische Kriterien) der optimalen Route Nr.2 (St.Veit-Speikboden).....	111
Tabelle 37: Ergebnisse (Untergrund & POI`s) der optimalen Route Nr.1 (St.Veit-Speikboden)	112
Tabelle 38: Ergebnisse (Gehzeit & geometrische Kriterien) der optimalen Route (Leppetalm-Mooseralm)	115
Tabelle 39: Ergebnisse (Untergrund & POI`s) der optimalen Route (Leppetalm-Mooseralm)	115
Tabelle 40: Routenverläufe der 4 Lösungen mit Gehzeit & Steigung	122
Tabelle 41: Ergebnis Labelvektor Lösung Nr.1 (Zwenewaldweg-Leisacher Alm)	122
Tabelle 42: Ergebnis Labelvektor Lösung Nr.2 (Zwenewaldweg-Leisacher Alm)	123
Tabelle 43: Ergebnis Labelvektor Lösung Nr.3 (Zwenewaldweg-Leisacher Alm)	123
Tabelle 44: Ergebnis Labelvektor Lösung Nr.4 (Zwenewaldweg-Leisacher Alm)	124
Tabelle 45: Ergebnis Labelvektor Lösung Nr.1 (St. Jakob Schule-Weiße Spitze).....	126
Tabelle 46: Ergebnis Labelvektor Lösung Nr.2 (St. Jakob Schule-Weiße Spitze).....	128
Tabelle 47: Ergebnisse (Gehzeit & geometrische Kriterien) der optimalen Route Nr.1 (Brunnalm Talstation-Knoten Großer Leppleskofl)	132
Tabelle 48: Ergebnisse (Untergrund & POI`s) der optimalen Route Nr.1 (Brunnalm Talstation-Knoten Großer Leppleskofl)	133
Tabelle 49: Ergebnisse (Gehzeit & geometrische Kriterien) der optimalen Route Nr.2 (Brunnalm Talstation-Langschneid)	135
Tabelle 50: Ergebnisse (Untergrund & POI`s) der optimalen Route Nr.2 (Brunnalm Talstation-Langschneid).....	135
Tabelle 51: Knotennummern mit Bezeichnungen.....	139

Inhaltsverzeichnis

1	Einführung	1
1.1	Überblick	1
1.2	Motivation und Ziele	1
1.2.1	Ziele der Arbeit	1
1.2.2	Forschungsfragen	3
1.3	Bestehende Systeme	4
1.4	Struktur der Arbeit	4
2	Graphentheorie	6
2.1	Graphen	6
2.2	Graphenelemente	7
2.2.1	Wichtige Graphenelemente	7
2.2.2	Komplexe Graphenelemente	9
2.3	Eigenschaften von Graphen	9
2.3.1	Geometrie	15
2.4	Speicherung von Graphen	15
2.4.1	Adjazenzliste	16
2.4.2	Indizierte Adjazenzliste	16
3	Routenplanung	18
3.1	Shortest Path Problem (SPP)	19
3.1.1	Optimierung eines einzelnen Kriteriums	19
3.1.2	Arten von Lösungs-Methodiken	20
3.2	Dijkstra's Shortest Path Algorithmus	20
3.2.1	Ablauf des Dijkstra Algorithmus	21
3.2.2	Label-setting Algorithmus	26
3.2.3	Label-correcting Algorithmus	27
3.2.4	Dijkstra's Algorithmus mit Rangfolge	27
4	Multikriterielle Optimierung	28
4.1	Einführung	28
4.1.1	Terminologie	29
4.2	Effizienz und Dominanz	30
4.2.1	Pareto Optimalität	30

4.2.2	Dominanz.....	30
4.2.3	Strikte Dominanz	31
4.3	Definition von Optimalität.....	32
4.3.1	Skalarisierungsmethoden.....	32
4.3.2	Andere Methoden	33
4.4	Multikriterielle Shortest Path Probleme	34
4.4.1	Lösungen von multikriteriellen Shortest Path Problemen	35
4.5	Dijkstra`s Algorithmus bezüglich multikriterieller Optimierung	36
4.5.1	Erweiterter Dijkstra Algorithmus.....	36
5	Konzepte von multikriterieller Optimierung für die Routenplanung.....	45
5.1	Überblick	45
5.1.1	Strikte Pareto-optimale Wege.....	45
5.1.2	Komplexitäten der multikriteriellen Optimierung.....	45
5.2	Einzelne Pareto-optimale Lösungen.....	46
5.2.1	Optimierung mit lexikographischer Optimalität	46
5.2.2	Optimierung mit Hilfe der Weighted Sum Methode.....	47
5.3	Teilmenge von Pareto-optimalen Lösungen	47
5.3.1	Teilmenge durch Kombination von Einzellösungen	47
5.3.2	Teilmenge durch Dominanz mit Begrenzung	51
6	Netzgenerierung	53
6.1	Überblick	53
6.1.1	Wandergebiet Defereggental.....	53
6.2	Geographische Daten	53
6.2.1	Modellierung der realen Welt.....	54
6.2.2	Geographische Datenquellen	56
6.3	Datenbasis	57
6.3.1	Datenbasis Wandergebiet Defereggental	57
6.3.2	Datenbasis allgemein	58
6.4	Digitalisierung des Wandergebietes.....	58
6.4.1	Open Source Service – Geoland.at	58
6.4.2	Digitalisierungsvorgang	59
6.5	Koordinatentransformation	62
6.5.1	Geographische Koordinaten.....	62
6.5.2	Transformation WGS 84 – Gauß-Krüger.....	63
6.6	Genauigkeit der Daten	66

6.7	Bestimmung der Parameter für die Bewertung der Kanten	67
6.8	Erstellen der Datenstruktur als Ausgangsbasis für die Routenplanung	73
6.8.1	Indizierte Adjazenzliste.....	73
6.9	Visualisierung des Wandergebietes	76
6.9.1	Visualisierung von geographischen Daten	76
6.9.2	Übersichtskarte Wandernetz	77
6.9.3	Web GIS	80
7	Anwendung des erweiterten Dijkstra Algorithmus auf das Wandergebiet.....	83
7.1	Kriterien für die multikriterielle Optimierung	83
7.2	Methoden der Optimierung	85
7.2.1	Anwendung von Skalarisierungsmethoden auf den Algorithmus.....	85
7.2.2	Anwendung von Nicht-Skalarisierungsmethoden auf den Algorithmus.....	86
7.3	Anfälligkeiten des Algorithmus bezüglich der Kriterien	90
7.3.1	Anfälligkeiten des Algorithmus bezüglich der Kriterien für die Berechnung einer Einzellösung.....	90
7.3.2	Anfälligkeiten des Algorithmus bezüglich der Kriterien für die Berechnung einer Teilmenge aller Lösungen durch Dominanz mit veränderter lexikographischer Begrenzung	92
7.3.3	Anfälligkeiten des Algorithmus bezüglich der Kriterien für die Berechnung aller Pareto- optimalen Lösungen	93
7.3.4	Auswirkungen von gerundeten Kriterien auf die Ergebnisse des Algorithmus.....	94
7.3.5	Anfälligkeiten des Algorithmus bezüglich der Kriterien für mehr als zwei Kriterien	95
7.3.6	Anzahl an effizienten Lösungen.....	95
7.4	Statistik und Analyse	100
8	Fallbeispiele.....	106
8.1	Fallbeispiel 1	106
8.2	Fallbeispiel 2	112
8.3	Fallbeispiel 3	115
8.4	Fallbeispiel 4	125
8.5	Fallbeispiel 5	131
9	Anwendung	137
9.1	Software Anforderungen.....	137
9.2	Benutzereingabe	138
9.3	Benutzerausgabe.....	142
10	Zusammenfassung und Ausblick	147
	Literaturverzeichnis.....	149

1 Einführung

1.1 Überblick

Routenplanung ist ein Begriff, der heute weit verbreitet ist. Doch erst durch die technologischen Entwicklungen der letzten Jahrzehnte kann man heute davon sprechen, dass die Routenplanung ein Bestandteil unseres Lebens geworden ist. Die Idee dabei ist also, dass man wissen will, wie die optimale Route für verschiedenste Anwendungen und Kriterien aussieht. Es gibt oft hunderte oder tausende Möglichkeiten, wie eine Route je nach Voraussetzungen und Bedürfnissen verläuft. Deshalb kann die Problemstellung, die optimale Route zu finden, nur mit Hilfe von Algorithmen gelöst werden.

Die Diskussion über Algorithmen, den Datenbestand, die Einstellungskriterien und der jeweilige Zweck der Routenplanung spielen eine Rolle und werden nachfolgend behandelt.

Wie bei allem im Leben, ist die Ausgangslage, die Datenbasis, entscheidend für den späteren Erfolg. Deshalb muss, wenn das Thema Routenplanung diskutiert wird, große Aufmerksamkeit auf die Ausgangsdaten und das vorhandene Netzwerk geworfen werden.

1.2 Motivation und Ziele

Routenplanung gibt es mittlerweile für verschiedenste Anwendungen, sei es die Routenplanung für Verkehrsmittel, für Roboter oder für Fußgänger. Ziel dieser Diplomarbeit ist es, eine multikriterielle Routenplanung für Fußgänger auf Wanderwegen in Osttirol zu entwickeln. Als Datenbasis dienen geometrisch definierte Knoten entlang verschiedener Wanderwege in der Tourismusregion Osttirol. Ausgehend davon sollen Wegenetze erstellt werden, wobei ein wichtiger Aspekt die Bewertung der Knoten-Kanten Struktur darstellt. Es gibt verschiedene Algorithmen, die die Routenplanung je nach Problemstellung und Optimierung einsetzen, sei es der Dijkstra-Algorithmus oder verschiedenste heuristische Ansätze, um Aufgabenstellungen, wie zum Beispiel das SPP (Shortest Path Problem) zu lösen. Allerdings ist dies in den meisten Fällen eine „single-criterion“ Optimierung.

1.2.1 Ziele der Arbeit

Ziel dieser Arbeit ist es, mehrere Kriterien in die Routenplanung einfließen zu lassen und verschiedene Wege miteinander zu vergleichen. Es soll die Möglichkeit bestehen, vordefinierte Einstellungskriterien auszuwählen, um so benutzerspezifisch den geeignetsten

Wanderweg zu finden. Kriterien, die auf Wanderwegen eine besondere Rolle spielen, sind zum Beispiel die Distanz, die Steigung, die Untergrundbeschaffenheit und die Höhe in der man sich bewegt, um einige zu nennen.

Es soll herausgefunden werden, für welche Kriterien der Algorithmus bei dieser Anwendung in einem Wandernetzwerk besonders anfällig ist, beziehungsweise welche Kriterien sinnvolle Lösungen liefern. Darüber hinaus ist die Entwicklung des Algorithmus von entscheidender Bedeutung, was auch die verschiedensten Methoden der Optimierung inkludiert.

Als Ergebnis soll ein Anwendungsprogramm entstehen, das eine benutzerspezifische Routenplanung für Wanderwege ermöglicht. Die optimalen Wanderrouten können somit vor der Begehung der Wege berechnet und dargestellt werden, wobei auch die genaue Bestimmung der Gehzeit einen wichtigen Aspekt darstellt.

Daraus ergeben sich drei Hauptbereiche dieser Arbeit:

- Netzgenerierung: Erstellen eines Wandernetzes mit Knoten- und Kantenstruktur
- Umsetzung eines Algorithmus für die multikriterielle Optimierung
- Untersuchung des Algorithmus bezüglich der verwendeten Kriterien, deren Einfluss und der erhaltenen Ergebnisse

Netzgenerierung

Die Routenplanung basiert auf einer geographischen Datenbasis, die sich in Form einer Knoten- und Kantenstruktur als Wegenetz darstellt, indem die Berechnung der optimalen Route durchgeführt werden kann. Diese Netzgenerierung erfolgt durch Verwendung vorhandener Daten und aus selbst generierten Daten, die durch einen Digitalisierungsvorgang generiert werden. Wie bei jedem Datensatz muss auf das Format, die Vollständigkeit, deren Plausibilität, die Struktur, die Konsistenz und die Sinnhaftigkeit der Daten geachtet werden.

Umsetzung eines Algorithmus für die multikriterielle Optimierung

Die Einführung mehrerer Kriterien führt zu einem MSPP (Multikriterielles Shortest Path Problem) (Pressl, 2012). Solch eine Optimierung kann mit einem herkömmlichen SPP nicht gelöst werden. Deshalb muss ein Algorithmus für ein SPP erweitert werden, um ein MSPP zu bewältigen. Das Hauptaugenmerk dieser Arbeit liegt auf dem erweiterten Algorithmus, den verschiedenen Methoden und den Optimierungsvarianten. Im praktischen Teil dieser Arbeit wird ein Algorithmus mit multikriterieller Optimierung entwickelt und umgesetzt, der von Benutzern/Benutzerinnen genutzt werden kann.

Untersuchung des Algorithmus

Entscheidend für den Erfolg eines Routenplanungsprogrammes ist die Sinnhaftigkeit und Nachvollziehbarkeit der erhaltenen Ergebnisse. Dies soll dadurch gewährleistet werden, dass die Anfälligkeit des Algorithmus für verschiedene Kriterien untersucht wird. Nicht jedes Kriterium ist geeignet um für den Benutzer/die Benutzerin (Wanderer/Wanderin) eine optimale Wanderroute zu berechnen. Verschiedenste Varianten bezüglich der Kriterienauswahl, der Kriterienreihenfolge und den Optimierungsmethoden werden untersucht und diskutiert. Darüber hinaus wird die Benutzereingabe und Benutzerausgabe sowie die Visualisierung thematisiert.

1.2.2 Forschungsfragen

Aus den zuvor beschriebenen Hauptbereichen ergeben sich verschiedene Forschungsfragen.

Netzgenerierung

- Wie sehen die vorhandenen Daten, die vom Tourismusverband Osttirol bereitgestellt wurden, aus?
- Welche Genauigkeit weisen die Daten auf?
- Sind die Daten vollständig und konsistent?
- Auf welches Koordinatensystem beziehen sich die Daten?
- Wie können neue Daten generiert werden und wie lassen sich diese Daten dann mit den bereits vorhandenen Daten kombinieren?
- Müssen die Daten transformiert werden?
- Wie kann ein vernünftiges Netzwerk, das den Ansprüchen der Routenplanung genügt, erstellt werden?

Umsetzung eines Algorithmus für die multikriterielle Optimierung

- Welcher Algorithmus kann für eine Routenplanung verwendet werden?
- Wie wird der Algorithmus erweitert, um eine multikriterielle Optimierung zu ermöglichen?
- Welche Optimierungsmethoden gibt es?
- Wie erfolgt die Umsetzung?
- Welche Kriterien fließen ein?
- Wie können die Ergebnisse am besten dargestellt werden?

Untersuchung des Algorithmus

- Welche Kriterien sind besonders anfällig für den Algorithmus?
- Wie können die Kriterien kombiniert werden?
- Welche Optimierungsvarianten liefern welche Ergebnisse?

- Wie können die Kriterien verändert werden, um verschiedene Ergebnisse zu erhalten?
- Wie sollte die Benutzereingabe aussehen?
- Ist der Algorithmus für die spezielle Anwendung eines Wandergebietes geeignet?

1.3 Bestehende Systeme

Es gibt bereits verschiedenste Anwendungen von Routenplanung im Zusammenhang mit multikriterieller Optimierung. Viele Anwendungen basieren zum Beispiel auf der Routenplanung für Fahrzeuge in einem Straßennetz und für Transportdienstleistungen. Es gibt auch Anwendungen für Fußgänger in Städten und spezielle Anwendungen, wie eine Routenplanung für Leute mit speziellen Bedürfnissen (Pressl, 2012). Eine Routenplanung mit multikriterieller Optimierung für Wanderer/Wanderinnen ist somit eine spezielle Anwendung, die noch nicht in diesem Ausmaß untersucht ist, wie es in vielen anderen Bereichen der Fall ist. In dieser Diplomarbeit wird die Routenplanung in einem neu entwickelten Wegenetz im Defereggental (in Osttirol) getestet und diskutiert.

1.4 Struktur der Arbeit

Die Arbeit besteht zum einen aus einem theoretischen Teil, der den Hintergrund der Routenplanung beleuchtet und zum anderen aus einem praktischen Teil, indem ein Projekt umgesetzt wird. Die ersten fünf Kapitel beschäftigen sich mit dem theoretischen Hintergrund, ab dem sechsten Kapitel geht es um die Umsetzung dieses Projektes.

Das **Kapitel 2** beschäftigt sich mit der Graphentheorie. Die Graphentheorie dient als Basis der Routenplanung. Es werden Graphenelemente und Grapheneigenschaften sowie deren Relevanz auf die Routenplanung beschrieben.

Das **Kapitel 3** beschäftigt sich mit der Routenplanung per se. Es werden Algorithmen und im Speziellen die „single-criterion“ Routenplanung diskutiert. Darüber hinaus wird der klassische Dijkstra Algorithmus als Basis für das MSPP beschrieben.

Im **Kapitel 4** wird auf die multikriterielle Optimierung geschaut. Es werden Begriffe wie Pareto Optimierung und Dominanz erklärt. Verschiedenste Lösungsmöglichkeiten und Optimierungsmethoden ergeben sich dabei. Es wird eine multikriterielle Version eines DSPA (Dijkstra's Shortest Path Algorithmus) diskutiert und erklärt.

Das **Kapitel 5** handelt von Konzepten der Implementierung eines multikriteriellen Algorithmus. Basierend auf den Formulierungen im Kapitel 4 werden verschiedene Lösungsmöglichkeiten miteinander verglichen.

Im **Kapitel 6** geht es um die Netzgenerierung. Es wird beschrieben, wie die Knoten- und Kantenstruktur entsteht, die als Basis für die Routenplanung dient. Dabei gilt es zu beachten, dass ein Teil der Ausgangsdaten bereits vorliegt, doch die restlichen Daten erst durch eine Digitalisierung generiert werden. Darüber hinaus wird auch das Thema Visualisierung behandelt.

Im **Kapitel 7** wird die Anwendung des erstellten Algorithmus auf das Wandergebiet untersucht, wobei es vor allem darum geht, die verschiedenen Kriterien und deren Einfluss auf den Algorithmus zu untersuchen. Es werden verschiedene Optimierungsmethoden behandelt und Anfälligkeiten des Algorithmus erläutert.

Im **Kapitel 8** werden verschiedene Fallbeispiele beleuchtet, um den Algorithmus, die Einstellungskriterien und die erhaltenen Ergebnisse verständlicher zu präsentieren.

Das **Kapitel 9** beschäftigt sich mit den Software Anforderungen, der Benutzereingabe sowie der Benutzerausgabe.

Im **Kapitel 10** wird die Arbeit zusammengefasst und es wird ein Resümee gezogen.

2 Graphentheorie

Die Graphentheorie beschäftigt sich mit der Definition und den Eigenschaften von Graphen, topologischen Fragen und den daraus folgenden Algorithmen, um Problemstellungen zu lösen. Eine klassische Problemstellung ist das SPP (Shortest Path Problem), aber es gibt noch zahlreiche andere (Pressl, 2012). Es werden in diesem Kapitel einige grundlegende Definitionen und Eigenschaften beschrieben. Darüber hinaus wird auf die Speicherungsmöglichkeiten von Graphen eingegangen.

2.1 Graphen

Wegenetze für die Routenoptimierung werden als Graphen dargestellt, welche als Basis für jeden Routenplanungsalgorithmus dienen (Pressl, 2012). Die Graphentheorie wurde vom Schweizer Mathematiker und Physiker Leonhard Euler eingeführt. Er wollte das bekannte Problem „The Seven Bridges of Königsberg“ lösen (siehe Abbildung 1). Die Aufgabe war es eine geschlossene Tour in der Stadt Königsberg zu finden, wobei jede der sieben Brücken nur einmal passiert werden sollte. Euler hat das Land und die Brücken als Graph modelliert und bewies, dass es keine Lösung gibt, da von jedem Knoten eine ungerade Zahl an Kanten abgeht (siehe Abbildung 2).

Als Basisdefinition eines Graphen gilt:

$$G = G(V, E)$$

V ... Set der Knoten

E ...Set der Kanten

Ein Graph besteht also aus Knoten (V) und Kanten (E), die eine topologische Beziehung haben (Wieser, 2013/14).

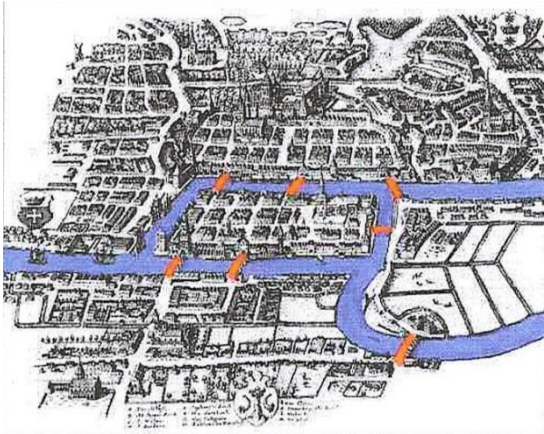


Abbildung 1: Brücken von Königsberg¹

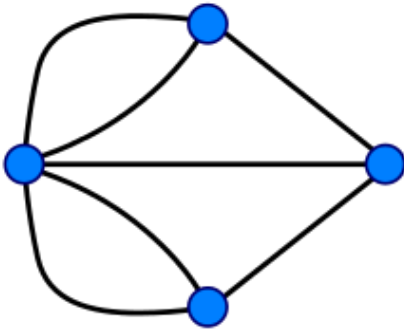


Abbildung 2: Graph von Brücken von Königsberg²

2.2 Graphenelemente

2.2.1 Wichtige Graphenelemente

Knoten und Kanten

Ein Graph $G = G(V, E)$ besteht aus Knoten $V = \{v_1, v_2, \dots, v_n\}$ und Kanten $E = \{e_1, e_2, \dots, e_n\}$, wobei eine Kante eine Verbindung zwischen zwei Knoten darstellt. Knoten und Kanten werden in der Abbildung 3 dargestellt.

¹ Bildquelle: <http://www.friderizianer.de/wordpress/wp-content/uploads/2011/10/bruecke-1.jpg>

² Bildquelle:

https://upload.wikimedia.org/wikipedia/commons/thumb/9/96/K%C3%B6nigsberg_graph.svg/220px-K%C3%B6nigsberg_graph.svg.png

Polygonpunkte und isolierte Punkte

Weitere Graphenelemente können auch Polygonpunkte sein, um eine Kurve zwischen zwei Knoten zu simulieren. Isolierte Punkte können Objekte außerhalb des Graphen beschreiben (z.B. Points of Interest). Siehe Abbildung 3.

Bögen

Eine Kante e kann durch eine gerichtete Kante ersetzt werden, um eine Richtung auszudrücken. Gerichtete Kanten werden als Bögen bezeichnet. Zwei gegensätzlich gerichtete Kanten sind konsistent mit einer normalen Kante (Hoffmann-Wellenhof, Legat, & Wieser, 2003, S. 85).

Traversen

Traversen werden für komplexe Kanten-Kanten Relationen verwendet. Eine Traverse verbindet 2 Kanten oder 3 Knoten miteinander. Eine Traverse beschreibt also eine Verbindung von einem Knoten v_i über v_j zu v_k . Somit können semantische Informationen (Topologie) dargestellt werden. Abbiegesituationen bei Kreuzungen können mit Hilfe von Traversen modelliert werden (Hoffmann-Wellenhof, Legat, & Wieser, 2003, S. 85).

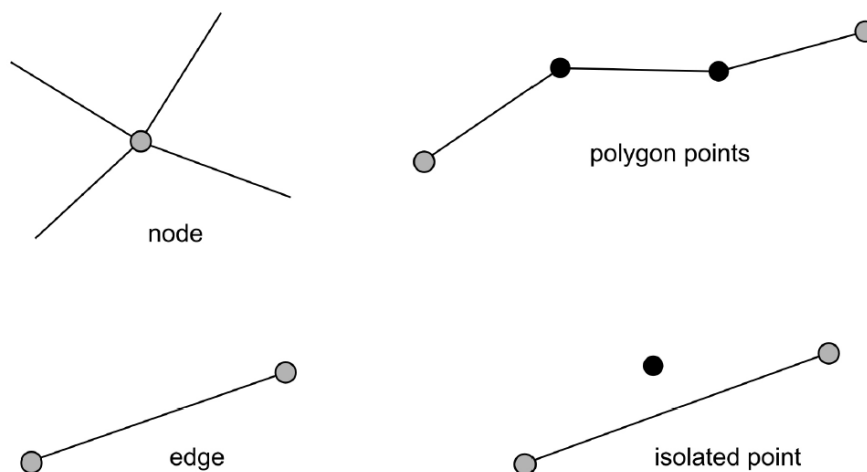


Abbildung 3: Wichtige Graphenelemente³

³ Bildquelle: (Pressl, 2012)

2.2.2 Komplexe Graphenelemente

Kette

Eine Kette ist eine Sequenz von Knoten und Kanten $(v_s, e_1, v_1, e_2, v_2, e_3, v_3, \dots, e_t, v_t)$ beginnend mit dem Knoten v_s und endend mit dem Knoten v_t .

Weg

Ein Weg ist eine Sequenz von Knoten und gerichteten Kanten $(v_s, a_1, v_1, a_2, v_2, a_3, v_3, \dots, a_t, v_t)$ beginnend mit dem Knoten v_s und endend mit dem Knoten v_t .

Kreis

Ein Kreis ist eine geschlossene Kette. Eine Kette ist eine einfache Kette, wenn sich alle Knoten unterscheiden außer $v_s = v_t$.

Zyklus

Ein geschlossener Weg wird Zyklus genannt. Wenn sich alle Knoten in einem Zyklus unterscheiden, außer $v_s = v_t$, dann nennt man ihn einen einfachen Zyklus.

2.3 Eigenschaften von Graphen

Ein Graph muss bestimmte Kriterien erfüllen, welche für die Routenplanung entscheidend sind (Hoffmann-Wellenhof, Legat, & Wieser, 2003, S. 84). Im Folgenden werden diese Eigenschaften diskutiert.

Grad des Knoten

Der Grad des Knoten (deg) ist die Anzahl der abgehenden oder eingehenden Kanten von einem bestimmten Knoten. Isolierte Punkte haben den Grad Null. Ein Knoten mit beispielsweise zwei eingehenden Kanten und einer abgehenden Kante hat den Grad $\text{deg} = 3$ (Pressl, 2012).

Finiter vs. infiniter Graph

Ein reales Navigationsnetz ist äquivalent mit einem finiten Graphen. Das bedeutet, dass ein Graph eine endliche Anzahl an Knoten und Kanten aufweist. In dieser Arbeit wird die Anzahl an Knoten mit n und die Anzahl der Kanten mit m angegeben. Die Anzahl n wird auch die Ordnung des Graphen genannt und die Anzahl m wird die Größe des Graphen $G(V, E)$ genannt.

Ein großes Netz wird üblicherweise in mehrere Subgraphen unterteilt. Ein Graph G_1 ist ein Subgraph von G , wenn G_1 komplett in G enthalten ist. Die Knoten und Kanten von G_1 sind Teilmengen von den Knoten und Kanten von G . Ein infiniter Graph besitzt unendlich viele Knoten und Kanten (Balakrishnan & Ranganathan, 1999, S. 14).

Einfacher vs. komplexer Graph

Ein einfacher Graph ist ein Graph ohne Schleifen (Kanten, bei denen der Startknoten gleich dem Endknoten ist) und ohne parallele Kanten (Kanten sind mit denselben Paaren von Knoten verbunden). In dieser Arbeit entsteht ein einfacher Graph.

Regelmäßiger vs. unregelmäßiger Graph

Ein regelmäßiger Graph ist ein Graph, indem alle Knoten den gleichen Grad (Anzahl eingehender oder abgehender Kanten) aufweisen. Reguläre Strukturen könnten angewendete Algorithmen vereinfachen, zum Beispiel mit Hilfe spektraler Methoden (Hoffmann-Wellenhof, Legat, & Wieser, 2003, S. 86). Normalerweise sind in realen Netzwerken irreguläre Strukturen zu finden.

Vollständiger Graph vs. unvollständiger Graph

Jedes beliebige Paar von Knoten ist durch eine Kante verbunden. Wenn man von n Knoten ausgeht, dann hat solch ein Graph zumindest $n(n - 1)/2$ Kanten. Üblicherweise sind reale Navigationsnetzwerke weit entfernt von diesem Szenario, jedoch gibt es spezielle Routing Probleme wie das TSP, denen ein kompletter Graph zugrunde liegt (Abbildung 4).

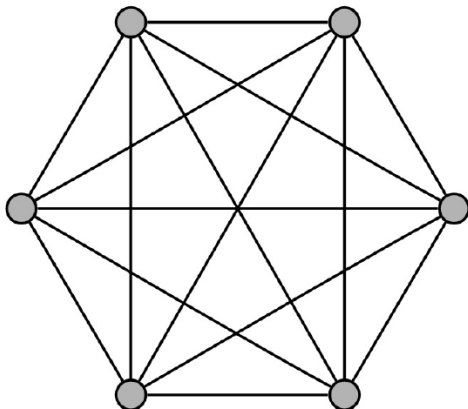


Abbildung 4: Vollständiger Graph⁴

⁴ Bildquelle: (Pressl, 2012)

Zusammenhängender vs. nicht zusammenhängender Graph

Ein Graph ist verbunden, wenn es zwischen jedem Paar von Knoten eine Kette oder einen Weg gibt. Ein Graph ist stark verbunden, wenn es einen Weg von v_s zu v_t und von v_t zu v_s für jedes Knotenpaar $\{v_s, v_t\}$ gibt (Hoffmann-Wellenhof, Legat, & Wieser, 2003, S. 87).

Planarer Graph vs. nicht planarer Graph

Ein planarer Graph kann in der Ebene dargestellt werden, wobei sich Kanten nur in Knoten schneiden. Für einen planaren Graph gilt die Ungleichung $m_e \leq 3n - 6$, wobei m_e die Anzahl der Kanten und n die Anzahl der Knoten darstellt. Ein Beweis dieser Euler-Gleichung existiert von Clark und Holton. Aufgrund der linearen Beziehung zwischen n und m_e , sind planare Graphen gut geeignet für Graphenberechnungen, bei denen die Komplexität des Algorithmus von n und m_e abhängt. Strikt planar sind zum Beispiel Netzwerke von Wasserwegen. Weiteres sind die meisten Straßennetzwerke planar, mit Ausnahme spezieller Szenarien, wie komplexe Autobahnauffahrten oder Brücken, die über eine andere Straße verlaufen (Hoffmann-Wellenhof, Legat, & Wieser, 2003, S. 87). Siehe Abbildung 5.

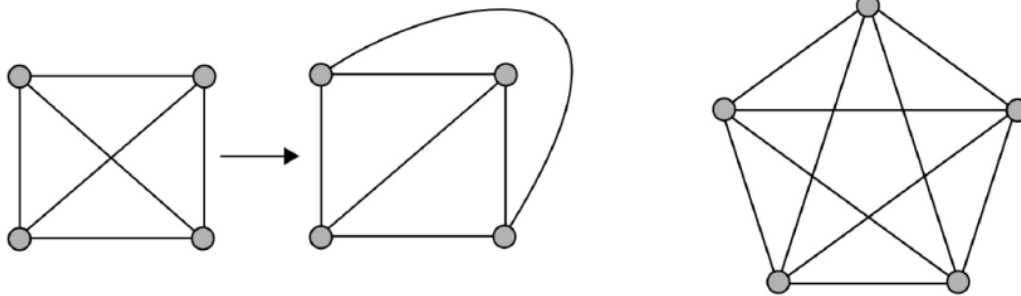


Abbildung 5: Planarer Graph (links) - nicht-planarer Graph (rechts) ⁵

Dünnere vs. dichtere Graph

Ein Graph mit $m_e \ll n(n - 1)/2$ wird als dünn bezeichnet. Planare Graphen sind ein gutes Beispiel. Ökonomische Speicherungstechniken können für solche Graphen angewendet werden (Hoffmann-Wellenhof, Legat, & Wieser, 2003, S. 87).

Gerichteter Graph

Bei gerichteten Graphen $G = G(V, A)$ existiert neben einem Set V der Knoten, auch ein Set $A = \{a_1, a_2, a_3, \dots, a_{m_a}\}$ mit m_a Bögen, anstatt einem Set E mit m_e Kanten. Gerichtete Graphen sind ein Standardtool, um Verkehrsnetzwerke zu modellieren. Auch in dieser Arbeit wird ein gerichteter Graph als Grundlage für die Anwendung des Algorithmus angenommen (Hoffmann-Wellenhof, Legat, & Wieser, 2003, S. 87). Siehe Abbildung 6.

⁵ Bildquelle: (Pressl, 2012)

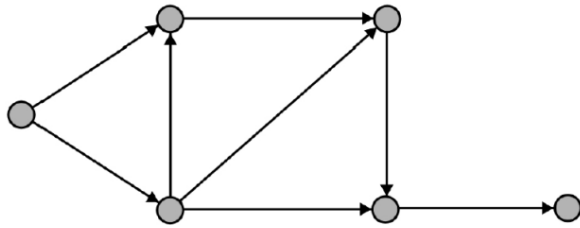


Abbildung 6: Gerichteter Graph⁶

Zyklisch vs. azyklischer Graph

Ein zyklischer Graph ist ein gerichteter Graph mit mindestens einem Zyklus. Eine Baumstruktur, bei der ein Ausgangsknoten und jeder andere Knoten durch genau einen Weg verbunden ist, ist ein azyklischer Graph (Hoffmann-Wellenhof, Legat, & Wieser, 2003, S. 87). Siehe Abbildung 7.

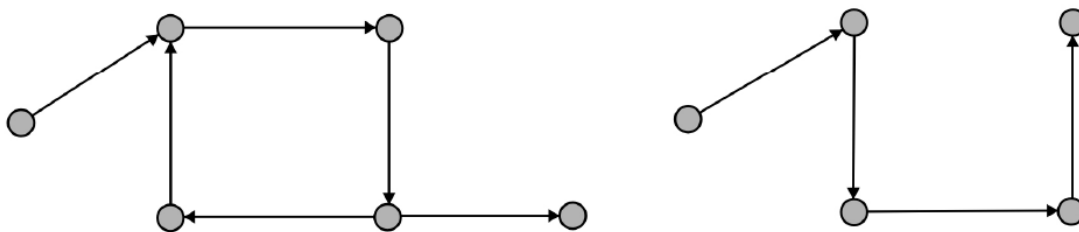


Abbildung 7: Zyklischer Graph (links) - azyklischer Graph (rechts)⁷

Bewerteter Graph

Ein bewerteter Graph ist ein gerichteter Graph $G = G(V, A, c)$, bei dem jeder Bogen eine Bewertung erhält. Ein bewerteter Graph wird auch als Netzwerk in der Graphentheorie beschrieben. Eine Kostenfunktion $c(a_{ij})$ weist jedem Bogen a_{ij} zwischen den Knoten v_i und v_j eine reale Bewertung c_{ij} zu. Diese Bewertungen (Label) können Kosten oder Gewichte repräsentieren. Kosten und Gewichte sind Werte, um die Schwierigkeiten der Durchfahrt eines Bogens auszudrücken (Hoffmann-Wellenhof, Legat, & Wieser, 2003, S. 87-88). Siehe Abbildung 8.

⁶ Bildquelle: (Pressl, 2012)

⁷ Bildquelle: (Pressl, 2012)

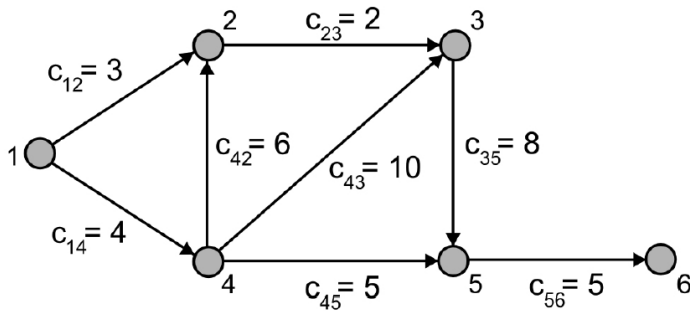


Abbildung 8: Bewerteter Graph⁸

Eine Bewertung wäre auch für ungerichtete Graphen möglich. Die Kosten c_{ij} bewerten einen Bogen in einem negativen Sinne, d.h. je höher die Kosten, desto schwieriger ist es den Bogen zu passieren. Im einfachsten Fall könnten die Kosten die geometrische Länge einer Strecke darstellen. Wird diese Länge mit einem bestimmten Widerstandsfaktor multipliziert, dann kann c_{ij} als monetäre Kosten oder als Transferzeit interpretiert werden, im Sinne einer thematischen Länge. Falls c_{ij} einen positiven Wert symbolisiert, kann nicht davon ausgegangen werden, dass die Kosten von v_i zu v_j plus v_j zu v_k größer als die Kosten von v_i zu v_k sind, da die geometrische Dreiecksungleichung $c_{ij} + c_{jk} > c_{ik}$ nicht gilt (Hoffmann-Wellenhof, Legat, & Wieser, 2003, S. 88). Siehe Abbildung 9.

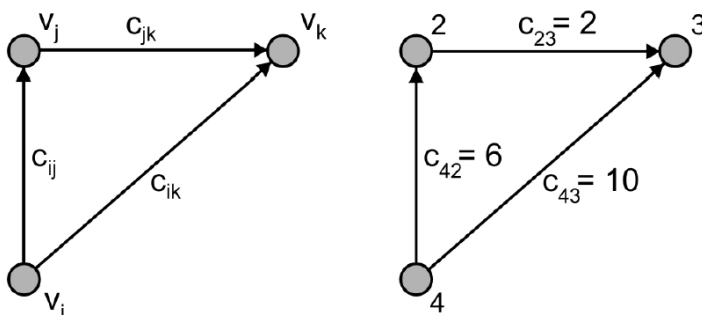


Abbildung 9: Dreiecksungleichung⁹

Im Gegensatz dazu, kann die Gewichtung p_{ij} eines Bogens im positiven Sinne gesehen werden, d.h. je höher die Gewichtung, desto einfacher ist es den Bogen zu passieren. Für positive Kosten $0 \leq c_{ij} \leq \infty$ lässt sich ein logarithmischer Zusammenhang darstellen

$$c_{ij} = -\log p_{ij}$$

wobei $1 \geq p_{ij} \geq 0$ angenommen wird. Eine hohe Gewichtung resultiert in niedrigen Kosten und umgekehrt.

⁸ Bildquelle: (Pressl, 2012)

⁹ Bildquelle: (Pressl, 2012)

Im Vergleich zur Kostenfunktion $c(a_{ij})$ für Bögen, gibt es analog dazu eine Kostenfunktion $b(t_{ijk})$, die für die Kosten b_{ijk} für Traversen t_{ijk} eingeführt wird. Der Kostenwert b_{ijk} entspricht hier den Kosten, die anfallen, wenn man von v_i über v_j nach v_k gelangt.

Es kann zum Beispiel das Umdrehen bei einer Wegkreuzung bewertet werden, falls das Umdrehen verboten ist, wird der Kostenwert auf unendlich gesetzt ($b_{ijk} = \infty$).

Das bedeutet, dass eine korrekte Bewertung von Graphenelementen, eine wichtige Rolle spielt, bezüglich des SPP (Shortest Path Problem). Des Weiteren stellt sich die Frage, wie man die Kosten c_{st} eines Weges vom Knoten v_s zum Knoten v_t erhält. Die einfachste Variante ist es, alle Kostenwerte der einzelnen Bögen entlang des Weges zu addieren. Betreffend der obengenannten Gleichung $c_{ij} = -\log p_{ij}$ und basierend auf logarithmischen Gesetzmäßigkeiten ist dies äquivalent mit dem Logarithmus des reziproken Produktes der Wahrscheinlichkeiten:

$$c_{st} = \sum_{path} c_{ij} = \log\left(\prod_{path} p_{ij}\right)^{-1}$$

Wenn bewertete Traversen Bestandteil des Weges sind, dann ändert sich die Gleichung in

$$c_{st} = \bar{b}_{ijk} \sum_{path} c_{ij} = \log\left(\prod_{path} p_{ij}\right)^{-\bar{b}_{ijk}}$$

, wobei \bar{b}_{ijk} die durchschnittlichen Kosten aller Traversen entlang des Weges darstellt (Hoffmann-Wellenhof, Legat, & Wieser, 2003, S. 88-89).

Statischer vs. dynamischer Graph

Ein statischer Graph besitzt zeitunabhängige Eigenschaften. Im Gegensatz dazu gibt es dynamische Graphen, deren Eigenschaften sich zeitlich verändern können. Dynamische Graphen modellieren beispielsweise Straßennetze, in denen Verkehrsbedingungen, wie Stau oder Baustellen, miteinbezogen werden (Hoffmann-Wellenhof, Legat, & Wieser, 2003, S. 89).

Deterministischer vs. stochastischer Graph

Bei deterministischen Graphen kennt man die Eigenschaften mit absoluter Gewissheit. Eigenschaften von stochastischen Graphen folgen einer Wahrscheinlichkeitsverteilungsfunktion. Bewertungen, die aus Informationen resultieren, von denen nicht mit absoluter Gewissheit davon ausgegangen werden kann, dass diese fehlerfrei vorliegen, wie zum Beispiel Informationen über das Wetter oder Verkehrsbedingungen, zählen zu stochastischen Größen (Hoffmann-Wellenhof, Legat, & Wieser, 2003, S. 89).

2.3.1 Geometrie

Die Geometrie unterteilt sich in Metrik und Topologie.

Metrik

Die Metrik definiert die Position der Graphenelemente innerhalb eines Koordinatensystems. Die Position der Graphenelemente wird durch Koordinaten dargestellt. Diese Positionen spielen für den eigentlichen Optimierungsprozess keine Rolle, sehr wohl aber für die Visualisierung und für SA (Spatial Analysis).

Topologie

Die Topologie beschreibt die Beziehungen von Objekten in einem Netzwerk. Sie definiert die Anordnung und Verbindung verschiedenster Elemente im Graphen. Die Topologie beschreibt also die Beziehung von Knoten und Kanten/Bögen eines Graphen.

2.4 Speicherung von Graphen

Es gibt verschiedene Speicherungstechniken, um Daten möglichst effizient zu speichern und um darauf zugreifen zu können. Man muss beachten, dass das Design von digitalen Karten mit Bezug auf das Modellieren von Transport-Netzwerken und die Repräsentation der Daten in der Speicherung am Computer zwei verschiedene Aspekte darstellen. Prinzipiell kann ein Graph und dessen Eigenschaften durch eine Matrix (zwei-dimensionales Array) oder durch mehrere Listen (ein-dimensionales Array) beschrieben werden. Im Folgenden werden die verschiedenen Speicherungstechniken beschrieben (Hoffmann-Wellenhof, Legat, & Wieser, 2003, S. 89).

Adjazenz Matrix

Die Adjazenz-Matrix \mathbf{A} ist eine $n \times n$ Matrix. Das Element A_{ij} der Zeile i und Spalte j hat folgende Bedeutung:

$$A_{ij} = \begin{cases} 1 & \text{wenn eine Kante } e_{ij} \text{ oder ein Bogen } a_{ij} \text{ existiert,} \\ 0 & \text{sonst.} \end{cases}$$

Im Falle eines ungerichteten Graphen ist die Matrix symmetrisch. Wenn die „1“ durch die Kosten c_{ij} eines Bogen ersetzt werden, dann spricht man von einer Kostenmatrix. Um mehrere Attribute eines Bogen zu speichern, muss die Matrix zu einem drei-dimensionalen Array erweitert werden (Hoffmann-Wellenhof, Legat, & Wieser, 2003, S. 89).

Indizierte Matrix

Die Indizierte-Matrix \mathbf{B} ist eine $n \times m_a$ Matrix. Diese Matrix wird meist für gerichtete Graphen eingesetzt und ist so strukturiert, dass jede Zeile p einen Knoten und jede Spalte q einen Bogen repräsentiert.

$$B_{pq} = \begin{cases} +1 & \text{wenn } v_p \text{ Startknoten von } a_q \text{ ist,} \\ -1 & \text{wenn } v_p \text{ Endknoten von } a_q \text{ ist,} \\ 0 & \text{wenn } a_q \text{ nicht inzident mit } v_p \text{ ist.} \end{cases}$$

Wie im Fall der Adjazenz-Matrix, kann auch hier die „1“ durch die Kosten eines Bogens oder eines anderen Attributes ausgetauscht werden. Im Vergleich zur Matrix \mathbf{A} zeichnet sich die Matrix \mathbf{B} durch Dünnheit aus. Wenn die Anzahl der Knoten und Kanten erhöht wird, erscheint der Fakt der unökonomischen Speicherung als stärker.

Die Adjazenz-Matrix und die Indizierte-Matrix sind daher nicht sehr geeignet um Transportnetzwerke zu repräsentieren (Hoffmann-Wellenhof, Legat, & Wieser, 2003, S. 90).

2.4.1 Adjazenzliste

Ein Graph kann mit Hilfe zweier Listen gespeichert werden: eine Knotenliste mit Knotenattributen und eine Bogenliste (Adjazenzliste), die die Verbindungen innerhalb des Graphen zwischen Knoten und Bögen repräsentiert. Für die Optimierung ist nur die Bogenliste von Bedeutung. Die Bogenliste enthält den Startknoten (Startknoten ID), den Endknoten (Endknoten ID) und die Kosten c_{ij} . Über Startknoten ID und Endknoten ID kann man auf die gewünschten Daten zugreifen (Pressl, 2012). Ein Beispiel einer Bogenliste eines gerichteten Graphen ist in Tabelle 1 dargestellt.

Tabelle 1: Adjazenzliste/Bogenliste

Bogen ID	Startknoten ID	Endknoten ID	c_{ij}
1	1	2	4
2	1	3	6
3	2	1	3
4	3	4	8
5	3	5	2

2.4.2 Indizierte Adjazenzliste

Ein eleganteres Verfahren stellt die indizierte Adjazenzliste dar. Ein Graph wird durch indizierte Listen repräsentiert. Als Basis dient dabei eine Knotenliste, die eine Knoten ID und die abgehenden Bögen enthält. Des Weiteren existiert eine Bogenliste, die die Bogen ID, den

Zielknoten und eine Bewertung enthält. Über die Information aus der Knotenliste, wie viele abgehende Bögen es pro Knoten gibt, kann ein korrekter und effizienter Zugriff auf die Kantenliste erfolgen (Pressl, 2012). Ein Beispiel ist in den nachfolgenden Tabellen dargestellt (Tabellen 2 und 3).

Tabelle 2: Indizierte Adjazenzliste / Knotenliste

Knoten ID	1	2	3	4	5
Index	1	4	6	6	8
Abgehende Bögen	3	2	0	2	0

Tabelle 3: Indizierte Adjazenzliste / Bogenliste

Bogen ID	1	2	3	4	5	6	7
Zielknoten	2	3	4	3	5	3	5
Kosten	4	13	3	11	3	8	6

3 Routenplanung

Definitionen und generelle Bezeichnungen

Das mathematische Konzept von Graphen und in spezieller Hinsicht von gerichteten Graphen wird verwendet, um Navigationsnetzwerke zu modellieren und simulieren. Ein gerichteter Graph $G(V, A)$ besteht aus einem Set V , bestehend aus n Knoten und einem Set A , bestehend aus m_a gerichteten Kanten, die auch Bögen genannt werden. Ein Knoten repräsentiert einen Kreuzungspunkt oder einen „dead-end“ Punkt. Ein Bogen beschreibt eine Richtung entlang einer Kante ausgehend von einem Startknoten zu einem Endknoten, zum Beispiel kann dies ein Bogen zwischen zwei Kreuzungspunkte oder ein Bogen zwischen einem Kreuzungspunkt und einem „dead-end“ Punkt sein. Zwei Knoten, die durch zumindest eine Kante miteinander verbunden sind, werden als adjazent bezeichnet. Wenn eine Verbindung von adjazenten Knoten in beide Richtungen vorhanden ist, benötigt es zwei Bögen, welche als inzident mit beiden Knoten bezeichnet werden. Darüber hinaus muss der gerichtete Graph bewertet werden, das bedeutet, dass ein realer Wert jedem Bogen zugeordnet wird. Ein bewerteter Graph wird häufig als Netzwerk bezeichnet und als $G(V, A, c)$ definiert, wobei eine Kostenfunktion $c(a_{ij})$ eine reale Bewertung c_{ij} einem individuellen Bogen a_{ij} vom Knoten v_i zum Knoten v_j zuweist. Die Variable c_{ij} wird als Kostennummer bezeichnet. Die Kostennummer beschreibt im Grunde wie schwierig es ist, einen Bogen von einem Knoten zum nächsten zu passieren und kann im Sinne von finanziellen Kosten, aber auch bezüglich einer geometrischen Länge, einer Transferzeit oder vielem mehr, betrachtet werden. Normalerweise können Kosten im Positiven wie im Negativen entstehen, allerdings werden meistens nur positive Kosten verwendet, um zusätzliche Probleme mit SP (Shortest Path)-Algorithmen zu vermeiden.

Ein Pfad zwischen einem Startknoten v_s und einem Zielknoten v_t ist als Sequenz von adjazenten Bögen definiert. Alle Bögen, die zwischen v_s und v_t durchlaufen werden, sind individuell einzigartig, d.h. sie unterscheiden sich voneinander. Ein Pfad wird als einfacher Pfad bezeichnet, wenn alle Knoten individuell einzigartig sind und er wird als Zyklus bezeichnet, wenn der Pfad geschlossen ist, d.h. $v_s = v_t$. Um die Kosten eines Pfades zu erhalten, werden die einzelnen Kosten der Bögen aufsummiert (Hoffmann-Wellenhof, Legat, & Wieser, 2003, S. 300).

3.1 Shortest Path Problem (SPP)

Das SPP (Shortest Path Problem) ist ein grundlegendes Problem in der Routenplanung und Routenoptimierung. Die Methodik, um das SPP zu lösen, bildet die Basis für jeden Routenplanungsalgorithmus. Die multikriterielle Routenplanung, die in dieser Arbeit behandelt wird, basiert auf dem SPP. Dabei geht es darum, den Pfad mit minimalen Kosten zu finden. In vielen Fällen wird der Pfad mit kürzestem Weg gesucht, aber häufig definieren auch andere Kriterien wie die Zeit, Kosten (Preis), Sicherheit, usw. die „beste“ Route (Jungnickel, 2002, S. 63). Alle Algorithmen, die in dieser Arbeit beschrieben werden, beziehen sich auf gerichtete Graphen. Im weiteren Verlauf werden die Bezeichnungen vereinfacht und somit werden die Elemente der gerichteten Graphen nur mehr mit Knoten und Kanten beschrieben.

Klassisches SPP

Das klassische SPP wird bei vielen verschiedenen Anwendungen genutzt und beschäftigt sich mit der Optimierung eines einzelnen Kriteriums. Das Problem kann in verschiedene Fälle eingeteilt werden: Es gibt „single-source/single-destination“ Probleme aber auch „single-source/multi-destination“ Probleme oder es geht darum, einen optimalen Pfad von jedem der mehrfach definierten Startknoten zu den mehrfach definierten Endknoten zu finden (Pressl, 2012).

3.1.1 Optimierung eines einzelnen Kriteriums

Die Optimierung eines einzelnen Kriteriums ist eine häufig angewendete Standard-Optimierung und das einfachste Problem von SP-Berechnungen. Für jede Kante k_{ij} vom Knoten v_i zum Knoten v_j existiert ein Wert c_{ij} . Dies könnte wie erwähnt Länge, Zeit oder irgendein anderes Kriterium, mit Ausnahme von negativen Werten, sein. Für die Berechnung wird ein Kostenlabel l_i für jeden Knoten $i = 1 \dots n$ gesetzt und während des Algorithmus upgedatet, indem die Pfadkosten vom Startknoten v_s zum i^{ten} Knoten gespeichert werden. Es müssen auch die Vorgängerknoten p_i jedes Knoten gespeichert werden, um den berechneten Pfad zurückverfolgen zu können (Pressl, 2012). Im Folgenden werden „single-source“ Fälle mit positiven Kosten betrachtet. Im weiteren Verlauf dieser Arbeit wird gezeigt, wie multikriterielle Routenplanung durchgeführt werden kann, indem zusätzliche Kriterien in einen erweiterten Algorithmus einfließen.

3.1.2 Arten von Lösungs-Methodiken

Exakte Algorithmen

Mit Hilfe von exakten Algorithmen können exakte und eindeutige Lösungen produziert werden, die den Vorteil haben, dass sie auch wirklich die optimale Lösung sind. Das Problem bei exakten Algorithmen besteht darin, dass für große Datenmengen große Herausforderungen an die Berechnungen gestellt werden, die möglicherweise nicht in angemessener Zeit gelöst werden können. Für „real-time“ Anwendungen kann das ein wichtiger Faktor sein, um auf einen Approximationsalgorithmus zurückzugreifen.

Heuristik

Heuristische Ansätze werden verwendet, um das Suchgebiet zu verringern, wobei ein kleinerer Graph und eine daraus resultierende schnellere Berechnung zur Folge sind. Der größte Nachteil bei dieser Methode ist, dass womöglich die optimale Lösung nicht gefunden werden kann. Ein bekannter Algorithmus im Zusammenhang mit heuristischen Ansätzen ist der A^* -Algorithmus (Hoffmann-Wellenhof, Legat, & Wieser, 2003, S. 315-318). Dieser Algorithmus verwendet einen heuristischen Wert, der zum Kostenwert jeder Kante hinzugefügt wird. Ein weiterer verbreiteter Ansatz ist der „straight-line distance“ Ansatz, welcher sich durch eine zielgerichtete Suche auszeichnet. Bei dieser Umsetzung wird der Kostenwert der Kante durch die Entfernung der Kante zum Ziel beeinflusst. Es wird die „straight line distance“ zum Kostenwert der Kante addiert und somit wird dieser Wert für Kanten, die näher am Ziel liegen, geringer.

3.2 Dijkstra's Shortest Path Algorithmus

Ein weit verbreiteter Algorithmus bei der Bewältigung von SP-Problemstellungen ist der Dijkstra Algorithmus, benannt nach Edsger W. Dijkstra. Dieser Algorithmus berechnet mit Hilfe von positiven Kosten c_{ij} den kürzesten Weg von einem Knoten v_s zu allen anderen Knoten eines Netzwerkes $G(V, A, c)$. Dijkstra's Algorithmus ist ein Label-setting Algorithmus. Während der Berechnungen werden temporäre und permanente Label in den Knoten gesetzt. Die temporären Label werden in T und die permanenten Label in P gespeichert. Jedem Knoten von V werden temporäre Labelvektoren l_i in einem gerichteten Graphen $G(V, E)$ zugeordnet. Für „single source“ – „single destination“ Problemstellungen wird überprüft, ob der Zielknoten aus dem Set der temporären Label T entfernt wurde. Somit kann der Algorithmus abbrechen (Hoffmann-Wellenhof, Legat, & Wieser, 2003, S. 303).

3.2.1 Ablauf des Dijkstra Algorithmus

Initialisierung

Der Algorithmus startet im Knoten v_s . Dieser Startknoten wird im Set der temporären Label T gespeichert. Das Set der permanent gesetzten Knotenlabel P ist zu Beginn leer. Das Label l_s ist null und wird dem Startknoten v_s zugeordnet. Die Label l_i aller anderen Knoten werden mit unendlich definiert. Die Initialisierung ist wie folgt definiert (Hoffmann-Wellenhof, Legat, & Wieser, 2003, S. 304-307):

$$T = \{v_s\}; P = \{\}$$

$$l_s = 0; l_i = \infty; \forall i \neq s.$$

Algorithmus

Erster Schleifendurchgang: Der Startknoten v_s ist der einzige Eintrag in T und der Knoten mit dem minimalen Label. Dieser Knoten mit minimalem Label wird ausgewählt, aus T entfernt und dem Set der permanent gesetzten Label P zugeschrieben. Im nächsten Schritt werden die Distanzen zu den Nachbarknoten von v_s betrachtet und zum Label l_s addiert. Wenn dieser berechnete Wert kleiner ist als das existierende Label l_i im betreffenden Knoten, dann wird das alte Label ersetzt. Für den ersten Schritt ist dies immer der Fall, da zu Beginn jedes Knotenlabel mit ∞ definiert ist, mit Ausnahme vom Label l_s . Die betreffenden Nachbarknoten werden als „besucht“ markiert, indem sie in das Set T aufgenommen werden und die Vorgänger p werden als v_s gesetzt. Die Nachbarknoten von v_s gehören nun zu T .

Zweiter Schleifendurchgang: Es werden alle Einträge in T betrachtet und der Knoten mit dem kleinsten Label l_i ausgewählt, aus T entfernt und permanent gesetzt, indem er in P aufgenommen wird. Im nächsten Schritt werden die Distanzen zu den Nachbarknoten vom Knoten v_i bestimmt und zum Label hinzugefügt. Dieser Wert wird dann mit den bestehenden Labelwerten des betreffenden Knoten verglichen und wenn der neu bestimmte Wert kleiner ist als der bestehende Wert, dann wird das Label überschrieben. Die Vorgänger p der upgedateten Knoten werden als v_i gesetzt. Dann werden die betreffenden Knoten wieder als „besucht“ markiert, indem sie in das Set T aufgenommen werden, sofern sie nicht schon in T existieren.

Restliche Schleifendurchgänge: Der Algorithmus startet wieder mit demselben Prozedere wie beim zweiten Schleifendurchgang, solange bis die Beendigung erfolgt. Alle Knoten, die sich im Set P befinden, ändern ihr Label nicht mehr und bleiben in P bis zum Ende der Berechnungen. Der Algorithmus beendet die Iterationen und somit die Berechnung, wenn das Set T leer ist. Dann sind die kürzesten Wege von einem Knoten zu allen anderen Knoten gefunden. Bei „single-source/single-destination“ Problemstellungen wird der Algorithmus

beendet, sobald der Zielknoten v_t erreicht und dieser ins Set der permanent gesetzten Label P aufgenommen wurde (Pressl, 2012).

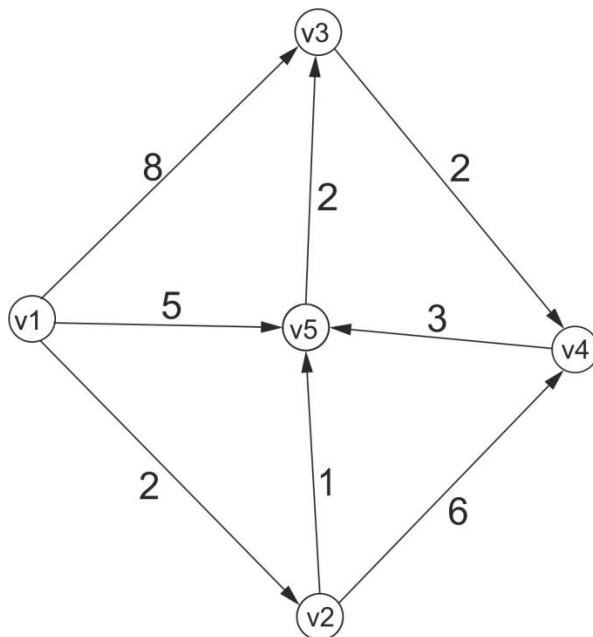


Abbildung 10: Beispielgraph mit Knotennummern und Kantenkosten

Betrachten wir nun ein praktisches Beispiel, um den Algorithmus anschaulicher darzustellen. Die Ausgangssituation ist in Abbildung 10 dargestellt. Die Graphik enthält einen Graph mit Knotennummern und Kostenwerte der Kanten. Abbildung 11 zeigt die Iterationsschritte des Dijkstra Algorithmus für den Graph in Abbildung 10 vom Startknoten v_1 zum Zielknoten v_4 . Die Knotennummern in Abbildung 10 werden nun durch die Knotenlabel des Algorithmus ersetzt.

Schritte des Algorithmus für das angeführte Beispiel

Initialisierung: Zuerst werden die Label initialisiert. Der Startknoten v_1 besitzt ein Initialisierungslabel von 0 und allen anderen Knoten wird das Label ∞ zugeschrieben.

Erster Schleifendurchgang: Startknoten ist der Knoten v_1 , er besitzt das Label 0 und ist zu Beginn der einzige Eintrag in T . Der Knoten mit minimalem Label wird gesucht, dies ist der Startknoten v_1 , er wird aus T gestrichen und permanent gesetzt, indem er in das Set $P = \{v_1\}$ aufgenommen wird. Im nächsten Schritt werden die Nachbarknoten betrachtet und die Distanzen zu diesen Knoten zum Label hinzugefügt. Dies beeinflusst die Knoten v_2 , v_3 und v_5 . Die Nachbarknoten werden als „besucht“ markiert und in das Set $T = \{v_2, v_3, v_5\}$ aufgenommen. Die Label dieser Knoten sind nun mit $l_2 = 2$, $l_3 = 8$ und $l_5 = 5$ definiert. Die Vorgänger dieser Knoten sind der Startknoten und wie folgt definiert:
 $p_2 = v_1$, $p_3 = v_1$, $p_5 = v_1$.

Zweiter Schleifendurchgang: Es wird wieder jener Knoten gesucht, der das minimale Label besitzt, ausgewählt, aus T gestrichen und in das Set P aufgenommen. Dies ist der Knoten v_2 mit dem Label $l_2 = 2$. Das Set P wird erweitert und enthält nun $P = \{v_1, v_2\}$. Dann werden wieder die Distanzen vom Knoten v_2 zu den Nachbarknoten betrachtet und diese dem Label hinzugefügt. Diese berechneten Werte werden dann mit den aktuellen Label in den Nachbarknoten verglichen und wenn der berechnete Wert kleiner ist als das aktuelle Label in einem Knoten, dann werden die alten Label überschrieben. Beim Knoten v_5 wird das vorhandene Label $l_5 = 5$ des ersten Schleifendurchganges durch $l_5 = 3$ ersetzt. Beim Knoten v_4 wird das Label auf $l_4 = 8$ gesetzt. Die betreffenden Knoten werden in T geschrieben, solange sie nicht schon in T vorkommen. Das Set T besteht nun aus $T = \{v_3, v_4, v_5\}$ mit den Labelwerten $l_3 = 8, l_4 = 8, l_5 = 3$ und den Vorgängern $p_3 = v_1, p_4 = v_2$ und $p_5 = v_2$.

Dritter Schleifendurchgang: Der nächste Knoten mit minimalem Label in T ist der Knoten v_5 mit $l_5 = 3$. Das Set P wird erweitert zu $P = \{v_1, v_2, v_5\}$. Dann werden wieder die Distanzen zu den Nachbarknoten zum Label hinzugefügt. Der neue Labelwert $l_3 = 5$, T enthält nun $T = \{v_3, v_4\}$ und die Vorgänger sind $p_3 = v_5$ und $p_4 = v_2$.

Vierter Schleifendurchgang: Der nächste Knoten mit minimalem Label aus T ist v_3 mit $l_3 = 5$. Das Set P wird erweitert und ist nun mit $P = \{v_1, v_2, v_3, v_5\}$ definiert. Anschließend werden wieder die Distanzen zu den Nachbarknoten betrachtet und zum Label hinzugefügt. Ein neues Label im Knoten v_4 mit $l_4 = 7$ entsteht, der Vorgänger ist $p_4 = v_3$ und ein einziger Eintrag in T bleibt mit $T = \{v_4\}$.

Fünfter Schleifendurchgang: Der einzig verbliebene Knoten im Set T wird ausgewählt, aus T gestrichen und in das Set P aufgenommen. Somit ergibt sich folgendes: $T = \{\}$, $P = \{v_1, v_2, v_3, v_4, v_5\}$. Da das Set T leer ist, beendet der Algorithmus die Berechnungen.

Um schlussendlich den kürzesten Weg vom Startknoten v_1 zum Zielknoten v_4 zu finden werden die einzelnen Stationen zurückverfolgt und somit erhält man folgenden Weg:
 $v_1 - v_2 - v_5 - v_3 - v_4$.

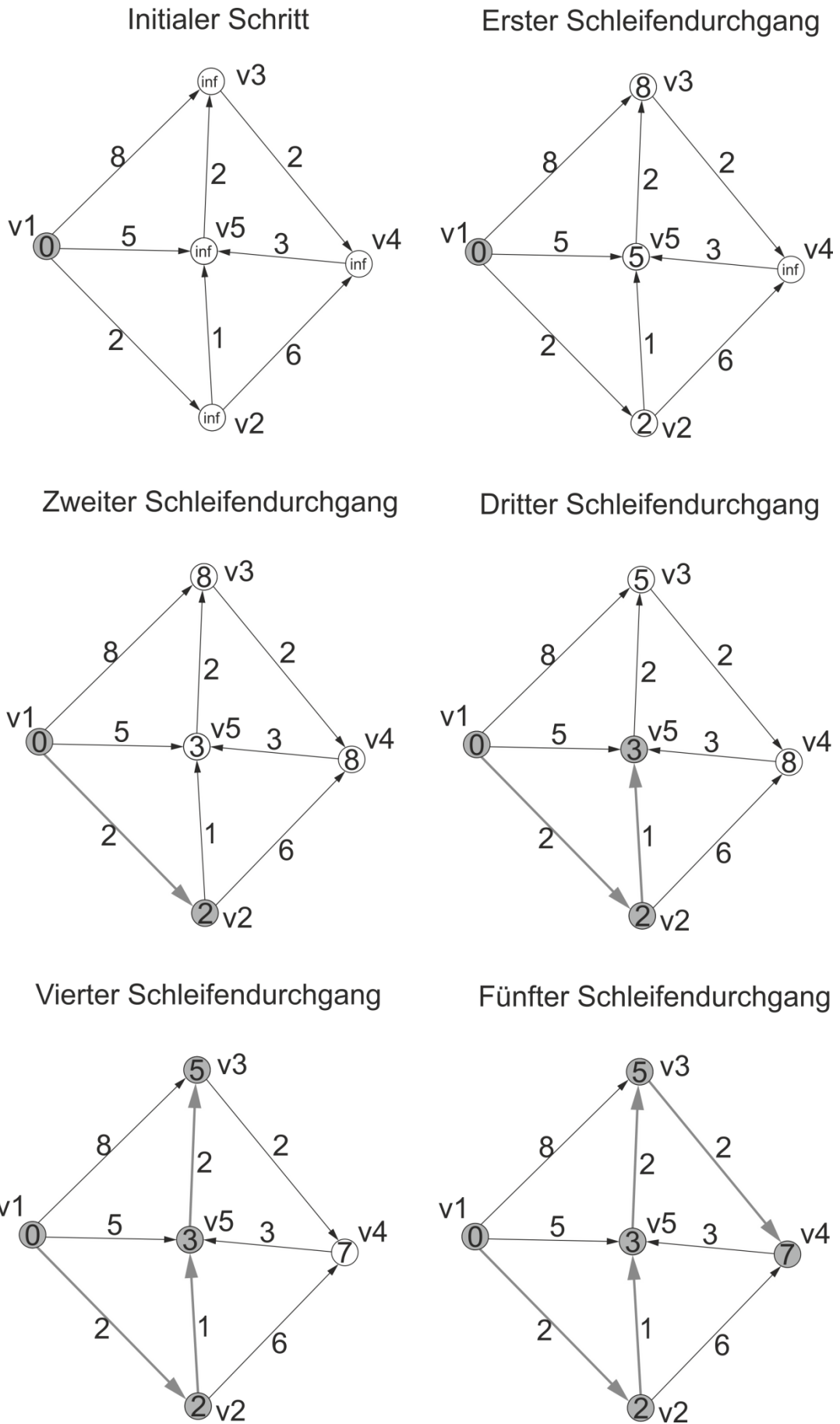


Abbildung 11: Prozedere des Dijkstra Algorithmus

Pseudocode

In Tabelle 4 werden die Symbole, die im Pseudocode verwendet werden, erläutert.

Tabelle 4: Pseudocode Symbole

Symbole	Erklärung
$:=$	Zuweisung zu einem Label oder Knoten
\leftarrow	Zuweisung zu einem Set
$\{x\}$	Set das aus dem Knoten x besteht
$\{\}$	Leeres Set
$X \setminus \{x\}$	Set X mit der Ausnahme vom Knoten x
$X + \{x\}$	Set X plus Knoten x
$X \leftarrow X \setminus \{x\}$	Knoten x wird aus Set X entfernt
$X \leftarrow X + \{x\}$	Knoten x wird dem Set X hinzugefügt
\in	Element von
\notin	Kein Element von
\wedge	Logisches „und“

Algorithmus 1. SP - Algorithmus basierend auf Dijkstra`s Ansatz (Hoffmann-Wellenhof, Legat, & Wieser, 2003, S. 304):

V ... Set aller Knoten

T ... Set aller Knoten mit einem temporären Label

P ... Set aller Knoten mit einem permanenten Label

N ... Set von adjazenten Knoten (Nachbarschaft)

v_s ... Startknoten

l_j ... Label eines Knoten v_j

p_j ... Vorgänger eines Knoten v_j

c_{ij} ... Kosten eines Bogen a_{ij}

1. $l_s := 0; T \leftarrow \{v_s\}; P \leftarrow \{\};$
2. FOR $v_j \in V \setminus \{v_s\}$ DO $l_j := \infty$ END FOR;
3. WHILE $T \neq \{\}$ DO
4. $v_i := v_j \in T$ mit min. l_j ;
5. $P \leftarrow P + \{v_i\}; T \leftarrow T \setminus \{v_i\}$
6. FOR $v_j \in N(v_i)$ DO
7. IF $(v_j \notin T) \wedge (v_j \notin P)$ THEN
8. $l_j := l_i + c_{ij}; p_j := v_i$;
9. $T \leftarrow T + \{v_j\}$ END IF;
10. IF $(v_j \in T) \wedge (l_i + c_{ij} < l_j)$ THEN
11. $l_j := l_i + c_{ij}; p_j := v_i$ END IF
12. END FOR
13. END WHILE

Diese Variante beschreibt den normalen Dijkstra Algorithmus, wobei der Algorithmus die kürzesten Wege von einem Startknoten zu allen anderen Knoten berechnet. Es gibt auch die Möglichkeit, am Beginn zu definieren, welcher Knoten der Zielknoten ist, um den Algorithmus, sobald dieser Zielknoten erreicht wurde, abubrechen. Dies erfolgt dadurch, dass eine weitere Bedingung am Beginn der Schleifendurchgänge gesetzt wird. Der Algorithmus bricht ab, sobald der Zielknoten permanent gesetzt wurde, d.h. in das Set P aufgenommen wurde.

Klassifikation von Algorithmen

Um SP - Algorithmen zu klassifizieren, werden einige spezielle Eigenschaften diskutiert und alternative Lösungsansätze betrachtet.

Breadth-first search vs. depth-first search strategies

Eine der Grundideen des Dijkstra Algorithmus ist, dass ein Suchbaum generiert wird, welcher iterativ wächst, indem ein Knoten nach dem anderen „gelabelt“ wird und so wird vom Startknoten ausgehend eine Art Baumstruktur gebildet. Wie es vom „breadth-first search“ Algorithmus bekannt ist, wird in einem idealen Fall der Suchbaum symmetrisch aufgebaut. Dies geschieht wenn alle Kanten gleiche Kosten aufweisen.

Im Gegensatz dazu gibt es den „depth-first search“ Algorithmus, bei dem sich der Suchbaum vornehmlich in eine bestimmte Richtung und zwar zum Zielpunkt ausbreitet (Hoffmann-Wellenhof, Legat, & Wieser, 2003, S. 307).

3.2.2 Label-setting Algorithmus

„Label-setting“ bedeutet, dass ein Knoten v_i , der aus dem Set der temporär gesetzten Knotenlabel entfernt wird, der Knoten mit minimalem Label ist. Beim Label-setting Algorithmus müssen die Kosten der Kanten positiv sein. Die Charakteristik des Label-setting Algorithmus ist, dass die Knoten maximal einmal ins Set T der temporären Knotenlabel aufgenommen werden und sobald sie aus T entfernt und in das Set P der permanent gesetzten Knotenlabel aufgenommen werden, erhalten Sie ihren permanenten Wert. Der entscheidende Punkt, der die Berechnungszeit prägt, ist wie das minimale Knotenlabel in T bei jeder Iteration gefunden wird (Pressl, 2012). Es werden verschiedene Ansätze bei den jeweiligen Algorithmen angewendet. Ein Beispiel für einen Label-setting Algorithmus ist der klassische Dijkstra Algorithmus, wie oben beschrieben.

3.2.3 Label-correcting Algorithmus

Der Dijkstra Algorithmus kann leicht zu einem Label-correcting Algorithmus erweitert werden. Im Vergleich zum Label-setting Algorithmus kann ein Knoten öfter T hinzugefügt und aus T entfernt werden, bis er seinen endgültigen Wert erhält. Ein Beispiel ist die Bellman-Ford Methode (Korte & Vygen, 2006, S. 159).

3.2.4 Dijkstra's Algorithmus mit Rangfolge

Den größten Einfluss auf die Berechnungszeit des Algorithmus hat der Vorgang der Knotenauswahl aus dem Set T . Anstatt alle Knoten zu scannen, kann die Berechnungszeit verkürzt werden, indem eine Rangfolge eingeführt wird, indem die Knoten gemäß ihrer Entfernung in einer Rangfolge gespeichert werden (Hoffmann-Wellenhof, Legat, & Wieser, 2003, S. 310). Beispiele für solche Vorgehensweisen sind „Binary Heap“, „Fibonacci Heap“ und „Dial's Implementation“ (Pressl, 2012).

4 Multikriterielle Optimierung

Basierend auf den konzeptionellen Grundlagen des SSPP (Single Criterion Shortest Path Problem), behandelt dieses Kapitel die multikriterielle Optimierung im Allgemeinen und die multikriterielle Optimierung für SP (Shortest Path)-Problemstellungen im Detail. Zu Beginn werden generelle Fragestellungen zum Thema Optimierung im Fall von mehreren Kriterien behandelt. Es werden Begriffe wie Effizienz und Pareto Dominanz erläutert, die als Basis dienen, um multikriterielle Problemstellungen zu diskutieren und um die beste Lösung innerhalb verschiedener Alternativen zu finden.

Das Ziel dieser Arbeit ist es für Wanderer/Wanderinnen, je nach Auswahl der verfügbaren Kriterien, die optimale Route im bestehenden Netz zu finden. Dabei stellen sich Fragen wie „Wie werden verschiedene Kriterien in den Berechnungsverlauf miteinbezogen, um eine optimale Route zu generieren?“ und „Wie kann mehr als eine effiziente Lösung für den Anwender/die Anwenderin bereitgestellt werden?“. Dieses Kapitel soll diese Fragen diskutieren und aufzeigen, welche Methoden es dabei gibt. In diesem Zusammenhang muss von einer „single-criterion“ Optimierung abgewichen werden, da diese Optimierung nur ein Kriterium im Optimierungsverlauf berücksichtigt. Um komplexere Aufgabenstellungen zu lösen und mehrere Kriterien miteinfließen zu lassen, muss auf eine multikriterielle Optimierung zurückgegriffen werden. Solche Problemstellungen werden als „Multi Criteria Shortest Path Probleme“ (MSPP) mit positiven Kantenlängen bezeichnet. Im Allgemeinen geht es dabei um den Begriff „Multiobjective Optimization“, den man in der Literatur wie in (Collette & Siarry, 2003, S. 20) häufig findet.

4.1 Einführung

Um den Begriff und die Thematik der multikriteriellen Optimierung anschaulicher darzustellen, wird nun ein einfaches Beispiel aus dem alltäglichen Leben betrachtet. Stellen wir uns vor, wir wollen ein neues Auto kaufen und es gibt vier Modelle zur Auswahl: einen VW Golf, einen Opel Astra, einen Ford Focus und einen Toyota Corolla. Wir bevorzugen ein billiges und PS-starkes Auto mit möglichst wenig Spritverbrauch. In diesem Fall haben wir ein Entscheidungsproblem mit vier Alternativen und drei Kriterien. Die Charakteristiken der vier Autos sind in Tabelle 5 aufgelistet. Dieses Beispiel ist (Ehrgott, 2005, S. 1-2) entnommen.

Tabelle 5: Kriterien und Alternativen des Autobei spiels

	Alternativen			
	VW	Opel	Ford	Toyota
Kriterien Preis (1000 Euro)	16.2	14.9	14.0	15.2
Verbrauch (l/100 km)	7.2	7.0	7.5	8.2
Power (kW)	66.0	62.0	55.0	71.0

Welches Auto ist nun die „beste“ Wahl, wenn es nicht das Auto gibt, das am billigsten und am stärksten ist und am wenigsten Verbrauch aufweist (Ehrgott, 2005, S. 2)? Eine Alternative muss gewählt werden, wobei der Kriterienkonflikt berücksichtigt werden muss.

4.1.1 Terminologie

$X = \{VW, Opel, Ford, Toyota\}$ wird das mögliche oder alternative Set genannt. Unter Berücksichtigung von zwei Kriterien werden diese definiert als f_1 (z.B. Preis) und f_2 (z.B. Verbrauch), wobei $f_1(x)$ und $f_2(x)$ als Kriterienfunktionen beschrieben werden. Das Optimierungsproblem kann wie folgt definiert werden:

$$\min_{x \in X} (f_1(x), f_2(x)), \text{ (Ehrgott, 2005, S. 2)}$$

Betrachtet man nun das Autobeiispiel mit den zwei Kriterien Preis und Spritverbrauch, dann ergeben sich folgende Werte für $f_1(x)$ und $f_2(x)$: $f_1(x) = \{16.2; 14.9; 14.0; 15.2\}$ und $f_2(x) = \{7.2; 7.0; 7.5; 8.2\}$. In Abbildung 12 werden die Kriterienwerte jeder Lösung dargestellt.

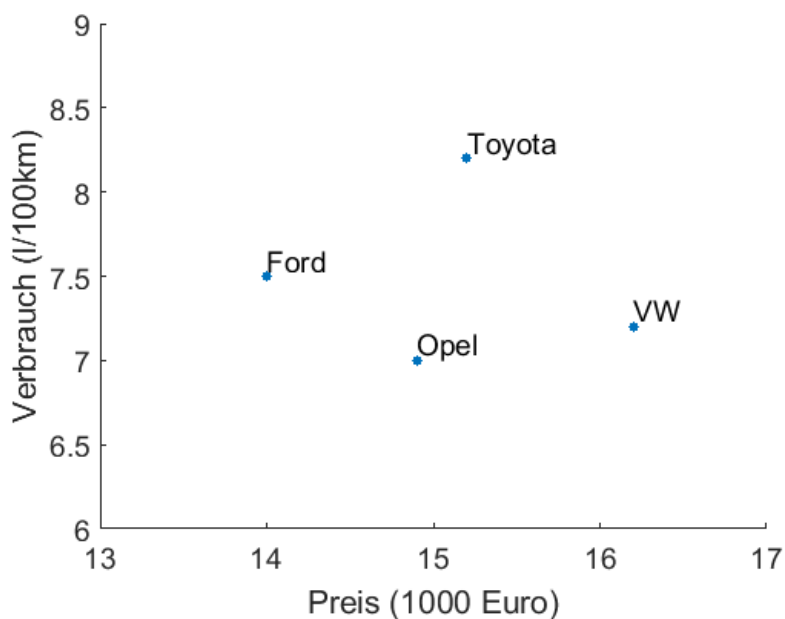


Abbildung 12: Kriterienwerte - Autobeiispiel

4.2 Effizienz und Dominanz

4.2.1 Pareto Optimalität

Wie bereits zuvor erwähnt wurde, gilt, dass durch die Anwendung der multikriteriellen Optimierung mehr als eine effiziente Lösung entsteht. In diesem Zusammenhang muss der italienische Ökonom und Soziologe *Vilfredo Pareto* (1848-1923) genannt werden. Er führte effiziente Lösungen im Sinne von Pareto Optima wie folgt ein (Tarapata, 2007):

Definition 1. „A solution is Pareto optimal if it cannot be improved upon from one criterion's perspective without making another criterion worse off“.

Die Herausforderung ist nun, diese effizienten Pareto Lösungen zu finden und die „besten“ auszuwählen (Pressl, 2012). Das Problem alle Pareto-optimalen Lösungen zu finden resultiert in einem „NP-hard“ („non-deterministic polynomial time“) Problem im Generellen. Solche Probleme können in polynomialer Zeit nicht gelöst werden. Bevor die Details zur Lösung von multikriteriellen Optimierungsproblemen diskutiert werden, müssen noch Konzepte wie Dominanz und Nicht-Dominanz erläutert werden. Durch die Einführung des Dominanz-Konzeptes wird der Suchraum, durch das Streichen ineffizienter Lösungen, reduziert (Ehrgott, 2005, S. 23), (Steuer, 1986, S. 3-4).

4.2.2 Dominanz

Es seien f_1, f_2, \dots, f_k eine Reihe von Kriterien, $f_1(x), f_2(x), \dots, f_k(x)$ die dazugehörigen Kriterienfunktionen und $f(x)$ die Mappingfunktion mit $f(x) = \{f_1(x), f_2(x), \dots, f_k(x)\}$. Die Definition der Dominanz kann wie folgt geschrieben werden:

Definition 2 (Ehrgott, 2005, S. 24). „A feasible solution $\hat{x} \in X$ is called efficient or Pareto optimal, if there is no other $x \in X$ such that $f(x) \leq f(\hat{x})$. If \hat{x} is efficient, $f(\hat{x})$ is called non-dominated. If $x^1, x^2 \in X$ and $f(x^1) \leq f(x^2)$ we say x^1 dominates x^2 and $f(x^1)$ dominates $f(x^2)$. The set of all efficient solutions $\hat{x} \in X$ is called efficient set or the set of all Pareto optimal solutions.“

Im Fall eines diskreten Optimierungsproblems mit zwei Kriterien ($k = 2$) ist $f(x)$ ein Set das zwei Kriterienfunktionen $f(x) = \{f_1(x), f_2(x)\}$ beinhaltet. Dann wird die Definition 1 beachtet:

Definition 3. „A solution is Pareto optimal if it cannot be improved upon from one criterion's perspective without making another criterion worse off“.

Bezüglich des Autobeispiels bedeutet das, wenn man Abbildung 12 betrachtet, dass $Ford = [14.0, 7.5]$ und $Opel = [14.9, 7.0]$ die beiden effizienten Lösungen sind.

Anhand dieses Autobeispiels soll nun Definition 3 näher erläutert werden. Bezüglich Opel, die Lösung kann nicht billiger werden ohne dass der Spritverbrauch ansteigt. Deshalb ist Opel eine Pareto-optimale Lösung. Betrachtet man Ford, so kann man sagen, dass der Spritverbrauch nicht weniger werden wird, ohne dass der Preis des Autos ansteigt. Deswegen ist Ford auch eine Pareto-optimale Lösung.

Im Falle von zwei Kriterien kann die Definition 2 wie folgt geschrieben werden:

Definition 4. „A feasible solution $\hat{x} \in X$ is called efficient or Pareto optimal, if there is no other $x \in X$ such that $f_1(x) \leq f_1(\hat{x})$ and $f_2(x) \leq f_2(\hat{x})$ “.

Bezüglich des Autobeispiels kann gesagt werden, dass es keine weitere Lösung gibt, die billiger ist und weniger Verbrauch aufweist als die Lösungen von Ford und Opel (Ehrgott, 2005, S. 24).

4.2.3 Strikte Dominanz

In Abbildung 13 werden Pareto-optimale Lösungen dargestellt. Die Lösungen x^3, x^4, x^5, x^6 und x^7 werden als strikte Pareto-optimale Lösungen bezeichnet (T'kindt & Billaut, 2002, S. 51). Die Definition für strikte Pareto Optima lautet wie folgt:

Definition 5. „A solution is strict Pareto optimal, if there is no other $x \in X$ such that $f(x) \leq f(\hat{x})$ with at least one strict inequality. If \hat{x} is efficient, $f(\hat{x})$ is called non-dominated. If $x^1, x^2 \in X$ and $f(x^1) \leq f(x^2)$ we say x^1 dominates x^2 and $f(x^1)$ dominates $f(x^2)$. The set $\hat{x} \in X$ is called the set of all strict Pareto optimal solutions.“

In der Literatur werden die Lösungen von Definition 2 auch manchmal als schwache Pareto-optimale Lösungen bezeichnet, wie auch in (T'kindt & Billaut, 2002, S. 51).

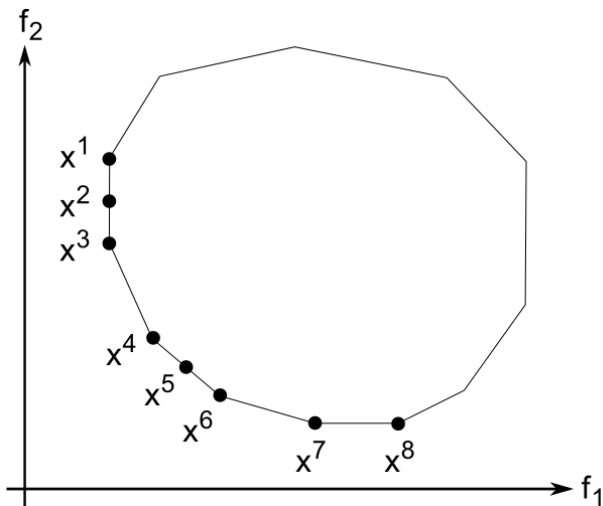


Abbildung 13: Pareto-optimale Lösungen¹⁰

4.3 Definition von Optimalität

Um multikriterielle Optimierungsprobleme in polynomischer Laufzeit zu lösen, müssen Approximationsmethoden angewendet werden. Approximation bedeutet in diesem Zusammenhang die Bestimmung einer Teilmenge aller Pareto-optimalen Lösungen, da alle Lösungen in polynomischer Laufzeit nicht ermittelt werden können (Ehrgott, 2005, S. 80). Folgend werden verschiedene Methoden aufgezeigt, die es ermöglichen die Anzahl an Lösungen zu reduzieren und effiziente Lösungen zu bestimmen.

4.3.1 Skalarisierungsmethoden

Skalarisierungs-Techniken sind Methoden, um multikriterielle Optimierungsprobleme in „single-criterion“ Optimierungsprobleme zu transformieren.

Weighted Sum Method

Die „weighted sum method“ (WSM), auch als „weighted sum scalarization“ bezeichnet, reduziert das multikriterielle Problem in ein „single-criterion“ Problem, indem positive Gewichte λ_i an k Kriterien f_i angebracht werden. Durch eine Aufsummierung der einzelnen Werte ergibt sich ein skalarer Wert, der minimiert wird. Diese Minimumsumme ist wie folgt definiert (Ehrgott, 2005, S. 65):

$$\min_{x \in X} \sum_{i=1}^k \lambda_i f_i(x)$$

¹⁰ Bildquelle: (Pressl, 2012)

Die Parameter $\lambda_1, \dots, \lambda_k$ drücken die unterschiedliche Wichtigkeit der einzelnen Kriterien aus. Das Optimierungsergebnis ist eine einzelne und schwache effiziente Lösung (Ehrgott, 2005, S. 68). Das bedeutet, dass unter Umständen andere und möglicherweise auch bessere Lösungen verloren gehen können. Um dieses Problem zu beseitigen, werden andere Skalarisierungsmethoden angewendet wie zum Beispiel die „ ϵ – Constraint Method“.

Im Kapitel 4.3.2 wird auf „Nicht-Skalarisierungsmethoden“ geschaut, um solche Probleme zu vermeiden, da dann alle Kriterien simultan und unterschiedlich behandelt werden.

Weitere Skalarisierungs-Methoden werden in (Ehrgott, 2005, S. 97) beschrieben, wie die schon erwähnte „ ϵ – Constraint Method“, die „Hybrid Method“, die „Elastic Constraint Method“ und einige andere.

4.3.2 Andere Methoden

Lexikographische Ordnung

Es wird eine hierarchische Ordnung eingeführt, indem man sagt, dass ein Kriterium das Hauptkriterium ist und alle anderen folgen in einer bestimmten Reihenfolge. Die Kriterienvektoren $f_1(x)$ und $f_2(x)$ können als Beispiel lexikographisch verglichen werden (Pressl, 2012). Um dies anschaulicher darzustellen, wird wieder das Autobispiel herangezogen. Die Reihenfolge der Kriterien könnte also so sein, dass der Preis wichtiger ist als der Spritverbrauch. In diesem Fall wird eine lexikographische Ordnung verwendet. Die Minimierung kann mathematisch wie folgt definiert werden (Ehrgott, 2005, S. 129):

$$\min_{x \in X} \text{lex}(f_1(x), f_2(x))$$

Definition 6. Das lexikographische Minimum für einen Zwei-Kriterien Fall ist wie folgt definiert:

$$(f_1, f_2) <_{\text{lex}} (f'_1, f'_2) \text{ if } (f_1 < f'_1) \text{ or } (f_1 = f'_1 \text{ and } f_2 < f'_2)$$

Diese Definition kann für mehr als zwei Kriterien leicht erweitert werden. Wenn man wieder an das Autobispiel von zuvor denkt, dann kann man sagen, dass $Ford = [14.0, 7.5]$ aus den Pareto-optimalen Lösungen ($Ford, Opel$) gewählt werden sollte. Aufgrund der lexikographischen Ordnung und der Reihung ($Preis > Verbrauch$) der Kriterien ist Ford die „beste“ Pareto-optimale Lösung. Beachtet werden muss dabei, dass wenn ein lexikographisches Minimum so wie in diesem Beispiel gewählt wird, sogar ein sehr guter Wert beim Spritverbrauch keine Kompensation eines hohen Preises sein kann (Ehrgott, 2005). In dieser Arbeit werden „Nicht-Skalarisierungsmethoden“ angewendet, um effiziente Lösungen zu erhalten und diese werden anschließend diskutiert und analysiert (siehe Kapitel 7).

4.4 Multikriterielle Shortest Path Probleme

Multikriterielle Shortest Path Probleme (MSPP) treten in verschiedenen Bereichen, wie in Computer-Netzwerken, in der Routenplanung, bei künstlicher Intelligenz, bei der Terminplanung und in vielen anderen Bereichen auf. MOSP (Multi-Objective Shortest Path)-Probleme können in vier Kategorien unterteilt werden (Tarapata, 2007):

- Anzahl der Optimierungskriterien
- Art des Problems (lexikographische Lösung, etc.)
- Lösungsmethode (label-setting, label-correcting, andere)
- Datenmodell (Arten der Netzwerk-Modelle, etc.)

Anwendungen, die sich mit dem MOSP-Problem auseinandersetzen, können in zwei häufig genutzte Bereiche eingeteilt werden: Computer-Netzwerk und Transport (Tarapata, 2007).

Im Vergleich zur „single-criterion“ Optimierung, wo die Kostenfunktion ein skalarer Wert ist, so ist sie bei der multikriteriellen Optimierung ein k -dimensionaler Vektor, der die heterogenen Kosten einer Kante repräsentiert (z.B. Gehzeit, Steigung), (Pressl, 2012).

In Abbildung 14 ist ein gerichteter Graph mit zwei Kriterien dargestellt, wobei die Kostenfunktionen aus zwei-dimensionalen Kostenvektoren bestehen. Es geht nun für den Algorithmus darum, die unterschiedlichen Kosten simultan zu berücksichtigen und zu verarbeiten. Speziell, wenn die Kosten gleich wichtig sind, ist es nicht einfach und klar, wie der kürzeste Weg bestimmt werden kann. Pareto-optimale Wege werden im nächsten Abschnitt eingeführt, um so multikriterielle Probleme zu lösen.

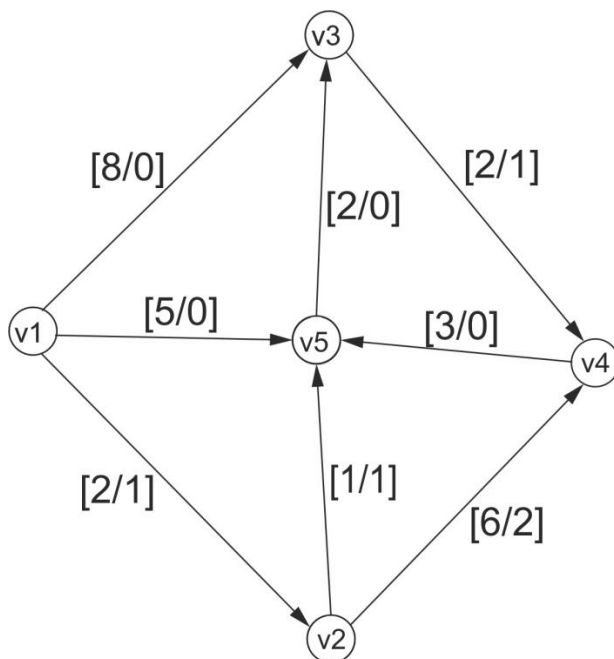


Abbildung 14: Kostenvektoren für die Kanten

4.4.1 Lösungen von multikriteriellen Shortest Path Problemen

Die Berechnung von effizienten Wegen basiert auf gerichteten Graphen $G = (V, E)$, bestehend aus einem Knotenset V und einem Kantenset E . Der Graph weist k -dimensionale Kostenvektoren $\vec{c}(e) = (c_1(e), \dots, c_k(e))$ mit k verschiedenen Kantenkosten für jede Kante $e \in E$ und k -dimensionale Labelvektoren $\vec{l}(v) = (l_1(v), \dots, l_k(v))$ für jeden Knoten $v \in V$ auf (Pressl, 2012).

Die ganzen Kosten werden für einen Weg wie folgt definiert:

Definition 7. „The summation of costs c for a Path P with m_p edges e is a vector sum, which is defined to be a component-by-component adding, (Schnee, 2009):“

$$\vec{c}(P) = \sum_{i=1}^{m_p} \vec{c}(e_i) = \left(\sum_{i=1}^{m_p} c_1(e_i), \sum_{i=1}^{m_p} c_2(e_i), \dots, \sum_{i=1}^{m_p} c_k(e_i) \right)$$

Es müssen einige theoretische Überlegungen gemacht werden, betreffend Pareto-optimale Wege.

Definition 8 (Martins, 1984). „If a path P in $G = (V, E)$ is a Pareto optimal path, then every sub-path of P is also a Pareto optimal path.“

Definition 9 (Martins, 1984). „Let P_1, P_2 and P_3 be three different Pareto optimal paths. If P_1 dominates P_2 and P_2 dominates P_3 , it follows that P_1 dominates P_3 .“

In Abbildung 15 werden alle Pareto-optimalen Lösungen für den Graph von Abbildung 14 vom Startknoten v_1 zu allen anderen Knoten, basierend auf Definition 2, in rot dargestellt. Alle anderen Lösungen sind in schwarz abgebildet.

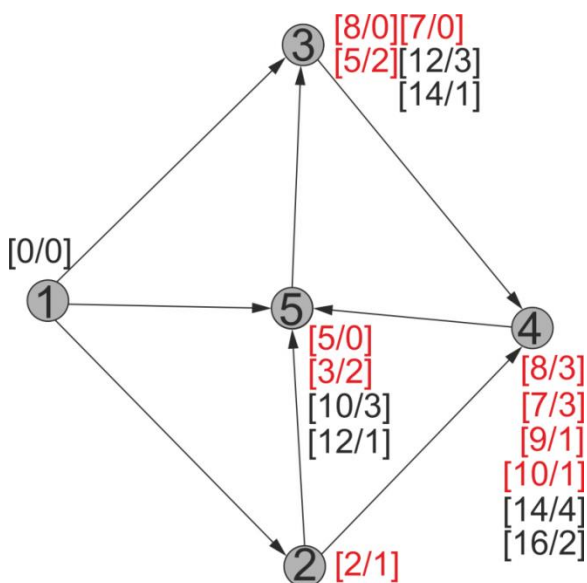


Abbildung 15: Alle Pareto-optimalen Lösungen des Beispielgraphen von Abbildung 14

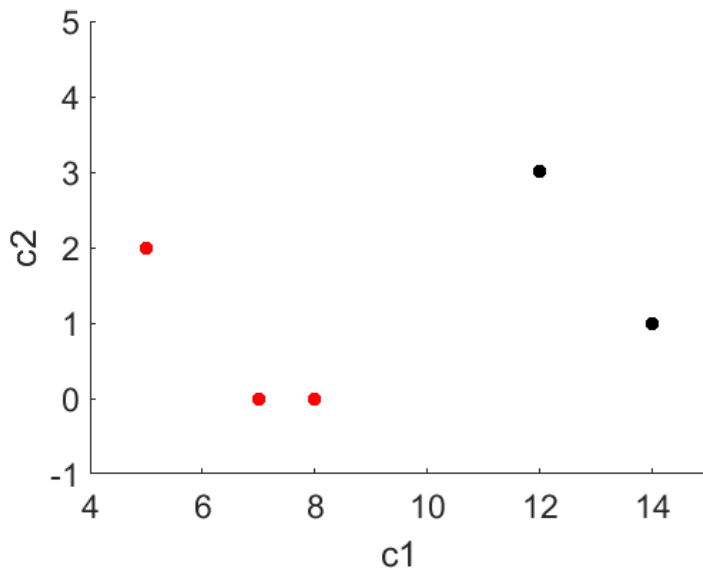


Abbildung 16: Illustration von Pareto-optimalen Lösungen vom Knoten 3 von Abbildung 15

In Abbildung 16 werden die Pareto-optimalen Lösungen vom Knoten 3 abgebildet (rote Punkte). Die Lösungen $(5/2)$ und $(7/0)$ sind strikte Pareto-optimale Lösungen. Die Anzahl an Pareto-optimalen Lösungen steigt mit zunehmender Knotenanzahl in einem Graph. Im schlimmsten Fall ist die Anzahl an Pareto-optimalen Lösungen exponentiell zu der Anzahl an Knoten (Ehrgott, 2005, S. 210), (Schnee, 2009). Dies ist ein entscheidender Faktor für die Laufzeit des Algorithmus. Deshalb werden Approximationsmethoden angewendet, die in den Kapiteln 5.2 und 5.3 diskutiert werden.

4.5 Dijkstra's Algorithmus bezüglich multikriterieller Optimierung

Die ersten, die den Dijkstra Algorithmus als Label-setting Algorithmus im Zusammenhang mit multikriterieller Optimierung eingeführt haben, waren (Hansen, 1979) und (Martins, 1984). Es kann ein erweiterter Dijkstra Algorithmus für SP-Probleme angewendet werden, um Pareto-optimale Wege zu finden. Die Einschränkung dabei ist, dass nur positive Kantenkosten im Graphen enthalten sind (Martins, 1984). Andernfalls muss ein Label-correcting Algorithmus verwendet werden, wie (Ehrgott, 2005, S. 224-226) zeigt. Es wird nun im Detail gezeigt wie ein Dijkstra Algorithmus für eine multikriterielle Optimierung implementiert wird und funktioniert.

4.5.1 Erweiterter Dijkstra Algorithmus

Dijkstra's Algorithmus kann verwendet werden, um alle Pareto-optimalen Wege sowie eine Teilmenge von Pareto-optimalen Wegen oder nur eine Pareto-optimale Lösung zu finden.

Verglichen mit der „single-criterion“ Optimierung ergeben sich einige Unterschiede beim Algorithmus (Pressl, 2012):

- Kantenkostenvektoren: Kostenwerte der „single-criterion“ Optimierung werden durch Kostenvektoren ersetzt
- Labelvektoren in den Knoten: Kantenkostenvektoren führen zu Labelvektoren in den Knoten; es werden die Label der Knoten der „single-criterion“ Optimierung durch einen oder mehrere Labelvektoren in den Knoten bei der multikriteriellen Optimierung ersetzt
- Approximationsmethoden: Methoden, um ein Set aus Pareto-optimalen Lösungen auszuwählen, werden verwendet, um die Anzahl der Lösungen zu reduzieren und damit die Laufzeit zu verringern (siehe Kapitel 5.3)
- Einschränkungen: Vordefinierte Beschränkungen und obere Schranken für die Vektorkomponenten werden verwendet, um die Menge an Lösungen zu reduzieren (siehe Kapitel 5.3.2)
- Beendigung: Die Beendigungskriterien unterscheiden sich von „single-criterion“ Anwendungen; der Algorithmus wird nicht beendet, wenn das erste Label im Zielknoten permanent gesetzt wurde

Der Algorithmus ist für weitere Kriterien erweiterbar, indem die Kantenkostenvektoren und die Labelvektoren der Knoten erweitert werden.

Berechnung aller Pareto-optimalen Wege

Der Algorithmus, der nun beschrieben wird, wurde 1984 von Martins (Martins, 1984) veröffentlicht. Es ist ein multikriterieller Label-setting Algorithmus der alle Pareto-optimalen Wege berechnet. Der Algorithmus wird in Algorithmus 2 näher beschrieben. Die Labelvektoren der Knoten werden erweitert, um eine multikriterielle Optimierung zu gewährleisten. Der Algorithmus verwendet die lexikographische Ordnung, um die Label auszuwählen und die Definition der Dominanz (Definition 2) für das Vergleichen von Labeln (Martins, 1984).

Es sei v_i ein Knoten aus $G = (V, E)$. Der Labelvektor des Knoten $\vec{l}(v_i)$ ist ein $k + 3$ Tupel, wobei k die Anzahl der Kosten darstellt. Der Labelvektor $\vec{l}(v_i) = (c_1, \dots, c_k, p_j, z, u)$ eines Knoten von $G = (V, E)$ besteht aus den Kosten c_1, \dots, c_k , dem Vorgänger $p_j \neq v_i$, z , das die Labelnummer vom Knoten p_j darstellt, und u , welches die Labelnummer im Knoten v_i symbolisiert. Das bedeutet also, dass ein Labelvektor aus einem k -dimensionalen Kostenvektor, einem Knotenvorgänger, also derjenige Knoten von dem aus das Label entstanden ist, einer Labelnummer, die aussagt, auf welches Label es sich im Vorgängerknoten bezieht, und einer Labelnummer im aktuellen Knoten besteht. Das Set T sei eine Liste temporärer Label, welche einer vordefinierten Ordnung unterliegt (z.B. lexikographisch). Das Set P sei eine Liste permanenter Label, die effiziente Wege speichert.

Label werden permanent gesetzt, anstatt Knoten (Ehrgott, 2005).

Algorithmus 2. SP - Algorithmus basierend auf Dijkstra's Ansatz mit multikriterieller Erweiterung (Martins, 1984):

V ... Set aller Knoten
 T ... Set aller temporären Label
 P ... Set aller permanenten Label
 N ... Set von adjazenten Knoten (Nachbarschaft)
 v_s ... Startknoten
 $\vec{l}(v_s)$... Labelvektor des Startknoten
 $\vec{l}(v_i)$... Labelvektor eines Knoten v_i
 p_j ... Vorgänger eines Knoten v_j
 \vec{c}_{ij} ... Kostenvektor eines Bogen a_{ij}

Eingabeparameter

- Graph $G = (V, E)$
- Kantenkostenvektor $\vec{c} = (c_1, \dots, c_k, 0, 0, 0) \in \mathbb{R}^+$
- Startknoten $v_s \in V$
- Zielknoten $v_t \in V$

Ausgabeparameter

- Alle Pareto-optimalen Lösungen vom Startknoten v_s zum Zielknoten v_t und zu allen anderen Knoten im Graph $G = (V, E)$
- Labelvektoren $\vec{l} = (c_1, \dots, c_k, p_j, z, u)$ im Set P

Initialisierung

$$\vec{l}(v_s) := (0, \dots, 0, 0, 0, 1).$$

$$\vec{l}(v) = \infty \text{ für alle } v \in V \setminus v_s.$$

$$T \leftarrow \{\vec{l}(v_s)\}. P \leftarrow \{\}.$$

Ablauf

1. WHILE $T \neq \{ \}$ DO
2. $\vec{l}(v_i) := \vec{l} \in T$ vom Knoten v_i mit $\min_{lex} \vec{l}(v_i)$;
3. $P \leftarrow P + \{\vec{l}(v_i)\}$; $T \leftarrow T \setminus \{\vec{l}(v_i)\}$;
4. FOR $v_j \in N(v_i)$ DO
5. $\vec{l}(v_j) := \vec{l}(v_i) + \vec{c}_{ij}$; $l_{p_j}(v_j) := v_i$; $l_z(v_j) := l_u(v_i)$; set $l_u(v_j)$;
6. $T \leftarrow T + \{\vec{l}(v_j)\}$;
7. IF $\vec{l}(v_j)$ is dominated by an existing label $\in T$ or P at v_j THEN
8. $T \leftarrow T \setminus \{\vec{l}(v_j)\}$; END IF;
9. IF $\vec{l}(v_j)$ dominates an existing label $\vec{l}(v_x)$ at v_j THEN
10. $T \leftarrow T \setminus \{\vec{l}(v_x)\}$; END IF;
11. END FOR
12. END WHILE

Basierend auf dem Graph von Abbildung 14 wird ein Beispiel des erweiterten Dijkstra Algorithmus durchgespielt, um alle Pareto-optimalen Wege zu finden. Die Iterationsschritte des Algorithmus und die Labelvektoren werden folgend beschrieben und in den Abbildungen 17-22 dargestellt. Label, die dominiert werden, werden gelöscht und herausgestrichen.

Schritte des Algorithmus für das angeführte Beispiel

Initialisierung: Der Startknoten $v_s = v_1$ wird gesetzt. Der Labelvektor des Startknoten $\vec{l}(v_s) = (0,0,0,0,1)$ wird in das Set T aufgenommen. Die Labelvektoren aller anderen Knoten werden mit ∞ definiert.

Erster Schleifendurchgang: Der Labelvektor des Startknotens ist der einzige Eintrag in T und wird permanent gesetzt. Dann werden die Labelvektoren der Nachbarknoten bestimmt. Die Labelvektoren $(2,1,1,1,1)$ im Knoten v_2 , $(8,0,1,1,1)$ im Knoten v_3 und $(5,0,1,1,1)$ im Knoten v_5 werden gesetzt. Diese Labelvektoren werden in das Set T aufgenommen.

Zweiter Schleifendurchgang: Da der Labelvektor des Knoten v_2 aus T lexikographisch der kleinste ist, wird er ausgewählt und permanent gesetzt. Basierend auf diesem Label werden die Nachbarknoten wieder betrachtet und upgedatet. Dies ergibt die Labelvektoren $(8,3,2,1,1)$ im Knoten v_4 und den zusätzlichen Labelvektor $(3,2,2,1,2)$ im Knoten v_5 , die dem Set T hinzugefügt werden. Da sich die beiden Labelvektoren im Knoten v_5 nicht gegenseitig dominieren (siehe Definition 2), bleiben sie in T .

Dritter Schleifendurchgang: Der nächste Labelvektor, der lexikographisch am kleinsten ist, ist $(3,2,2,1,2)$ im Knoten v_5 . Dieser wird ausgewählt und permanent gesetzt. Die Nachbarknoten werden upgedatet, was zu einem neuen Labelvektor $(5,2,5,2,2)$ im Knoten v_3 führt. Der neue Labelvektor wird dem Set T hinzugefügt, da er nicht dominiert wird.

Vierter Schleifendurchgang: Nun ist der Labelvektor $(5,0,1,1,1)$ des Knoten v_5 lexikographisch der kleinste in T , er wird ausgewählt und permanent gesetzt. Der Nachbarknoten v_3 wird upgedatet und erhält ein weiteres Label mit $(7,0,5,1,3)$. Auch dieses Label wird nicht dominiert und wird im Set T aufgenommen.

Fünfter Schleifendurchgang: Das nächste Label, das permanent gesetzt wird, ist der Labelvektor $(5,2,5,2,2)$ des Knoten v_3 . Der Nachbarknoten v_4 wird upgedatet, wobei der Labelvektor $(7,3,3,2,2)$ entsteht. Beide Labelvektoren im Knoten v_4 dominieren sich nicht gegenseitig und sind deshalb in T vorhanden.

Sechster Schleifendurchgang: Der lexikographisch kleinste Labelvektor in T ist nun $(7,0,5,1,3)$ des Knoten v_3 . Dieser wird wieder ausgewählt und permanent gesetzt. Dem Nachbarknoten v_4 wird das neue Label mit $(9,1,3,3,3)$ zugeschrieben und, da es nicht dominiert wird, wird es in T aufgenommen.

Siebter Schleifendurchgang: Der nächste Labelvektor, der permanent gesetzt wird, ist $(7,3,3,2,2)$ des Knoten v_4 . Der Nachbarknoten v_5 wird mit dem Labelvektor $(10,3,4,2,3)$ upgedatet. Dieses Label wird von den Labeln $(5,0,1,1,1)$ und $(3,2,2,1,2)$ dominiert und wird somit gelöscht.

Achter Schleifendurchgang: Der lexikographisch kleinste Labelvektor ist nun $(8,0,1,1,1)$ des Knoten v_3 . Der Nachbarknoten v_4 wird mit dem Label $(10,1,3,1,4)$ upgedatet und, da dieses Label nicht dominiert wird und auch kein anderes Label dominiert, wird es in T aufgenommen und auch die bereits vorhandenen Labelvektoren dieses Knoten bleiben T erhalten.

Neunter Schleifendurchgang: Der nächste Labelvektor, der permanent gesetzt wird, ist $(8,3,2,1,1)$ des Knoten v_4 . Durch das Updaten des Nachbarknoten v_5 entsteht der Labelvektor $(11,3,4,1,4)$, der dominiert wird und somit gelöscht wird.

Zehnter Schleifendurchgang: Der nächste Labelvektor, der lexikographisch am kleinsten ist, ist $(9,1,3,3,3)$ des Knoten v_4 . Es wird wieder der Nachbarknoten v_5 upgedatet, was zu einem Labelvektor von $(12,1,4,3,5)$ führt. Dieser wird dominiert und gelöscht.

Elfter Schleifendurchgang: Es ist nur mehr ein Eintrag in T vorhanden. Dieser Labelvektor $(10,1,3,1,4)$ des Knoten v_4 wird permanent gesetzt. Der Nachbarknoten v_5 wird noch einmal upgedatet, da aber auch dieses Label $(13,1,4,4,6)$ dominiert wird, wird es gelöscht.

Da nun die Liste der temporär gesetzten Label T leer ist, wird der Algorithmus beendet. Es sind nun alle Pareto-optimalen Wege vom Knoten v_1 zu allen anderen Knoten ermittelt und werden durch die Labelvektoren repräsentiert. Schlussendlich können die Pareto-optimalen Wege gefunden werden indem man die Vorgängerlabel p_j und z verwendet und die Knoten zurückverfolgt. Wenn man für das vorliegende Beispiel annimmt, man will vom Startknoten v_1 zum Zielknoten v_4 , dann erhält man die in der Tabelle 6 dargestellten Pareto-optimalen Wege.

Tabelle 6: Pareto-optimalen Wege vom Knoten 1 bis zum Knoten 4

Wege	Lösung 1	Lösung 2	Lösung 3	Lösung 4
Vom Knoten	1	1	1	1
	2	2	5	3
Bis Knoten	4	5	3	4
Bis Knoten		3	4	
Bis Knoten		4		
Kosten c_1	8	7	9	10
Kosten c_2	3	3	1	1

Der Unterschied zwischen der „single-criterion“ Optimierung oder dem Finden einer einzigen approximierten Lösung und der multikriteriellen Optimierung, wobei mehrere Lösungen entstehen, muss hier erwähnt werden. Wenn eine einzelne Lösung berechnet wird, dann kann der Algorithmus abbrechen, sobald ein Label im Zielknoten v_t permanent gesetzt wurde. Dies ist bei MOLSA (Multiobjective Label-Setting Algorithmus) aufgrund der Existenz von mehr als einem effizienten Weg, zwischen einem Paar von Knoten, nicht möglich (Pressl, 2012). Eine gesamte Lösung kann nur produziert werden, wenn alle effizienten Wege zu allen Knoten bestimmt werden. Andernfalls resultiert eine Approximation, wenn man ein Abbruchkriterium einführt. Dieses Abbruchkriterium könnte ein räumliches Kriterium bezüglich des Suchgraphen oder ein Kriterium mit Bezug auf die Laufzeit (also z.B. Laufzeitbegrenzung) des Algorithmus sein. Wie bereits erwähnt wurde, kann die Berechnung aller Pareto-optimalen Wege nicht in polynomischer Zeit erfolgen (Pressl, 2012). Deshalb wird oft nur eine Teilmenge aller effizienten Lösungen berechnet (Kapitel 5).

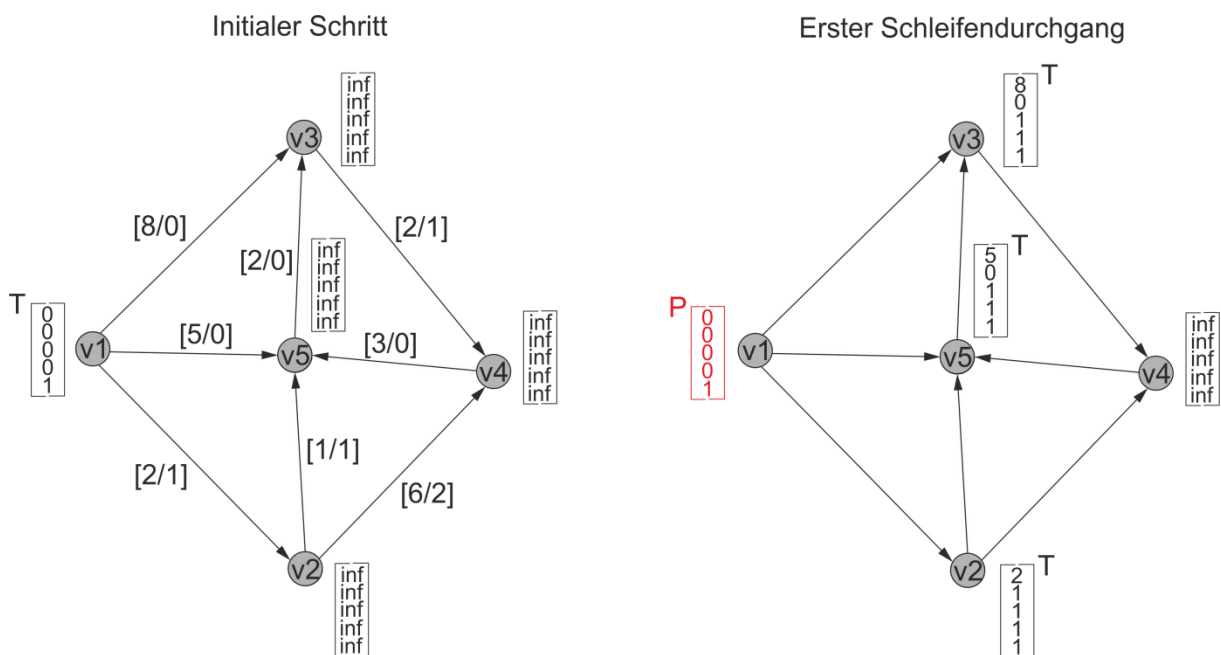


Abbildung 17: Initialisierung und erster Schritt des erweiterten Dijkstra Algorithmus, um alle Pareto-optimalen Wege zu finden, mit 2 Kriterien

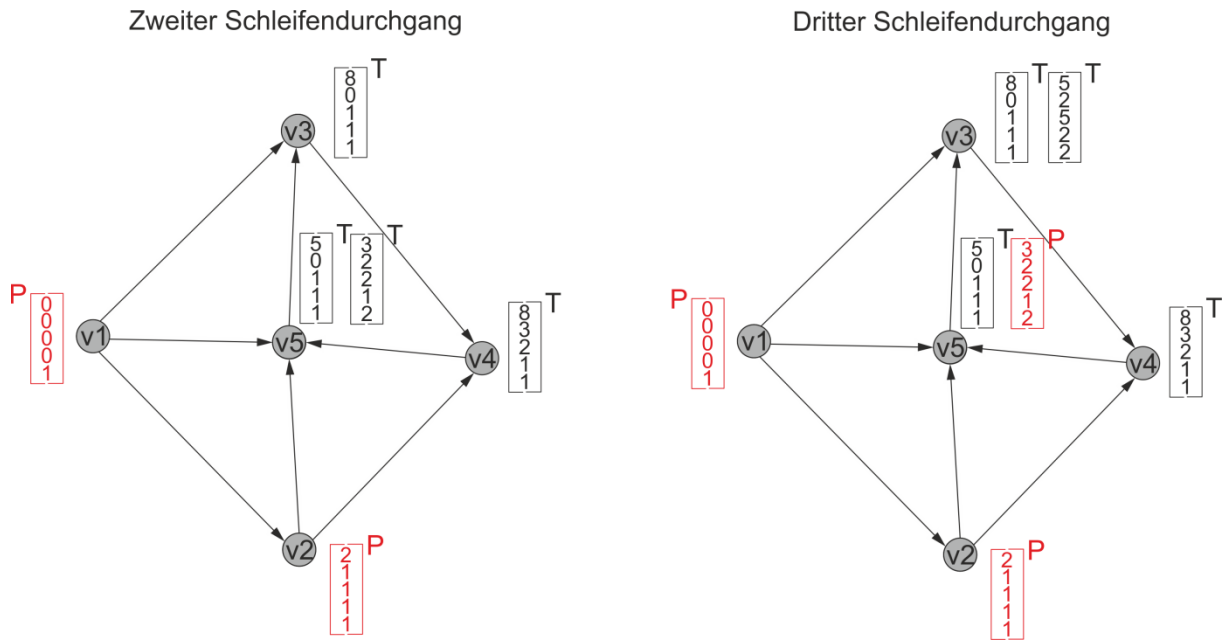


Abbildung 18: Zweiter und dritter Schritt des erweiterten Dijkstra Algorithmus, um alle Pareto-optimalen Wege zu finden, mit 2 Kriterien

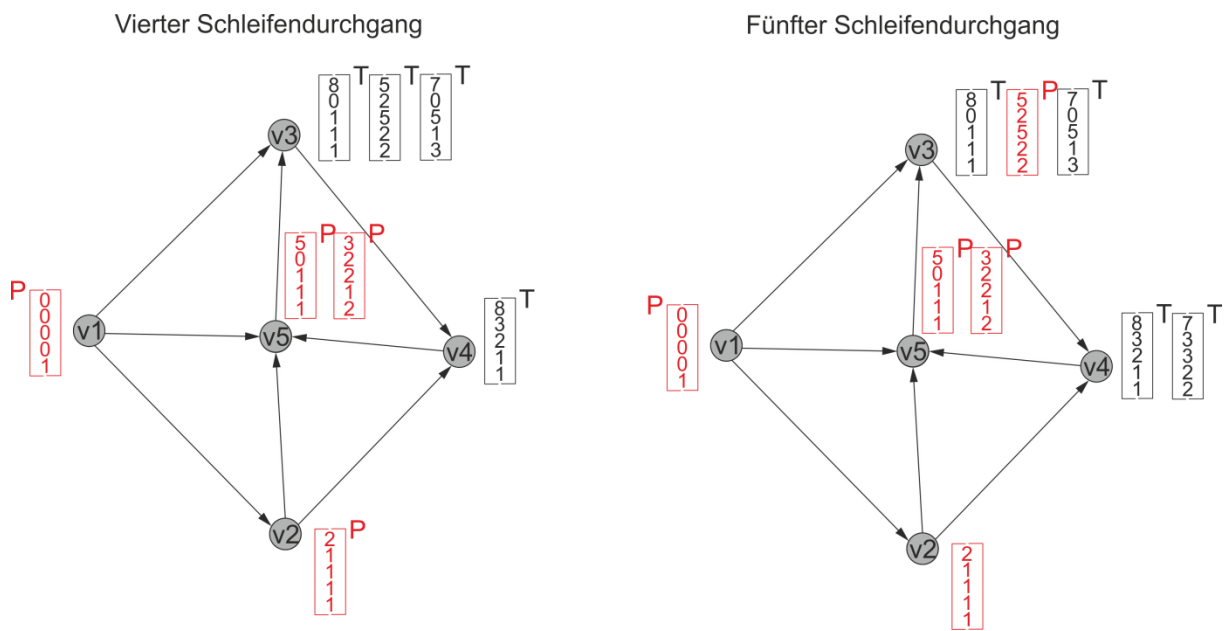


Abbildung 19: Vierter und fünfter Schritt des erweiterten Dijkstra Algorithmus, um alle Pareto-optimalen Wege zu finden, mit 2 Kriterien

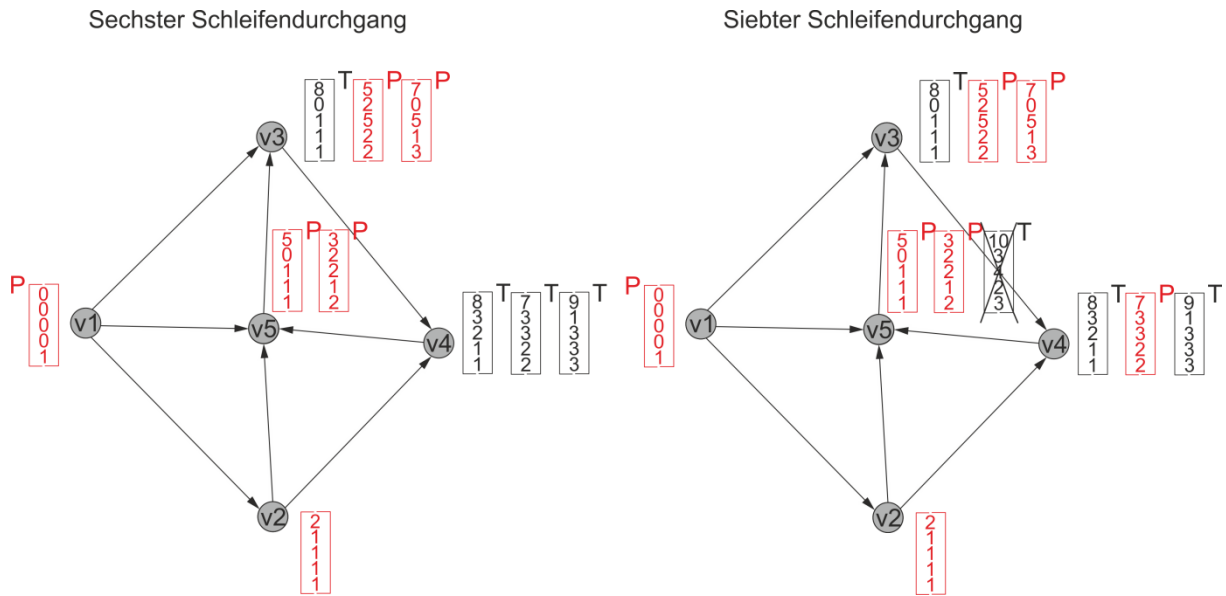


Abbildung 20: Sechster und siebter Schritt des erweiterten Dijkstra Algorithmus, um alle Pareto-optimalen Wege zu finden, mit 2 Kriterien

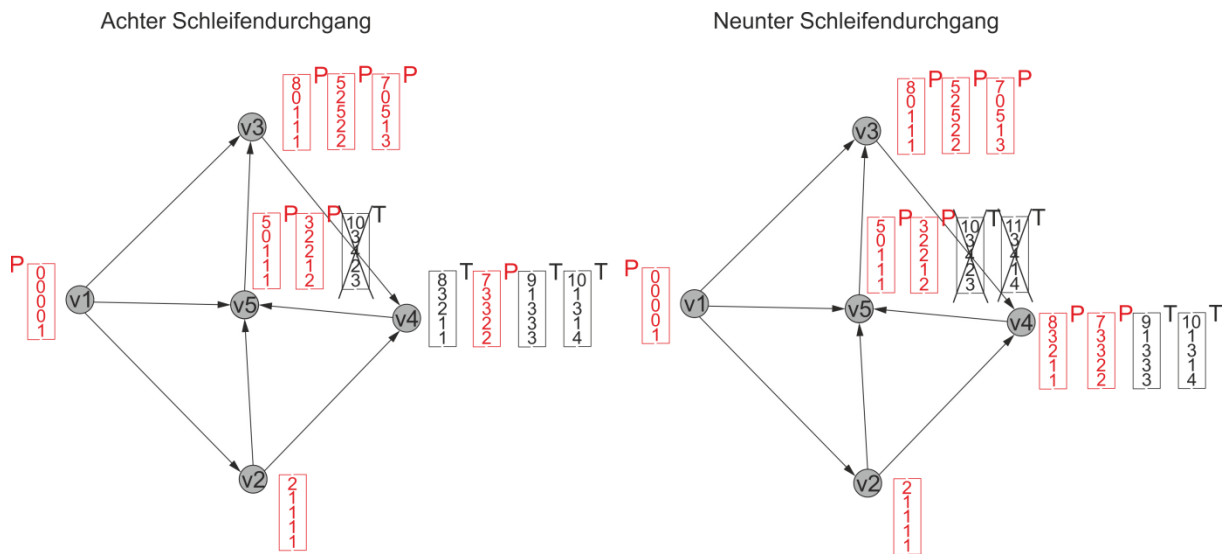
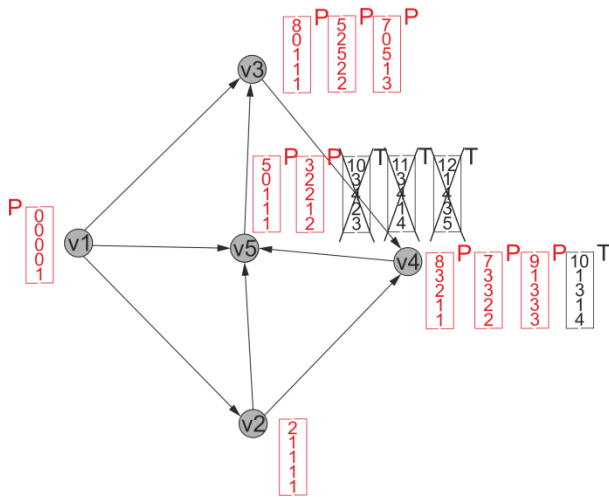


Abbildung 21: Achter und neunter Schritt des erweiterten Dijkstra Algorithmus, um alle Pareto-optimalen Wege zu finden, mit 2 Kriterien

Zehnter Schleifendurchgang



Elfter Schleifendurchgang

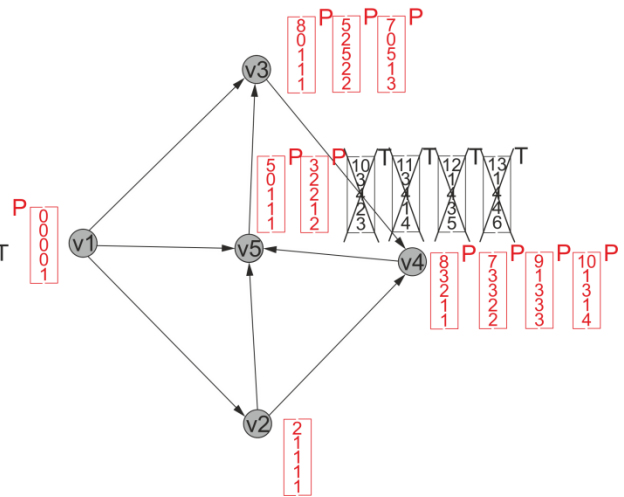


Abbildung 22: Zehnter und elfter Schritt des erweiterten Dijkstra Algorithmus, um alle Pareto-optimalen Wege zu finden, mit 2 Kriterien

5 Konzepte von multikriterieller Optimierung für die Routenplanung

In diesem Kapitel werden verschiedene Konzepte der multikriteriellen Optimierung für die Routenplanung, die auf den theoretischen Definitionen von Kapitel 4 basieren, beschrieben. Im weiteren Verlauf der Arbeit wird dann eine multikriterielle Routenplanung, mit verschiedenen Methoden und Varianten, für Wanderwege realisiert (ab Kapitel 6).

5.1 Überblick

5.1.1 Strikte Pareto-optimale Wege

In der Routenplanung will man meistens strikte Pareto-optimale Lösungen, da schwache Lösungen für den Anwender/die Anwenderin meistens von geringerem Interesse sind (Pressl, 2012). Wenn man im Falle des Wandergebietes zum Beispiel die zwei Kriterien Gehzeit und schöne Aussicht annimmt und es zwei Lösungen gibt, die eine idente Gehzeit aufweisen, so wählt man üblicherweise den Weg mit der schöneren Aussicht. Oder man möchte als zweites Kriterium neben der Gehzeit die durchschnittliche Steigung berücksichtigen, so wird man normalerweise für zwei Wege gleicher Gehzeit den Weg mit geringerer durchschnittlicher Steigung wählen. Dies hängt natürlich immer vom Nutzer/von der Nutzerin und womöglich noch von weiteren Kriterien ab, aber im idealisierten Fall, ist dies so anzunehmen.

5.1.2 Komplexitäten der multikriteriellen Optimierung

Wenn man die Resultate und Komplexitäten an Lösungen der multikriteriellen Optimierungen betrachtet, so lassen sich diese in drei verschiedene Gruppen einteilen. Diese sind wie folgt (Schnee, 2009):

- Alle Pareto-optimale Lösungen (exponentielle Laufzeit, gesamte Lösung)
- Eine Teilmenge von Pareto-optimale Lösungen (polynomische Laufzeit, Approximation)
- Eine einzelne Pareto-optimale Lösung (polynomische Laufzeit, Approximation)

Die erste Variante, d.h. alle Pareto-optimale Lösungen, wurde in Kapitel 4 behandelt. Die zweite und die dritte Variante werden in diesem Kapitel behandelt. Wie bereits erwähnt wurde, ist die Einführung der Varianten zwei und drei notwendig, um die Berechnungen in polynomischer Zeit durchzuführen.

In Abbildung 23 werden Approximationsmethoden dargestellt, um diese zwei Varianten (Einzellösung und Teilmenge der Lösung) zu lösen. Im weiteren Verlauf werden sie beschrieben und diskutiert. Die Beschreibung der Varianten basiert auf einer Zwei-Kriterien-Optimierung. Für mehrere Kriterien können die Methoden aber einfach erweitert werden (Pressl, 2012).

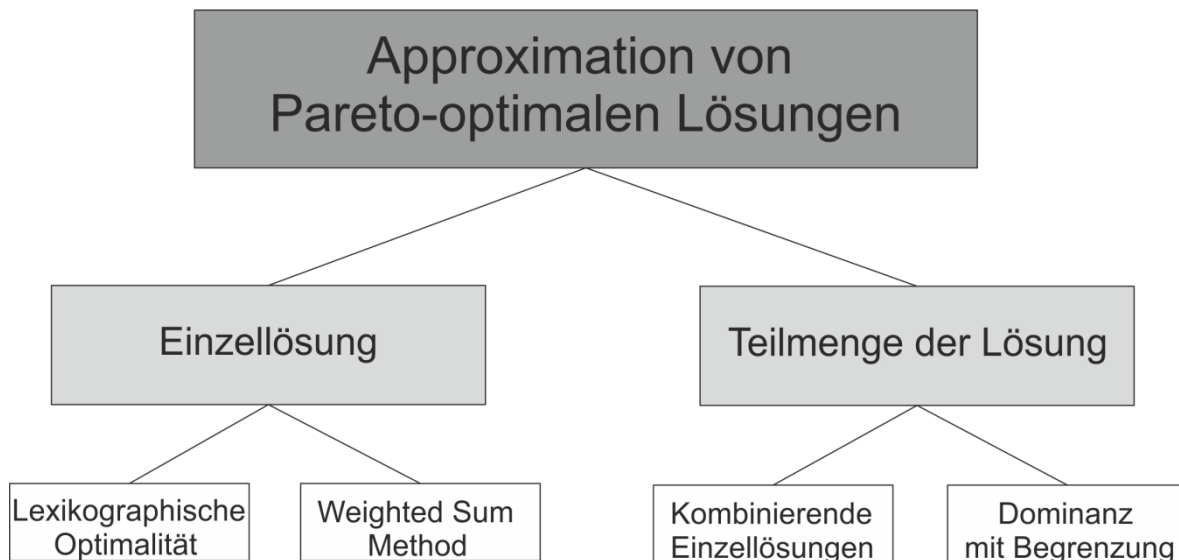


Abbildung 23: Approximationsmethoden der multikriteriellen Optimierung für die Routenplanung

5.2 Einzelne Pareto-optimale Lösungen

5.2.1 Optimierung mit lexikographischer Optimalität

Im Fall einer Zwei-Kriterien Optimierung werden die zwei Kriterien $f_1 = \text{"Gehzeit"}$ und $f_2 = \text{"Schöne Aussicht"}$ berücksichtigt. Wenn man die lexikographische Optimalität (siehe Kapitel 4.3.2) anwendet, so liegt der Fokus auf der Zeit und der schnellste Weg wird berechnet. Ein Beispielgraph, der auf dem Graph von Abbildung 14 beruht, wird in Abbildung 24 dargestellt. Dem Kriterium $f_2 = \text{"Schöne Aussicht"}$ werden hierbei skalare Werte zugeordnet, wobei der Wert 0 die schönste Aussicht repräsentiert und je höher der Wert wird, desto schlechter ist die Aussicht. Das lexikographische Minimum betreffend den Kriterien $f_1 = \text{"Gehzeit"}$ und $f_2 = \text{"Schöne Aussicht"}$ für den Zielknoten v_4 ist die Lösung $(7,3)$. Das Ergebnis liefert eine Gehzeit von 7 Stunden und eine schöne Aussicht entlang des Weges, die mit dem Wert 3 repräsentiert wird und im Vergleich mit anderen Lösungen als „eher durchschnittlich bis schlecht“ interpretiert werden kann. Vergleicht man nun diese Lösung mit allen Pareto-optimalen Lösungen dieses Beispielgraphen (siehe Tabelle 6) so sieht man, dass die Lösungen $(8,3)$, $(9,1)$ und $(10,1)$ aufgrund der Approximation mit lexikographischem Minimum verloren gehen.

5.2.2 Optimierung mit Hilfe der Weighted Sum Methode

Wie im Kapitel 4.3.1 beschrieben ist die WSM (Weighted Sum Method) eine Linearkombination der Optimierungskriterien, wobei positive Gewichte verwendet werden. Wendet man diese Methode bei einer Zwei-Kriterien-Optimierung mit den Optimierungskriterien f_1 und f_2 an, so kann die Minimierung der Linearkombination wie folgt geschrieben werden:

$$\min_{x \in X} \sum_{i=1}^2 \lambda_i f_i(x) = \lambda_1 f_1(x) + \lambda_2 f_2(x)$$

Die Gewichte λ_1 und λ_2 beschreiben den Einfluss der Kriterien. Wenn zum Beispiel das erste Kriterium $f_1(x)$ doppelt so wichtig ist wie das zweite Kriterium $f_2(x)$, so wird $\lambda_1 = 2$ und $\lambda_2 = 1$ definiert. Die Linearkombination sieht dann wie folgt aus:

$$\min_{x \in X} (\lambda_1 f_1(x) + \lambda_2 f_2(x)) = \min_{x \in X} (2f_1(x) + 1f_2(x))$$

5.3 Teilmenge von Pareto-optimalen Lösungen

5.3.1 Teilmenge durch Kombination von Einzellösungen

Zwei-Kriterien Optimierung mit lexikographischer Optimalität, wobei eine andere Kriterienreihenfolge verwendet wird

Wenn man die Reihenfolge der Kriterien verändert, so ergeben sich, neben den bereits vorhandenen, neue Lösungen. Betrachtet man das Beispiel vom Kapitel 5.2.1 so wird die Reihenfolge verändert und die berücksichtigten Kriterien sind $f_1 = \text{"Schöne Aussicht"}$ und $f_2 = \text{"Gehzeit"}$, wobei wieder eine lexikographische Optimierung angewendet wird. In der Praxis kann dies leicht umgesetzt werden, da die Laufzeit des Algorithmus für eine lexikographische Lösung gering ist und somit eine zweite Berechnung einfach durchzuführen ist. Jetzt, mit geänderter Reihenfolge, ist die Lösung (1,9) das lexikographische Minimum im Zielknoten v_4 . Das Ergebnis liefert eine schöne Aussicht, die mit dem Wert 1 repräsentiert wird und als „sehr gut“ interpretiert werden kann und eine Gehzeit von 9 Stunden. Die entsprechenden Schritte des Algorithmus sind in Abbildung 25 dargestellt. Beachtet werden muss, dass die Reihenfolge auch alle Labelvektoren und Kantenkostenvektoren betrifft. Im Vergleich zu den Schritten des Algorithmus in Abbildung 24 ändert sich die Sequenz, wie die Label entstehen. Beim zweiten Schleifendurchgang in Abbildung 25 wird der Knoten v_5 mit (0,5) permanent gesetzt anstatt des Knoten v_2 , wie in Abbildung 24. Auch das Label (0,7) im Knoten v_3 wird in Abbildung 25 schon im zweiten Schleifendurchgang erzeugt, was dann das Vorgängerlabel des Label (1,9) im Zielknoten v_4 ist.

In diesem Fall ist das Label (1,9) im Zielknoten, die Lösung mit der schönsten Aussicht unter allen Pareto-optimalen Lösungen und unter allen Lösungen mit dem gleichen Maß an Schönheit der Aussicht, die Lösung mit geringster Zeitdauer.

Es muss hier erwähnt werden, dass im schlimmsten Fall die Zeitkomponente sehr hoch sein kann, da eine Lösung mit einer gering höheren Bewertung der schönen Aussicht eine Lösung mit sehr hoher Zeitdauer nicht dominiert. Es könnte so auch eine Lösung entstehen, die zum Beispiel mit (1,30) das lexikographische Minimum darstellt, wenn die Lösung die einzige mit der Bewertung 1 für die schöne Aussicht ist. Diese Umstände führen im nächsten Kapitel zur Einführung von Beschränkungen bezüglich der Kriterien (Pressl, 2012).

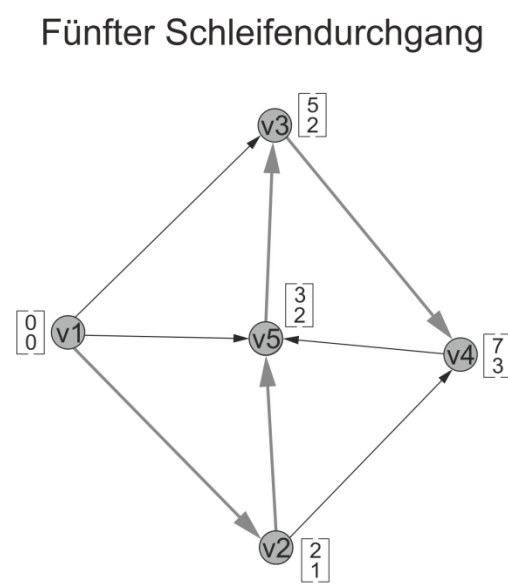
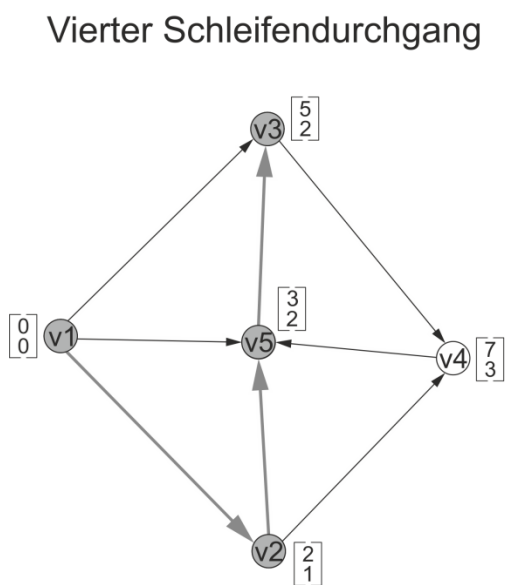
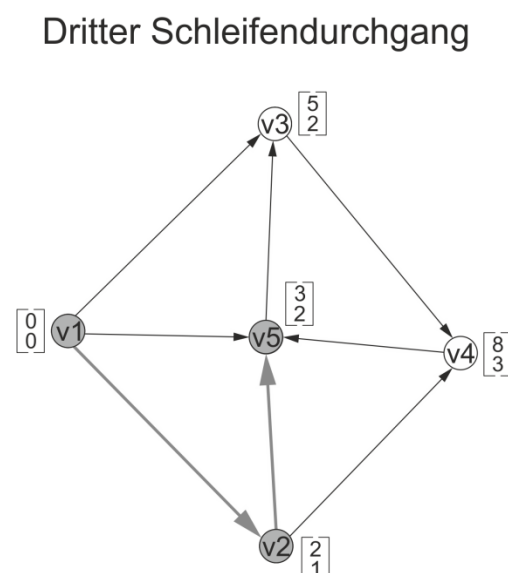
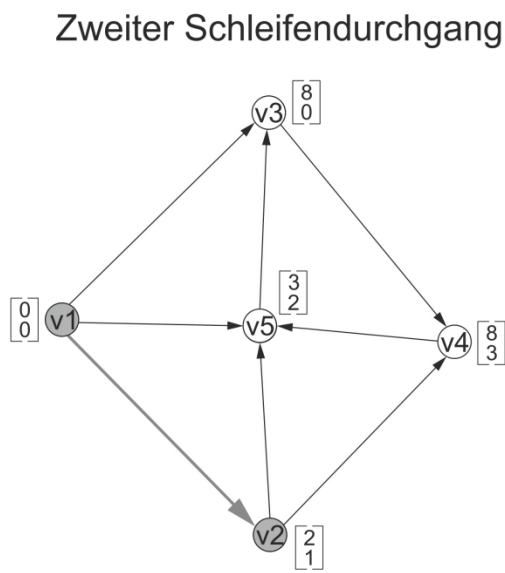
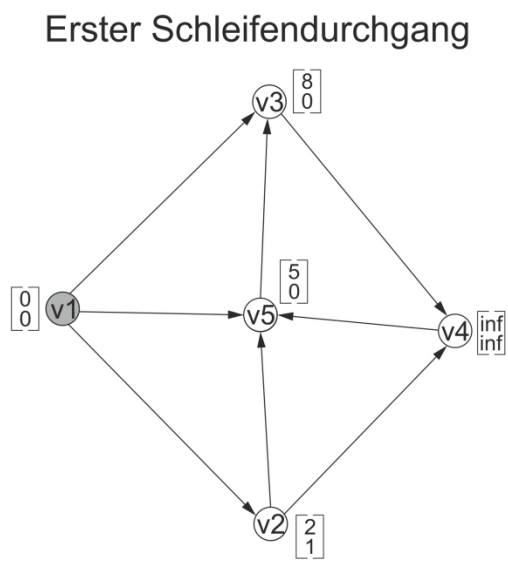
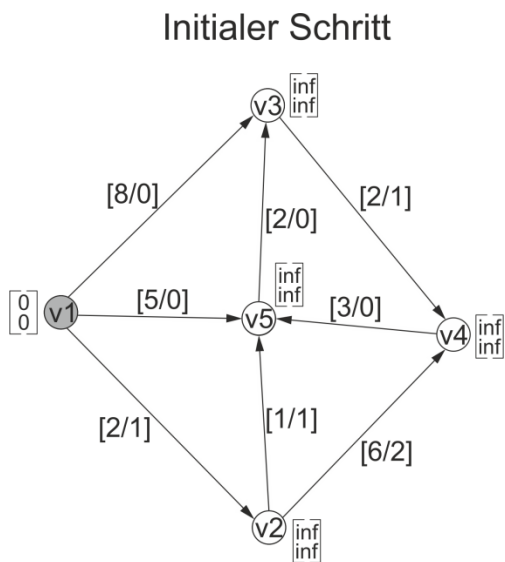


Abbildung 24: Ablauf des erweiterten Dijkstra Algorithmus um eine Pareto-optimale Lösung zu berechnen (Lexikographisch mit f_1 ="Gehzeit" und f_2 ="Schöne Aussicht")

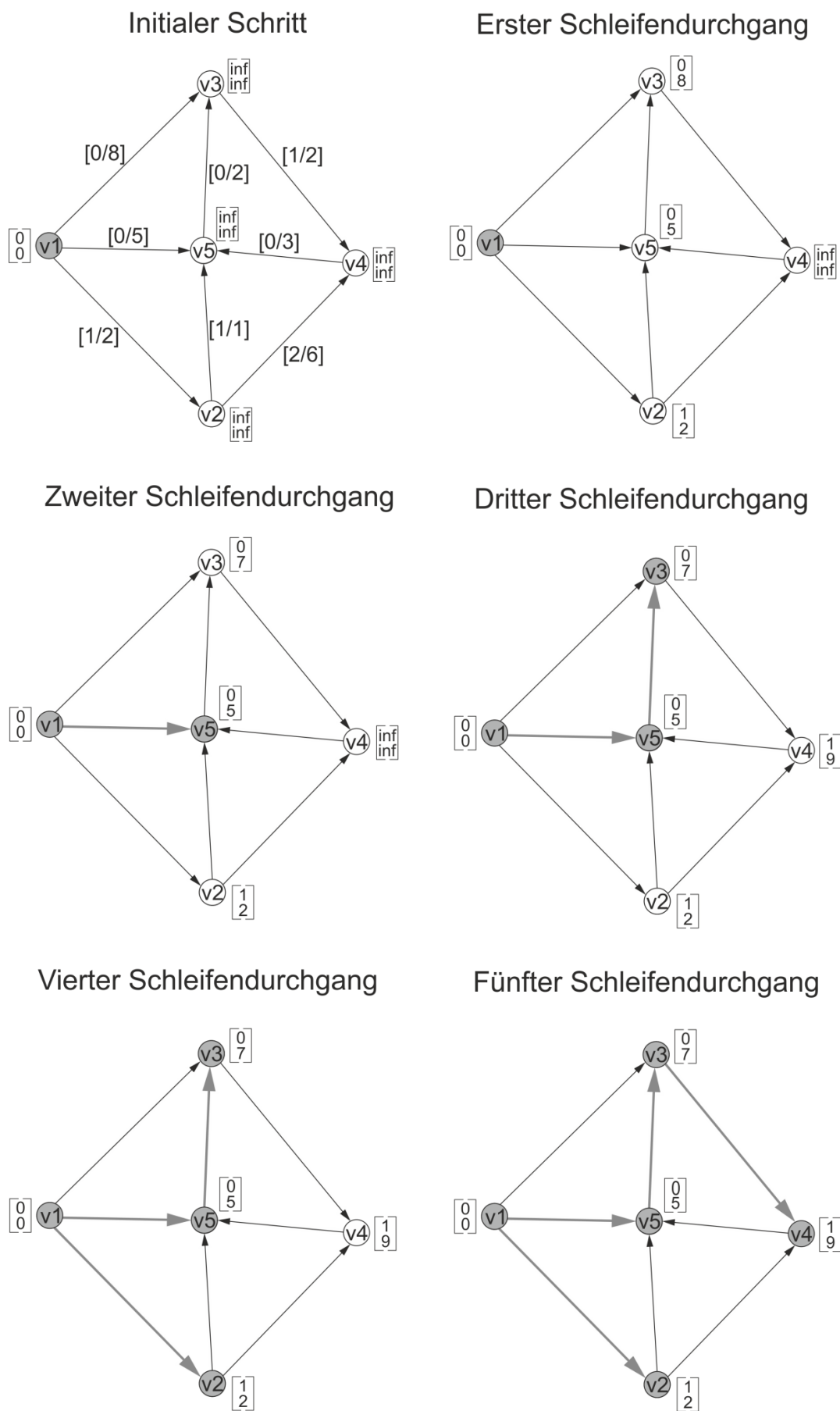


Abbildung 25: Ablauf des erweiterten Dijkstra Algorithmus um eine Pareto-optimale Lösung zu berechnen (Lexikographisch mit f_1 ="Schöne Aussicht" und f_2 ="Gehzeit")

Generell kann man sagen, dass durch die Anwendung des lexikographischen Minimums mit unterschiedlicher Reihenfolge der Kriterien bei einer Zwei-Kriterien Optimierung zwei strikte Pareto-optimale Lösungen entstehen. Diese Lösungen weisen einmal den Minimumwert von f_1 und einmal von f_2 auf. Diese Lösungen sind in Abbildung 26 dargestellt. Alle anderen strikten Pareto-optimale Lösungen dazwischen können mit dieser Methode nicht gefunden werden (Pressl, 2012).

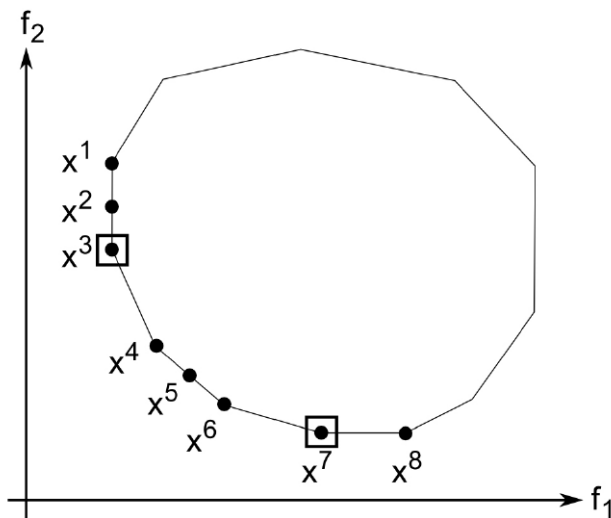


Abbildung 26: Lexikographische Minimumslösungen¹¹

Die WSM Methode mit unterschiedlichen Gewichtungskombinationen

Teilmenge von Pareto-optimale Lösungen können auch durch unterschiedliche Gewichtungskombinationen, die auf den Algorithmus angewendet werden, erzeugt werden. Eine Lösung kann zum Beispiel durch das Anwenden von $\lambda_1 = 1$ und $\lambda_2 = 2$ und eine andere durch das Anwenden von $\lambda_1 = 2$ und $\lambda_2 = 1$ entstehen.

5.3.2 Teilmenge durch Dominanz mit Begrenzung

Zwei-Kriterien Optimierung mit Beschränkungen der Kriterien

Basierend auf der Methode alle Pareto-optimale Lösungen zu finden, ist diese Methode eine, die Beschränkungen bezüglich der Kriterien einführt. Das reduziert die Anzahl an Pareto-optimale Lösungen in den Knoten und beschleunigt somit den Algorithmus. Es entstehen so nur Pareto-optimale Lösungen P , die vordefinierten Beschränkungen bezüglich der Kriterien genügen. Die Beschränkungen sind durch die Ungleichung $c_i(P) \leq \tau_i$ (Hallam,

¹¹ Bildquelle: (Pressl, 2012)

Harrison, & Ward, 2001) für das i -te Kriterium des Weges definiert. Die obere Grenze des i -ten Kriteriums ist dabei τ_i . Diese Erweiterung des MOSP (Multi-Objective Shortest Path)-Problems ist diesbezüglich ein MOSP-Problem mit Beschränkungen und wird auch als RSPP (Restricted Shortest Path Problem) bezeichnet. Der Algorithmus überprüft die Komponenten der Labelvektoren auf Tauglichkeit, bezüglich der vordefinierten Beschränkungen mit Hilfe der Ungleichung von (Hallam, Harrison, & Ward, 2001). Label die nicht dominiert werden, werden gelöscht, sobald der Grenzwert von mindestens einem Kriterium überschritten wird.

6 Netzgenerierung

6.1 Überblick

Ziel ist es ein geeignetes Testgebiet zu erschaffen, das für die Anwendung eines erweiterten Dijkstra Algorithmus mit multikriterieller Wegeoptimierung verwendet werden kann. Es soll eine Knoten-Kanten Struktur in Form eines gerichteten Graphen geschaffen werden. Da Wanderwege in beide Richtungen begehbar sind, wird eine Kante durch zwei Bögen (beide Richtungen) dargestellt.

Bevor auf die tatsächliche Netzgenerierung eingegangen wird, werden zuerst Themen wie geographische Daten behandelt.

6.1.1 Wandergebiet Defereggental

Das Wandergebiet befindet sich in Osttirol, genauer gesagt im Zentrum von Osttirol und umfasst vorwiegend das Defereggental. Es liegt mitten im Nationalpark Hohe Tauern und wird bestens eingerahmt und umschlossen von Bergen des Defereggengebirges, der Rieserfernergruppe, der Lasörlinggruppe und der Schobergruppe. Der Schutz des Nationalparks Hohe Tauern macht das Defereggental zu einem der unberührtesten Hochgebirgstälern in den Alpen und ist somit ein geeignetes Ziel für Wandertouristen/Wandertouristinnen und natürlich für einheimische Wanderer/Wanderinnen¹². Das Wandergebiet umfasst dabei die drei Gemeinden Hopfgarten, St. Veit und St. Jakob, die im Defereggental gelegen sind. Hopfgarten im Defereggental ist auf 1107 *m* Seehöhe gelegen und befindet sich am Taleingang ganz im Osten. St. Veit im Defereggental liegt auf 1495 *m* Seehöhe und ist am Berghang zwischen den Gemeinden Hopfgarten und St. Jakob gelegen. St. Jakob im Defereggental liegt im Westen des Defereggentals auf 1398 *m* Seehöhe. Diese drei Gemeinden stellen wichtige Ausgangspunkte im Wandergebiet dar. Darüber hinaus finden Wanderer/Wanderinnen sehr viel Ruhe und intakte Natur im Wandergebiet Defereggental.

6.2 Geographische Daten

Geographische Daten werden oft als „spatial data“, „geospatial data“ oder „geographical information“ bezeichnet. Geographische Daten werden verwendet, um die reale Welt als eine digitale Karte zu modellieren. In den letzten Jahren hat die Popularität von digitalen

¹² Quelle: <http://www.defereggental.org/>

Karten stark zugenommen und sie sind für ökonomische und wissenschaftliche Analysen unumgänglich. Digitale Karten bieten relevante Informationen für eine große Menge von unterschiedlichen Anwendungen. Bei solchen Anwendungen werden digitale Karten nicht nur für die Visualisierung verwendet. Analysen, die auf geographischen Daten basieren, können durchgeführt werden. Hauptbereiche in diesem Zusammenhang sind zum Beispiel Planungsfragen (z.B. Raumplanung), Vorhersagen (z.B. Gefahrenzonenpläne für Hochwasser, usw.), Routenplanung oder die Überwachung verschiedenster Aufgaben. Heutzutage bieten viele Firmen digitale Karten für verschiedenste Anwendungen an. Die weitverbreitete Nutzung solcher Daten hat dazu geführt, dass Personen freiwillig Daten generieren und beispielsweise ihre Heimatgemeinde kartieren. Solche „volunteered geographic information“ (VGI) Plattformen sind weit verbreitet und können von jedem benutzt werden. Ein bedeutendes Beispiel ist hier die Open Source Datenbank „Open Street Map“¹³.

Darüber hinaus beschäftigt sich dieses Kapitel mit digitalen Karten für die Routenplanung, die Erschaffung eines Wanderwegenetzes und dabei spielt die Kombination unterschiedlicher Datensätze eine wichtige Rolle.

6.2.1 Modellierung der realen Welt

Die Oberfläche der Erde kann beschrieben werden, indem geographische Daten als digitale Karten abgebildet werden. Solch ein Modellierungsprozess ist immer eine Vereinfachung der realen Welt und bedeutet eine Reduktion von Komplexitäten (Bartelme, Geoinformatik: Modelle, Strukturen, Funktionen, 2005, S. 43). Das Ergebnis ist eine Abstraktion der realen Welt. Die Realität kann modelliert und abgebildet werden, indem Punkte, Linien und Polygone verwendet werden, um digitale Karten zu erstellen. Digitale Karten werden mit Hilfe ihrer Koordinaten, Topologie und Semantik in einer Datenbank mit spezifischen Datenstrukturen und Dateistrukturen am Computer gespeichert. Siehe Abbildung 27.

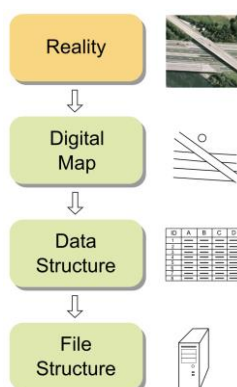


Abbildung 27: Flowchart eines Datenmodellierungsprozesses¹⁴

¹³ Quelle: www.openstreetmap.org

¹⁴ Bildquelle: (Pressl, 2012)

Digitale Karten

Die reale Welt kann als digitale Karte abgebildet werden und diese kann für verschiedenste Anwendungen verwendet werden. Im Vergleich zu analogen Karten sind digitale Karten flexibler, da sie bearbeitet werden können. Daten können leicht hinzugefügt und upgedatet werden und sie können sehr viel mehr Informationen beinhalten. Diese Informationen können von „Spatial-Analysefunktionen“ abgefragt und weiterverarbeitet werden. Digitale Karten können durch Vektordaten repräsentiert werden, wobei Entitäten durch die Verwendung von Punkten, Linien und Polygonen definiert werden (Bartelme, Geoinformatik: Modelle, Strukturen, Funktionen, 2005, S. 53). Durch die Definition von Entitäten und Relationen lassen sich Elemente gleicher Bauart, gleicher Eigenschaften und gleicher Beziehungsanforderungen zu Klassen zusammenfassen. Dem Konzept von Entitäten und Relationen (ER-Konzept oder engl. entity-relationship-approach) liegen folgende Komponenten zugrunde:

- Entitäten, die zu Entitätsklassen zusammengefasst sind
- Beziehungen (Relationen) sowohl zwischen Entitäten als auch zwischen Entitätsklassen
- Eigenschaften (Attribute) für Entitäten und Relationen

Eine Entität ist ein eindeutig identifizierbares, mit Eigenschaften ausgestattetes Ganzes, das sich dadurch unmissverständlich von anderen Entitäten abhebt.

Thematische Informationen werden als Attribute der Objekte gespeichert (siehe Abbildung 28). Darüber hinaus beschreiben Wechselbeziehungen zwischen Entitäten, wie die Entitäten miteinander verbunden sind. Diese Wechselbeziehungen werden als Topologie bezeichnet (siehe Kapitel 2.3.1).

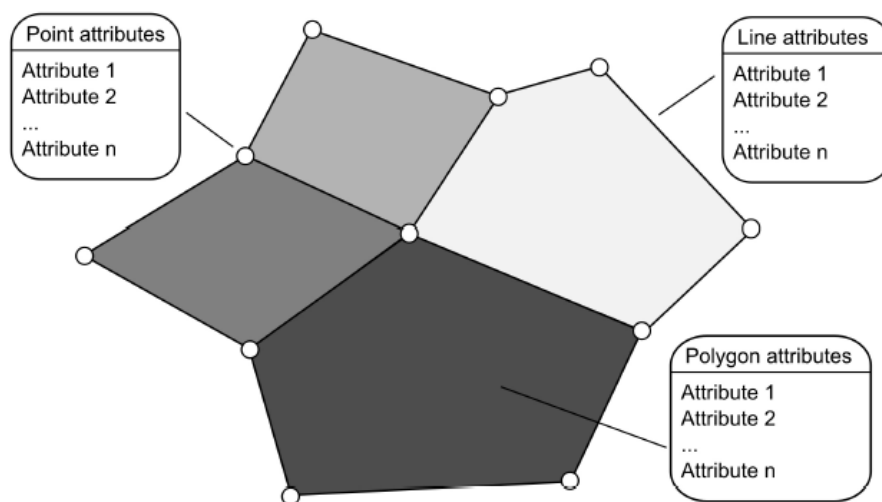


Abbildung 28: Geographische Vektordaten und deren Attribute¹⁵

¹⁵ Bildquelle: (Pressl, 2012)

Digitale Karten für die Routenplanung

Digitale Karten für die Routenplanung bestehen hauptsächlich aus Punkten und Linien, die einen Graph bilden und die dann als Knoten und Kanten bezeichnet werden (siehe Kapitel 2). In einer digitalen Wanderkarte für die Routenplanung werden die Wanderwege von Kanten und die Kreuzungspunkte der Wanderwege, die Start- und Endpunkte (z.B. Gipfelpunkt) und die Lage von Hütten oder anderen Sehenswürdigkeiten, von Knoten repräsentiert. Polygone können auch verwendet werden um beispielsweise Hütten abzubilden, dies wird aber in dieser Arbeit nicht angewendet. Ein topologisch korrekter Graph für die Routenplanung muss bestimmten Eigenschaften genügen. Solch ein Graph muss verbunden, gerichtet und bewertet sein (siehe Kapitel 2.3). In Abbildung 29 wird eine einfache Abstraktion eines solchen Graphen für ein Wanderwegenetz mit Knoten und Kanten und dazugehörigen Attributen dargestellt.

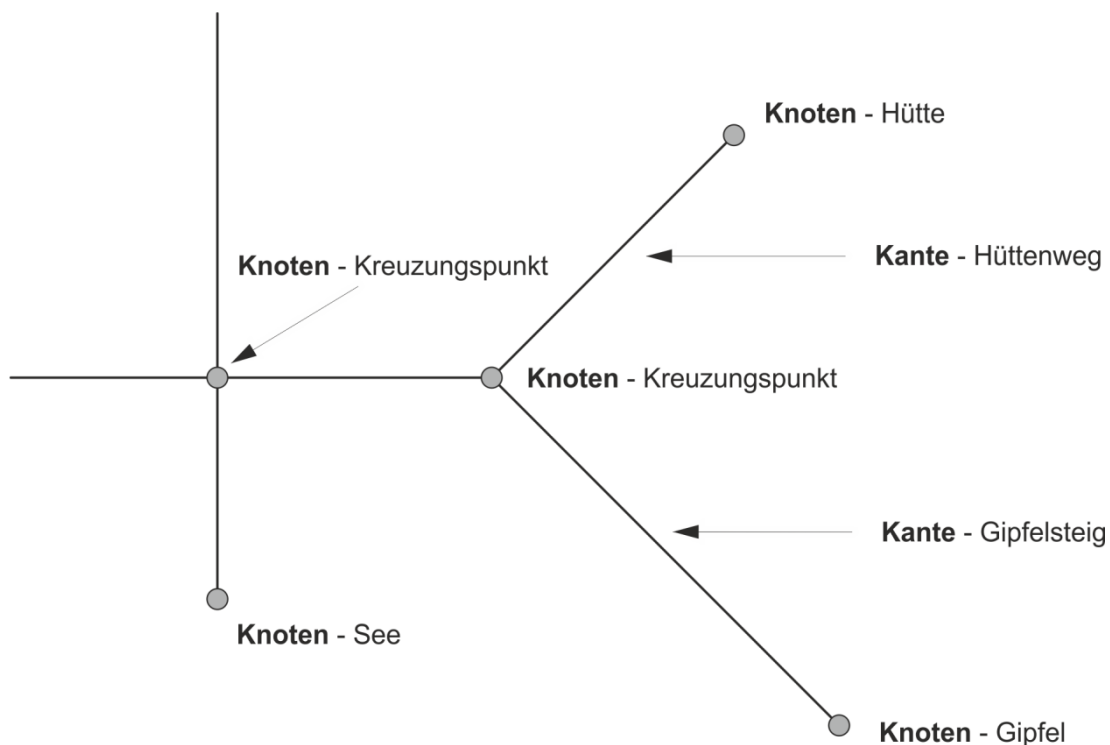


Abbildung 29: Geographische Daten für die Routenplanung inklusive Attributinformationen

6.2.2 Geographische Datenquellen

Es gibt heutzutage viele verschiedene Anbieter von digitalen Karten, die lokale und globale geographische Daten für verschiedenste Bereiche und Anwendungen anbieten. Wenn eine neue Anwendung, die auf geographischen Daten basiert, entwickelt wird, wird die Frage gestellt: „Welche digitale Karte unterstützt meine Anforderungen am besten?“ Die gleiche Frage stellt sich bezüglich der Entwicklung einer Routenplanung für

Wanderer/Wanderinnen. Auf der einen Seite ist es gut, wenn die geographischen Daten einen sehr großen Bereich oder sogar die ganze Welt abdecken, um den Service möglichst vielen Menschen zugänglich zu machen. Aber auf der anderen Seite wird dadurch die Genauigkeit geringer und es werden weniger detailliertere Informationen bereitgestellt, wie wenn es sich um lokale Daten und um ein spezifisches Themengebiet handelt. Im Bereich der Routenplanung für Wanderer/Wanderinnen sind detaillierte Informationen bezüglich der Wege wichtig. Zu beachten ist aber, dass eine lokale Lösung mit vielen spezifischen Informationen die globale Verfügbarkeit des Systems reduziert. Um dieses Problem zu beseitigen, können verschiedene Datenquellen in einer Anwendung kombiniert werden. Bezüglich der Datenkombination im entwickelten Routenplaner für Wanderer/Wanderinnen sind folgende Datenquellen zu nennen:

- Globale Daten: WMS Dienst (OpenStreetMap, World Imagery, World Topographic Map, World Street Map)
- Lokale Daten: Vermessene und digitalisierte Wanderwege in Osttirol (Koordinaten und Höheninformationen)
- Zusatzinformationen: POI`s (Points of Interest), Untergrundbeschaffenheit – weitere Informationen bezüglich Hütten, Gipfel, Sehenswürdigkeiten, usw.

6.3 Datenbasis

6.3.1 Datenbasis Wandergebiet Defereggental

Es wurden im Zuge von Vermessungsarbeiten vom Tourismusverband Osttirol Wanderwege in Osttirol aufgenommen, die als Grundlage für diese Arbeit dienen. Als Datenbasis stehen somit geometrisch definierte Punkte entlang dieser Wanderwegen zur Verfügung. Alle 5-20 Meter entlang der Wege wurde ein Punkt mit Koordinaten und Höhenwert aufgenommen. Es stehen zurzeit 76 vermessene Wanderwege in Osttirol zur Verfügung. Im Rahmen dieser Arbeit werden allerdings nur einige wenige dieser Wege verwendet, da nur ein paar dieser Wege im Defereggental liegen.

Nach der Untersuchung der vorliegenden Daten, die der Tourismusverband Osttirol bereitstellt, wurde allerdings festgestellt, dass die vorliegenden Daten bei weitem nicht ausreichen, um ein geeignetes Netzwerk mit einer Knoten-Kanten Struktur zu erstellen, da immer nur einzelne Wege vom Tal bis auf den Berg vorliegen. Für die Realisierung dieser Arbeit muss also ein Netz generiert werden. Es müssen daher Wanderwege digitalisiert werden, um ein möglichst gutes Netz erstellen zu können.

6.3.2 Datenbasis allgemein

Die Datenbasis stellt in jedem bearbeiteten Projekt eine besondere Rolle dar, da ausgehend davon wesentliche Aspekte mitbestimmt werden, wie die Genauigkeit, die Konsistenz, die Plausibilität, die Vollständigkeit und die Aussagekraft erhaltener Ergebnisse. Die Datenbasis dient also als Ausgangspunkt für die Umsetzung und für die erhaltenen Ergebnisse, wobei Ergebnisse immer nur so aussagekräftig sein können wie die Daten, mit denen diese Ergebnisse erzielt werden. Da die Wanderwege mit einem GNSS (Global Navigation Satellite System)-Receiver vermessen wurden, sind die erhaltenen Punkte als ellipsoidische Koordinaten im System WGS (World Geodetic System)-84 definiert. Das WGS-84 ist ein globales Koordinatensystem, das vor allem in der Satellitengeodäsie häufig verwendet wird.

6.4 Digitalisierung des Wandergebietes

Wie bereits erwähnt müssen Wege zuerst digitalisiert werden, um ein vollständiges Wanderwegenetz erstellen zu können. Die Digitalisierung erfolgt mit Hilfe der Open Source Möglichkeiten, die www.geoland.at anbietet, und analogen Wanderkarten des betreffenden Gebietes.

6.4.1 Open Source Service – Geoland.at

Die Open Source Plattform www.geoland.at ist ein Geodatenverbund der Bundesländer in Österreich. Dieser Geodatenverbund soll einen offenen und einfachen, österreichweiten Zugriff auf Geodaten und Services der Länder für unterschiedlichste Zwecke ermöglichen. Die Offenheit des Geodatenverbundes bietet überdies die Möglichkeit, mit verteilter Datenhaltung nach den Grundsätzen der Subsidiarität, auf Grundlage internationaler Standards (OGC, ISO, CEN) und nationaler Normen (ISO, EN, ON), weitere Geobasis- und Fachdaten einzubinden und zu verknüpfen. Der Benutzer kann somit über einen einfachen Webbrowser bundeslandübergreifend GIS-Daten abfragen, visualisieren und ausdrucken ¹⁶.

Durch die Kombination von Geodaten der einzelnen Bundesländer ergibt sich eine Grundkarte von Österreich die als „basemap.at“ bezeichnet wird.

¹⁶ Quelle: www.geoland.at

„basemap.at“

„basemap.at“ ist eine internetfähige Grundkarte von Österreich, basierend auf den Geodaten der Länder und deren Partnern, frei verfügbar und performant. Es gibt verschiedene Produktvarianten wie die Standardkarte, ein Orthofoto, eine Overlay Karte und einige andere. Die maximale Auflösung des Orthofotos beträgt 29 cm.

Die Karte deckt das gesamte österreichische Staatsgebiet homogen und flächendeckend ab und wird im 2-Monats-Rhythmus vollautomatisch aktualisiert. Die Karte und deren Geodatengrundlagen sind einem stetigen Aktualisierungsprozess unterworfen. Sowohl das kartographische Erscheinungsbild als auch die Geodatengrundlagen selbst werden laufend verbessert und aktualisiert. „basemap.at“ ist eine Rasterkarte in Form eines vorgenerierten Kachel-Caches, in der „Web Mercator Auxiliary Sphere“ und damit kompatibel zu den gängigen weltweiten Basiskarten wie beispielsweise jenen von OpenStreetMap, Google Maps und Bing Maps¹⁷.

Für den durchgeführten Digitalisierungsvorgang in dieser Arbeit wird auf „geoland.at“ und die Grundkarte von „basemap.at“ zurückgegriffen.

6.4.2 Digitalisierungsvorgang

Ähnlich der Datenstruktur der vorliegenden, vermessenen Wanderwege, wird alle 2-20 Meter ein Punkt, der Koordinaten und Höhenwerte enthält, digitalisiert. In den Abbildungen 27 bis 32 ist der Digitalisierungsvorgang abgebildet.



Abbildung 30: Ausschnitt aus "www.geoland.at" - Digitalisierung eines Weges (1)

¹⁷ Quelle: www.basemap.at



Abbildung 31: Ausschnitt aus "www.geoland.at" - Digitalisierung eines Weges (2)

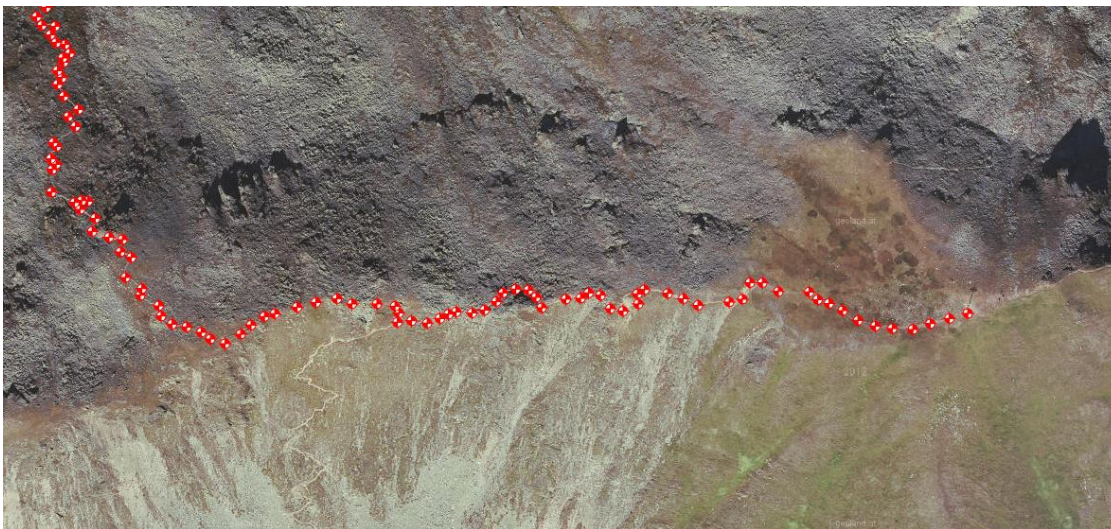


Abbildung 32: Ausschnitt aus "www.geoland.at" - Digitalisierung eines Weges (3)

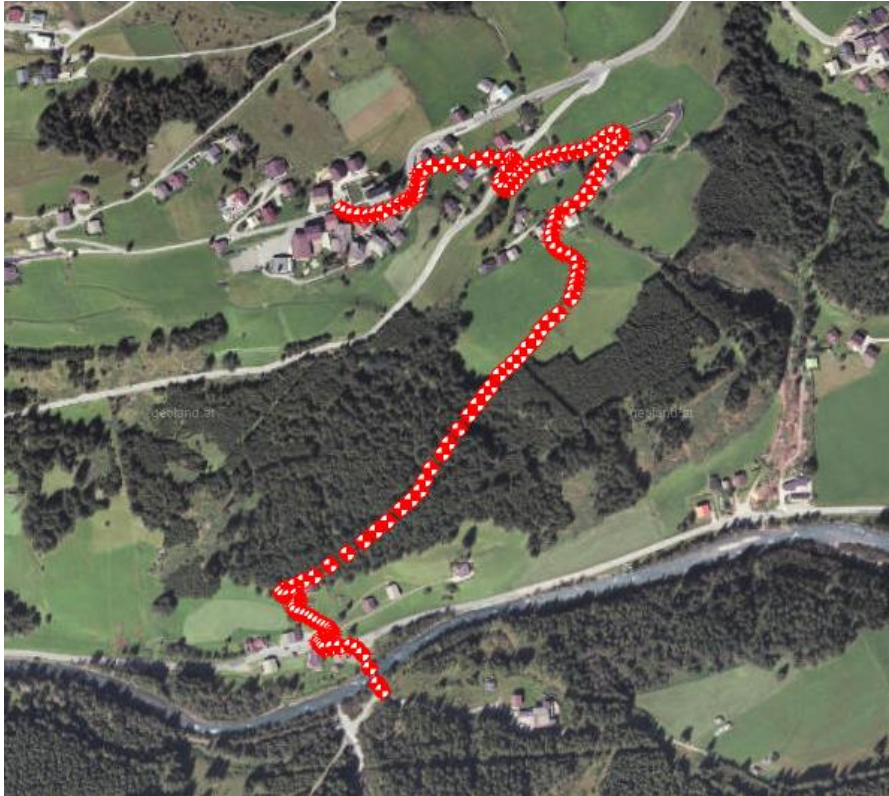


Abbildung 33: Ausschnitt aus "www.geoland.at" - Digitalisierung eines Weges (4)



Abbildung 34: Ausschnitt aus "www.geoland.at" - Digitalisierung eines Weges (Höhenwerte)

Die Wege werden so digitalisiert, wie die Kanten im Netzwerk liegen sollen. Es muss bei den Knoten (d.h. Anfangs- und Endpunkt des digitalisierten Weges) darauf geachtet werden, dass diese in ein und demselben Punkt ident sind. Das erstellte Netzwerk weist schlussendlich 486 Bögen (243 Kanten) auf, wobei nur 42 dieser Bögen aus den vom Tourismusverband Osttirol bereitgestellten Daten stammen (siehe Tabelle 7). Das heißt 444 Bögen resultieren aus dem Digitalisierungsvorgang. Insgesamt besteht das Netzwerk aus 13122 Polygonpunkten, wobei ca. 11000 davon manuell erfasst wurden.

Tabelle 7: Knoten- und Kantenanzahl des Wandernetzes

	Knoten	Bögen	Polygonpunkte
Datenanzahl vom Tourismusverband Osttirol	-	42	3122
Digitalisierte Datenanzahl	-	444	11000
Gesamtanzahl	174	486	13122

Die erzeugten Punkte liegen nach der Bearbeitung und Verspeicherung in einer Kantenstruktur im GK (Gauß-Krüger)-System vor. Allerdings nicht jene Daten, die vom Tourismusverband stammen. Deshalb muss auf ein einheitliches Koordinatensystem geachtet werden, damit die Erstellung des Netzwerkgraphen korrekt erfolgen kann. Die bereits vorliegenden Daten, die im WGS-84 Koordinatensystem sind, werden ins GK-System transformiert.

6.5 Koordinatentransformation

6.5.1 Geographische Koordinaten

Geographische Koordinaten werden in geographischer Breite und geographischer Länge angegeben. Die geographische Breite φ bezeichnet den Winkel zwischen der Ellipsoidnormalen durch einen Punkt P und der Äquatorebene (positiv nach Norden). Die geographische Länge λ ist definiert als Winkel zwischen der Meridianebene durch den Punkt und der Ebene des Nullmeridians (positiv nach Ost) (Hofmann-Wellenhof & Kienast, Bezugssysteme, 2009).

Abhängig davon, ob man globale oder lokale Koordinatensysteme definiert, liegen verschiedene Ellipsoide zu Grunde. Das WGS-84 (World Geodetic System 1984) ist ein häufig verwendetes Ellipsoid im Bezug auf globale Koordinatensysteme. Das Global Positioning Satellite System (GPS) zum Beispiel, nutzt dieses Ellipsoid als Basisellipsoid. Dem WGS-1984 liegt ein geozentrisches Niveauellipsoid, das durch die große Halbachse a , die geozentrische Gravitationskonstante GM , den normalisierten Kugelfunktionskoeffizienten $\bar{C}_{2,0}$ des Schwerepotentials und die mittlere Erdrotationsgeschwindigkeit ω definiert ist, zugrunde. Die Zahlenwerte sind wie folgt festgelegt:

$$a_{WGS} = 6\,378\,137\text{ m}$$

$$b_{WGS} = 6\,356\,752.31425\text{ m}$$

$$GM = 3\,986\,005 \cdot 10^8 \frac{\text{m}^3}{\text{s}^2}$$

$$\bar{C}_{2,0} = -484.166\,85 \cdot 10^{-6}$$

$$\omega = 7\,292\,115 \cdot 10^{-11} \text{ rad s}^{-1}$$

Die Parameter GM , $\bar{C}_{2,0}$ und ω sind physikalisch definiert und die Parameter a_{WGS} und b_{WGS} sind rein geometrisch definiert (Hofmann-Wellenhof & Kienast, Bezugssysteme, 2009).

Da man in der Landesvermessung üblicherweise im Gauß-Krüger System arbeitet und um die einzelnen Parameter zwischen den definierten Punkten zu berechnen, werden die definierten Punkte vom System WGS-84 in das Gauß-Krüger System transformiert. Das Bessel-Ellipsoid ist jenes, das im Zuge lokaler Koordinatensysteme in Österreich häufig benutzt wird. Als Ausgangsparameter dienen die folgenden Zahlenwerte:

$$a_{Bessel} = 6\,377\,397.155\,08\text{ m}$$

$$b_{Bessel} = 6\,356\,078.962\,90\text{ m}$$

Das Gauß-Krüger System beruht auf dem Bessel-Ellipsoid.

6.5.2 Transformation WGS 84 – Gauß-Krüger

Als erstes werden die ellipsoidischen (geographischen) Koordinaten in kartesische Koordinaten umgerechnet. Dazu findet eine Transformation der ellipsoidischen Koordinaten ins erdfeste Äquatorsystem statt. Man kann dem geozentrischen System X, Y, Z ein Ellipsoid (in diesem Fall das WGS-84 Ellipsoid) so zuordnen, dass der Mittelpunkt des Ellipsoids im Ursprung des erdfesten Äquatorsystems liegt, die Rotationsachse des Ellipsoids mit der Z-Achse des erdfesten Äquatorsystems und die Nullmeridianebene des Ellipsoids mit der Meridianebene von Greenwich zusammenfällt. Dann gelten folgende Transformationsformeln, die angewendet werden, um die Transformation durchzuführen:

$$X = \left(\frac{c}{V} + h \right) \cdot \cos \varphi \cdot \cos \lambda$$

$$Y = \left(\frac{c}{V} + h \right) \cdot \cos \varphi \cdot \sin \lambda$$

$$Z = \left(\frac{b}{V} + h \right) \cdot \sin \varphi$$

Mit Hilfe folgender Parameter:

$$c = \frac{a_{WGS}^2}{b_{WGS}}$$

$$V = \sqrt{1 + e'^2 \cdot \cos^2 \varphi}$$

$$e'^2 = \frac{a_{WGS}^2 - b_{WGS}^2}{b_{WGS}^2}$$

Anschließend erfolgt eine 7 Parameter Helmert-Transformation, um vom globalen ins lokale System zu gelangen. Mit Hilfe österreichweiter gültiger Parameter aus (Hofmann-Wellenhof & Kienast, Bezugssysteme, 2009) kann eine näherungsweise Transformation durchgeführt werden. Folgende Parameter und Formeln werden dazu verwendet:

$$\text{Verschiebung: } c_x = -577.33 \quad c_y = -90.13 \quad c_z = -463.92$$

$$\text{Verdrehung: } d\alpha_1 = +15.864^{cc} \quad d\alpha_2 = +4.537^{cc} \quad d\alpha_3 = +16.358^{cc}$$

$$\text{Maßstab: } d_\mu = -2.42 \cdot 10^{-6}$$

$$R = \begin{bmatrix} 1 & d\alpha_3 & -d\alpha_2 \\ -d\alpha_3 & 1 & d\alpha_1 \\ d\alpha_2 & -d\alpha_1 & 1 \end{bmatrix}$$

$$koord_{lokal} = c + d_\mu \cdot R \cdot koord_{global}$$

Dann müssen die lokalen kartesischen Koordinaten wieder in ellipsoidische Koordinaten umgerechnet werden. Dies ist die Umkehr der Transformation von oben, wo ellipsoidische Koordinaten ins erdfeste Äquatorsystem transformiert wurden. Mit folgenden Parametern und Formeln wird die Transformation realisiert:

$$\varphi = \arctan \left\{ \frac{Z + e'^2 \cdot b_{Bessel} \cdot \sin^3 \vartheta}{p - e'^2 \cdot a_{Bessel} \cdot \cos^3 \vartheta} \right\}$$

$$\lambda = \arctan \frac{Y}{X}$$

$$h = \frac{p}{\cos \varphi} - \frac{c}{V}$$

Mit Hilfe folgender Parameter:

$$c = \frac{a_{Bessel}^2}{b_{Bessel}}$$

$$p = \sqrt{X^2 + Y^2}$$

$$\vartheta = \arctan \left\{ \frac{Z \cdot a_{Bessel}}{p \cdot b_{Bessel}} \right\}$$

$$e'^2 = \frac{a_{Bessel}^2 - b_{Bessel}^2}{b_{Bessel}^2}$$

Schlussendlich erfolgt die GK-Abbildung vom Ellipsoid in die Ebene. Zuerst wird mit Hilfe von Hilfsgrößen und Koeffizienten des Bessel-Ellipsoids unter anderem die Meridianbogenlänge bestimmt und somit kann eine Abbildung auf die Ebene vollzogen werden. Dazu muss als erstes die Meridianbogenlänge $B(\varphi)$ bestimmt werden. Diese lässt sich mit Hilfe folgender Koeffizienten für das Besselellipsoid realisieren:

$$\alpha = 6366742.5203 \text{ m}$$

$$\beta = -2.51127456 \cdot 10^{-3}$$

$$\gamma = 2.62771 \cdot 10^{-6}$$

$$\delta = -3.42 \cdot 10^{-9}$$

$$B(\varphi) = \alpha \cdot [\varphi + \beta \cdot \sin 2\varphi + \gamma \cdot \sin 4\varphi + \delta \cdot \sin 6\varphi + \dots]$$

Mit folgenden Bezeichnungen (Hofmann-Wellenhof & Kienast, Bezugssysteme, 2009)

$B(\varphi)$ Meridianbogenlänge vom Äquator

$N = \frac{a^2}{b\sqrt{1+\eta^2}}$ Normalkrümmungsradius für φ

$\eta^2 = e'^2 \cdot \cos^2 \varphi$ Hilfsgröße

$e'^2 = \frac{a^2-b^2}{b^2}$ Zweite numerische Exzentrizität

$t = \tan \varphi$ Hilfsgröße

λ_0 Geographische Länge des Hauptmeridians

$l = \lambda - \lambda_0$ Längenunterschied zum Hauptmeridian

kann die Umrechnung von φ , λ in x , y durch die Formeln

$$\begin{aligned} x = & B(\varphi) + \frac{t}{2} N \cdot \cos^2 \varphi \cdot l^2 + \frac{t}{24} \cdot N \cdot \cos^4 \varphi \cdot (5 - t^2 + 9\eta^2 + 4\eta^2) \cdot l^4 \\ & + \frac{t}{720} \cdot N \cdot \cos^6 \varphi \cdot (61 - 58t^2 + t^4 + 270\eta^2 - 330t^2\eta^2) \cdot l^6 \\ & + \frac{t}{40320} \cdot N \cdot \cos^8 \varphi \cdot (1385 - 3111t^2 + 543t^4 - t^6) \cdot l^8 + \dots \end{aligned}$$

$$\begin{aligned}
y &= N \cdot \cos \varphi \cdot l + \frac{1}{6} \cdot N \cdot \cos^3 \varphi \cdot (1 - t^2 + \eta^2) \cdot l^3 \\
&+ \frac{1}{120} \cdot N \cdot \cos^5 \varphi \cdot (5 - 18t^2 + t^4 + 14\eta^2 - 58t^2\eta^2) \cdot l^5 \\
&+ \frac{1}{5040} \cdot N \cdot \cos^7 \varphi \cdot (61 - 479t^2 + 179t^4 - t^6) \cdot l^7 + \dots
\end{aligned}$$

erfolgen.

Die GK-Abbildung ist eine Zylinderprojektion, wobei das Ellipsoid zuerst auf einen berührenden oder schneidenden Zylinder abgebildet und dieser dann abgewickelt wird. Bei der GK-Abbildung berührt der Zylinder entlang eines Meridians (Grundmeridian oder Hauptmeridian), der längentreu abgebildet wird. Die Gauß-Krüger Projektion wird in Österreich, Deutschland und vielen anderen Ländern verwendet. Bei einer Sonderform der GK-Abbildung, der UTM-Abbildung (Universal Transverse Mercator) wird der Grundmeridian nicht längentreu, sondern mit dem Maßstab $m = 0.9996$ abgebildet. Die UTM-Projektion wird in vielen Ländern verwendet und ist das zukünftige Weltsystem (Hofmann-Wellenhof & Kienast, Bezugssysteme, 2009).

Die Gauß-Krüger Abbildung ist durch folgende Forderungen definiert:

- Konforme Abbildung des Ellipsoids in die Ebene
- Ein Meridian, der als Hauptmeridian bezeichnet wird, soll in die Abszissenachse des ebenen isothermen Koordinatensystems, das geodätisch orientiert ist, abgebildet werden
- Die Länge des Hauptmeridians soll bei der Abbildung in die Ebene erhalten bleiben; der Hauptmeridian soll also längentreu abgebildet werden

6.6 Genauigkeit der Daten

Die Datenbasis, die diesem Projekt zugrunde liegt, sind ellipsoidische Koordinaten und Höheninformationen von Punkten entlang von Wanderwegen. Es muss hier beachtet werden, dass entlang dieser Wege der Abstand der Punkte und auch die Genauigkeit nicht konstant sind. Einerseits gibt es Bereiche entlang der Wege, die nicht optimal gelegen sind, um GNSS-Punkte aufzunehmen, andererseits sind manuell aufgenommene Punkte entlang eines Weges immer mit menschlichen Einschätzungen behaftet, die das Ergebnis in weiterer Folge beeinflussen können. Bezüglich der ersten Aussage, dass es Bereiche gibt, die nicht optimal liegen, um GNSS-Punkte aufzunehmen, ist eine genauere Betrachtung dieser Tatsache notwendig. Signale die vom Satelliten an Empfängergeräten empfangen werden, können durch den sogenannten Mehrwegeeffekt abgelenkt werden (Hofmann-Wellenhof, Satellitengestützte Positionierung, 2010). Dies tritt vor allem auf, wenn sich Objekte in

unmittelbarer Nähe befinden, wie zum Beispiel hohe Bäume, Hänge oder Wände. Des Weiteren spielt natürlich die Satellitenkonstellation eine wichtige Rolle. Übergreifend auf den zweiten Aspekt, der oben genannt wurde, kann hier erwähnt werden, dass auch die zeitliche Dauer, wie lange man einen Punkt aufnimmt, Einfluss auf die Genauigkeit hat, wobei hier der menschliche Einfluss zum Tragen kommt. Tatsache ist allerdings, dass es sich hier um Wanderwege handelt und nicht um hochgenaue Vermessung, somit kann der Genauigkeit der aufgenommenen Punkte nicht unbedingt die höchste Beachtung geschenkt werden, da es nicht unbedingt eine Rolle spielt, ob innerhalb des Weges der Punkt ein paar Zentimeter rechts oder links liegt.

Darüber hinaus muss beachtet werden, dass nur ein geringer Teil der Knoten-Kanten Struktur, die in diesem Projekt entsteht, als Datenbasis durch die bereits vermessenen Wege abgedeckt wird. Der übrige Teil muss manuell digitalisiert werden. Dies wird über die Plattform www.geoland.at ermöglicht, wobei jeder einzelne Punkt von Hand digitalisiert werden muss und die Daten entsprechend weiterverarbeitet werden müssen. Durch dieses Vorgehen entstehen Unterschiede in der Konsistenz und Genauigkeit der beiden Datensätze. Da die Höhe von ein und demselben Punkt im System der vermessenen Wege mit dem System von www.geoland.at nicht exakt übereinstimmt, muss eine Anpassung erfolgen, damit möglichst konsistent gearbeitet werden kann. Dieser Aspekt kommt dadurch zustande, dass die vermessenen Wegpunkte GNSS-Höhen darstellen, die einerseits fehlerbehaftet sein können, andererseits liegt der Plattform www.geoland.at ein Gelände- und Höhenmodell zugrunde, das auch eine gewisse Ungenauigkeit aufweist. Dieses digitale Geländemodell wird aus „Airborne Laserscan“ Daten erstellt und besitzt ein Raster von $10\text{ m} \times 10\text{ m}$.

Um die Digitalisierung so exakt wie möglich durchführen zu können, werden analoge, hochgenaue Wanderkarten mit einem Maßstab von 1:30000 verwendet. Durch die Digitalisierung der Wege ergeben sich teilweise Ergebnisse der Höheninformation, die aufgrund der unterschiedlichen Systeme eine Differenz aufweisen, wobei darauf geachtet wird, dass beispielsweise die Gipfelpunkte den korrekten Höhenwert (aus analoger Wanderkarte) zugewiesen bekommen, da im digitalen Geländemodell Ungenauigkeiten vorhanden sein können.

6.7 Bestimmung der Parameter für die Bewertung der Kanten

Um ausgehend von geometrisch definierten Punkten ein Knoten-Kanten System, welches zu einem gerichteten Graphen werden soll, zu schaffen, müssen die Kanten zwischen den Knoten bewertet werden. Zum besseren Verständnis muss erwähnt werden, dass eigentlich die Bögen bewertet werden, da eine Kante in beide Richtungen begehbar ist und durch zwei Bögen dargestellt wird. Zur vereinfachten Formulierung wird nur mehr der Begriff Kante verwendet.

Für die Bewertung müssen Parameter bestimmt werden. Es werden zwischen zwei Polygonpunkten verschiedene Statistikparameter bestimmt, die wie folgt dargestellt werden:

- Distanz (horizontal)
- Höhenunterschied
- Schrägdistanz
- Richtungswinkel
- Steigung

Die Ermittlung der aufgelisteten Parameter lässt sich durch einfache mathematische Berechnungen vollziehen. Folgend werden diese beschrieben:

$$\text{Distanz (Horizontaldistanz): } d_{12} = \sqrt{\Delta x_{12}^2 + \Delta y_{12}^2}$$

$$\text{Höhenunterschied: } h_{12} = \Delta \text{höhe}$$

$$\text{Schrägdistanz: } s_{12} = \sqrt{d_{12}^2 + h_{12}^2}$$

$$\text{Richtungswinkel: } \tan v_{12} = \frac{\Delta y_{12}}{\Delta x_{12}}$$

$$\text{Steigung in Prozent: } \textit{steigung}_{12} = \left(\frac{h_{12}}{d_{12}}\right) \cdot 100$$

Diese Statistikparameter beziehen sich nun immer auf zwei Polygonpunkte innerhalb einer Kante. Nun sollen für eine Kante, Statistikparameter erschaffen werden. Dies führt dann zu folgenden Parametern:

- Weglänge (Horizontaldistanz)
- Weglänge (Schrägdistanz)
- Höhenunterschied
- Totaler Höhenunterschied
- Durchschnittliche Steigung
- Relative Steigung
- Durchschnittliche Höhe
- Durchschnittlicher Höhenunterschied

Folgend wird mathematisch beschrieben, wie man ausgehend von den oben aufgelisteten Parametern diese Kantenparameter bestimmt (die Kante besteht aus $n + 1$ Polygonpunkten (PP)):

Weglänge (Horizontaldistanz): $\sum_{PP}(d_{ij})$

Weglänge (Schrägdistanz): $\sum_{PP}(s_{ij})$

Höhenunterschied: $\sum_{PP}(h_{ij})$

Totaler Höhenunterschied: $\max(höhe) - \min(höhe)$

Durchschnittliche Steigung: $\frac{\sum_{PP}(steigung_{ij})}{n}$

Relative Steigung: $\left(\frac{Höhenunterschied}{Weglänge(hori)}\right) \cdot 100$

Durchschnittliche Höhe: $\frac{\sum_{i=1}^{n+1}(höhe)}{n}$

Durchschnittlicher Höhenunterschied: $\left(\frac{Höhenunterschied}{n}\right)$

Dies sind die berechneten Parameter, die für die Kantenbewertung herangezogen werden. Allerdings werden im weiteren Verlauf nur mehr 7 dieser weiterverwendet und zwar: *Weglänge (Horizontaldistanz)*, *Weglänge (Schrägdistanz)*, *Höhenunterschied*, *Totaler Höhenunterschied*, *Durchschnittliche Steigung*, *Relative Steigung*, *Durchschnittliche Höhe*.

Dies sind alles geometrische Kriterien, die in die Bewertung der Kanten und somit in die Routenplanung einfließen. Des Weiteren gibt es nicht-geometrische Kriterien, wie die Untergrundbeschaffenheit und POI's (Points of Interest). Die Untergrundbeschaffenheit hat einen wesentlichen Einfluss auf das Gehverhalten und somit auf die Gehzeit. Deshalb wird der Parameter Untergrundbeschaffenheit in dieser Arbeit eingeführt. Die Bestimmung des Untergrundes ist allerdings aufwendig, da die Untergrundbeschaffenheit manuell mit Hilfe von Orthofotos, Satellitenbildern und Fotos erfasst und dokumentiert werden muss. Das gesamte Wandergebiet (begehbare Wege in beide Richtungen) weist eine Weglänge von 667 km auf. Entlang des gesamten Wandernetzes werden die einzelnen Wegabschnitte, also die Kanten des Graphen, in Bereiche mit unterschiedlicher Untergrundbeschaffenheit eingeteilt. Dabei gibt es folgende Klassifizierung für den Untergrund (siehe Tabelle 8):

Tabelle 8: Parameter der Untergrundbeschaffenheit

<i>Straße</i>	<i>Weg</i>	<i>Pfad</i>
<ul style="list-style-type: none">• <i>Asphalt</i>	<ul style="list-style-type: none">• <i>Schotter</i>• <i>steinig</i>• <i>sandig/erdig</i>• <i>Wiese</i>	<ul style="list-style-type: none">• <i>Wald</i>• <i>steinig</i>• <i>Standard</i>• <i>am Fels</i>• <i>Wiese</i>

Eine Kante von A nach B wird bezüglich der Untergrundbeschaffenheit nach ihrer Zusammensetzung in Prozent betrachtet. Ein Beispiel könnte sein, dass die Kante von A nach B zu 20% aus einer Asphaltstraße, zu 30% aus einem Schotterweg und zu 50% aus einem Wiesenpfad besteht.

Als weiteres Kriterium für die Routenplanung werden POI`s eingeführt. Diese müssen ebenfalls manuell untersucht werden, um für jede Kante eine möglichst objektive Beurteilung nach verschiedensten Gesichtspunkten durchzuführen. Nachfolgend werden die möglichen POI`s aufgelistet:

- *Sehenswürdigkeit*
- *See*
- *schöner Weg*
- *schöne Aussicht*
- *ruhig*
- *über Waldgrenze*
- *Wald*
- *am Bach*
- *Almhütte*
- *am Grad entlang*
- *Brücke/über Bach*
- *kleine Bäume/leicht bewaldet*

Jede Kante wird mit Hilfe dieser POI`s klassifiziert.

Zusätzlich wird die Gehzeit als Kriterium eingeführt. Dieses Kriterium wird aus den vorhandenen Kriterien einer Kante berechnet. Für die Berechnung der Gehzeit fließen mehrere Faktoren ein. Der Begriff Marschzeitberechnung dient in Wanderkreisen als Gehzeitberechnung. Alpine Vereine verwenden heutzutage standardisierte Verfahren, um auch für weniger bergerfahrene Wanderer/Wanderinnen zu realistischen Gehzeit Angaben zu kommen. Diese Verfahren orientieren sich eher am langsamen Wanderer/an der langsamen Wanderin, da zu optimistisch angenommene Wegstundenzahlen dazu führen können, dass man sich in der tatsächlich benötigten Auf- oder Abstiegszeit verschätzt und in

die Dunkelheit oder in Bergnot gerät. Man rechnete früher mit etwa 4,2 km pro Stunde mit Zu- und Abschlügen je nach Steigung oder Gefälle. 2004 wurden diese Erfahrungswerte in die DIN (Deutsches Institut für Normung) 33466 *Wegweiser für Wanderwege* aufgenommen. Wie auch im *Wegehandbuch der Alpenvereine* gilt folgende Bemessungsgrundlage¹⁸:

Ein Wanderer/eine Wanderin legt in einer Stunde zurück:

- 300 Meter in Aufstieg
- 500 Meter in Abstieg
- 4 Kilometer Horizontale Entfernung

Die tatsächliche Gehzeit einer Strecke lässt sich errechnen, indem von den für Horizontal- und Vertikale Entfernung errechneten Zeiten der kleinere Wert halbiert und zum größeren addiert wird (siehe Tabelle 9).

Tabelle 9: Beispiel zur Berechnung der Gehzeit aus dem *Wegehandbuch der Alpenvereine*

Höhenunterschied: 900 m	$900\text{ m} / 300\text{ m} = 3\text{ h}$
Horizontale Entfernung: 8 km	$8\text{ km} / 4\text{ km} = 2\text{ h}$
Ansatz	$(2\text{ h}) / 2 + 3\text{ h} = 4\text{ h}$
Ergebnis	Die Gehzeit beträgt somit 4 Stunden.

Dieser Ansatz wird noch erweitert, da weitere Parameter aus der Statistikberechnung vorliegen und somit können diese eingerechnet werden. Als weitere Parameter zur Bestimmung der Gehzeit fließen noch die durchschnittliche Höhe, in der man sich bewegt, die Steigung und die Untergrundbeschaffenheit ein. Da die oben genannten Richtwerte sich eher am langsamen Wanderer/an der langsamen Wanderin orientieren und durch die zusätzliche Einführung von weiteren Parametern sich die Ergebnisse der Berechnungen eher vergrößern, werden diese Richtwerte leicht verändert, da dies realistischere Ergebnisse liefert. Somit werden folgende Richtwerte verwendet:

Ein Wanderer legt in einer Stunde zurück (veränderte Richtwerte) :

- 400 Meter in Aufstieg
- 650 Meter in Abstieg
- 4.5 Kilometer Horizontale Entfernung

Wie schon erwähnt werden darüber hinaus zusätzliche Parameter zur Berechnung eingeführt. Es wird die durchschnittliche Höhe eines Weges betrachtet und je nachdem in welcher Höhe man sich befindet, wird ein bestimmter Faktor eingerechnet. Das gleiche gilt für die Steigung und für den Untergrund. Je größer die Steigung desto größer der Faktor, der die Berechnung der Gehzeit beeinflusst. Zusätzlich spielt die Untergrundbeschaffenheit eine

¹⁸ Quelle: *Wegehandbuch der Alpenvereine*

Rolle. Mit Hilfe der folgenden Tabellen (Tabellen 10-12) und Formeln wird die Berechnung der Gehzeit näher beschrieben:

Tabelle 10: Bestimmung des Höhenfaktors für die Berechnung der Gehzeit

Höhe	Faktor Höhe
< 1500 m	1
1500 – 2500 m	1.05
2500 – 3000 m	1.15
> 3000 m	1.3

Tabelle 11: Bestimmung des Steigungsfaktors für die Berechnung der Gehzeit

Steigung	Faktor Steigung
< 15 %	1
15 – 25 %	1.05
25 – 35 %	1.1
35 – 45 %	1.2
45 – 55 %	1.3
> 55 %	1.4

Tabelle 12: Bestimmung des Untergrundfaktors für die Berechnung der Gehzeit

Untergrund	Faktor Untergrund
<i>Asphaltstraße</i>	1
<i>Schotterweg</i>	1.05
<i>steiniger Weg</i>	1.1
<i>sandiger/erdiger Weg</i>	1.2
<i>Wiesenweg</i>	1.05
<i>Waldpfad</i>	1.1
<i>steiniger Pfad</i>	1.15
<i>Standardpfad</i>	1.1
<i>Pfad am Fels</i>	1.35
<i>Wiesenzpfad</i>	1.1

Da eine Kante meistens aus mehreren Untergründen besteht, muss der Untergrundfaktor auf die einzelnen Abschnitte aufgerechnet werden. Beispielsweise besteht eine Kante zu 20 % aus einem Schotterweg, zu 35 % aus einem Waldpfad und zu 45 % aus einem steinigen Pfad. Somit ergibt sich für den endgültigen Faktor Untergrund dieser Beispielkante:

$$\text{Faktor Untergrund} = 0.2 \cdot 1.05 + 0.35 \cdot 1.1 + 0.45 \cdot 1.15 = 1.1125$$

Daraus ergibt sich folgende Formel für die Berechnung der Gehzeit:

$$a = \frac{\text{Höhenunterschied [m]}}{400 \text{ m}}$$

$$b = \frac{\text{Horizontalentfernung [m]}}{4500 \text{ m}}$$

Wenn $b < a$:

$$\text{Gehzeit} = \left(a + \frac{b}{2}\right) \cdot \text{Faktor Höhe} \cdot \text{Faktor Steigung} \cdot \text{Faktor Untergrund}$$

Wenn $a < b$:

$$\text{Gehzeit} = \left(b + \frac{a}{2}\right) \cdot \text{Faktor Höhe} \cdot \text{Faktor Steigung} \cdot \text{Faktor Untergrund}$$

Schlussendlich kann das Kriterium Gehzeit noch gerundet werden. Die Rundung der Kriterien hat einen wesentlichen Einfluss auf die erhaltenen Ergebnisse, da, je größer der Rundungsfaktor ist, desto wahrscheinlicher ist es, dass mehrere gleiche Ergebnisse produziert werden. Der Rundungsfaktor dieses und weiterer Kriterien wird in Kapitel 7.1, wenn es dann um die tatsächliche Anwendung der Kriterien geht, näher beschrieben.

6.8 Erstellen der Datenstruktur als Ausgangsbasis für die Routenplanung

Als Basis für die multikriterielle Routenplanung dient eine Datenstruktur, auf die der Algorithmus zugreift. Diese Datenstruktur muss alle notwendigen Informationen bezüglich der Knoten- und Kantenstruktur, der Bewertungen (Kriterien) und weitere Attributinformationen enthalten.

6.8.1 Indizierte Adjazenzliste

Wie bereits in Kapitel 2.4.2 erwähnt ist das Schema und die Funktionsweise einer indizierten Adjazenzliste gut geeignet, um eine Datenstruktur mit effizientem Zugriff zu erstellen. Eine indizierte Adjazenzliste besteht aus einer Knotenliste und einer Bogenliste. Die Knotenliste ist so gestaltet, dass mit Hilfe der Knoten ID und einem Index die abgehenden Bögen pro Knoten ermittelt werden können (siehe Tabelle 13). Der Aufbau der Bogenliste ist so gestaltet, dass die 486 verfügbaren Bögen des Wandernetzes mit Hilfe einer Bogen ID und dem Start- und Zielknoten zeilenweise gespeichert werden. In jeder dieser 486 Zeilen sind

neben der ID, dem Startknoten und dem Zielknoten noch die Statistikparameter (geometrische Kriterien), POI's, die Untergrundbeschaffenheit und die Attributinformationen enthalten (siehe Tabelle 14, 15).

Tabelle 13: Datenstruktur (indizierte Adjazenzliste / Knotenliste) - Ausgangsdaten für den Algorithmus

Knoten ID	1	2	3	4	5	...
Index	1	5	8	11	12	...
Abgehende Bögen	4	3	3	1

Tabelle 14: Datenstruktur (indizierte Adjazenzliste / Bogenliste) - Ausgangsdaten für den Algorithmus mit Gesamtstatistik

ID	Startkn.	Zielkn.	Statistik	POI	Untergrund	Attributinformation
1	1	152	[1114.1,8.2,9.8,...]	11	[0.25,0.5,...]	'Weg Bloßbrücke'
2	1	153	[1130.9,0.1,0.4,...]	1,8,11	[0,0.35,...]	'Dorfweg am Bach'
3	1	155	[1109.3,0.6,0.6,...]	1,8,12	[0,0.15,...]	'Damweg Plon'
4	1	162	[1261.1,13.6,15.1,...]	5,7	[0,1,...]	'Zwenewaldweg'
5	2	3	[2054.8,18.2,21.1,...]	2,3,5,6,9	[0,0,...]	'Geigenseesteig'
6	2	9	[1894.2,12.7,12.9,...]	5,11,12	[0,0,...]	'Gagenweg'
7	2	158	[1796.3,1.2,1.0,...]	9,12	[0,1,...]	'Weg Bloßhuetten'
8	3	2	[2054.8,-18.2,21.1,...]	2,3,5,6,9	[0,0,...]	'Geigenseesteig'
9	3	4	[2440.7,9.3,9.7,...]	2,3,4,5,6	[0,0,...]	'Pumperseesteig'
10	3	5	[2613.9,34.7,36.0,...]	2,3,4,5,6	[0,0,...]	'Regensteinpfad'
...

Tabelle 15: Datenstruktur - Ausgangsdaten für den Algorithmus mit Einzelstatistik

ID	Startkn.	Zielkn.	Statistik	POI	Untergrund	Attributinformation
1	1	152	27 × 9	11	[0.25,0.5,...]	'Weg Bloßbrücke'
2	1	153	61 × 9	1,8,11	[0,0.35,...]	'Dorfweg am Bach'
3	1	155	42 × 9	1,8,12	[0,0.15,...]	'Damweg Plon'
4	1	162	234 × 9	5,7	[0,1,...]	'Zwenewaldweg'
5	2	3	399 × 9	2,3,5,6,9	[0,0,...]	'Geigenseesteig'
6	2	9	189 × 9	5,11,12	[0,0,...]	'Gagenweg'
7	2	158	19 × 9	9,12	[0,1,...]	'Weg Bloßhuetten'
8	3	2	399 × 9	2,3,5,6,9	[0,0,...]	'Geigenseesteig'
9	3	4	69 × 9	2,3,4,5,6	[0,0,...]	'Pumperseesteig'
10	3	5	113 × 9	2,3,4,5,6	[0,0,...]	'Regensteinpfad'
...

Interessant sind hierbei die Spalten Statistik, POI und Untergrund, die die Kriterien darstellen. Es gibt, wie in den Tabellen 14 und 15 dargestellt, zwei Ausgangsdatensätze. Einmal enthält die Statistik (Tabelle 15) die einzelnen Bewertungen, für die Abschnitte zwischen den Polygonpunkten, einer Kante. Ein Eintrag in der Statistikspalte besteht dabei aus einer Matrix, die die einzelnen Werte und Bewertungen für die Polygonpunkte einer

Kante enthält. Enthalten ist die Distanz zwischen zwei Polygonpunkten (Horizontal- und Schrägdistanz), die Höhe der Polygonpunkte, der Höhenunterschied zwischen zwei Polygonpunkten, die Steigung zwischen zwei Polygonpunkten und die Koordinaten im GK-System und im WGS-84 System. Das andere Mal enthält die Statistik (Tabelle 14) die Gesamtstatistik für die Kante selbst. Im weiteren Verlauf wird mit der Gesamtstatistik der Kanten gerechnet. Betrachtet man diese Spalte nun genauer, so lässt sie sich mit einem Beispieleintrag folgendermaßen beschreiben (siehe Tabelle 16, 17).

Tabelle 16: Eintrag aus Statistikspalte in Ausgangsdatenstruktur (Zellen 1-5)

<i>Statistik [Zellen 1-5]</i>				
Ø Höhe [m]	Relative Steigung [%]	Ø Steigung [%]	Ø Höhenunterschied zw. Wegpunkten [m]	Ø Länge zw. Wegpunkten (schräg) [m]
1114.1	8.23	9.82	0.8	10.4

Tabelle 17: Eintrag aus Statistikspalte in Ausgangsdatenstruktur (Zellen 6-10)

<i>Statistik [Zellen 6-10]</i>				
Ø Länge zw. Wegpunkten (horizontal) [m]	Höhenunterschied [m]	Totaler Höhenunterschied [m]	Weglänge (schräg) [m]	Weglänge (horizontal) [m]
10.3	22	24	270.2	267.3

Das Kriterium Gehzeit wird erst beim Durchlaufen des Algorithmus berechnet und ist noch nicht in den Ausgangsdaten vorhanden. Des Weiteren sind die Spalten POI und Untergrund zu betrachten. In der Spalte POI stehen die Bewertungen der Wege nach der oben genannten Klassifizierung. Dabei stehen Zahlenwerte von 1-12 für die POI's zur Verfügung. Dies ist in Tabelle 18 definiert:

Tabelle 18: ID der POI's

ID	POI
1	<i>Sehenswürdigkeit</i>
2	<i>See</i>
3	<i>schöner Weg</i>
4	<i>schöne Aussicht</i>
5	<i>ruhig</i>
6	<i>über Waldgrenze</i>
7	<i>im Wald</i>
8	<i>am Bach</i>
9	<i>Almhütte entlang des Weges</i>
10	<i>Grad entlang</i>
11	<i>Brücke / überqueren eines Baches</i>
12	<i>leicht bewaldet</i>

In der Untergrundspalte stehen die prozentuellen Anteile der verschiedenen Untergründe von 1 bis 10. In Tabelle 19 sind diese aufgelistet und nummeriert. Nach dieser Nummerierung sind sie eingetragen.

Tabelle 19: ID des Untergrunds

ID	Untergrund
1	<i>Asphaltstraße</i>
2	<i>Schotterweg</i>
3	<i>steiniger Weg</i>
4	<i>sandiger / erdiger Weg</i>
5	<i>Wiesenweg</i>
6	<i>Waldpfad</i>
7	<i>steiniger Pfad</i>
8	<i>Standardpfad</i>
9	<i>Pfad am Fels</i>
10	<i>Wiesenzpfad</i>

6.9 Visualisierung des Wandergebietes

Als Datengrundlage für die Routenplanung mit multikriterieller Optimierung dient die in Kapitel 6.8.1 beschriebene Datenstruktur. Im weiteren Verlauf der Arbeit (ab Kapitel 7) wird die Anwendung des Algorithmus auf das erstellte Wandergebiet erläutert und diskutiert. Schlussendlich sollen die Ergebnisse, die der Algorithmus produziert, dargestellt werden. Diese Visualisierung erfolgt mit Hilfe des Programmes *Matlab R2016a*, mit dem auch die Umsetzung des Algorithmus realisiert wurde.

6.9.1 Visualisierung von geographischen Daten

Visualisierung ist der Prozess bei dem Informationen synoptisch, zum Zweck der Wiedererkennung, Kommunikation und um Muster und Strukturen zu erkennen, repräsentiert werden. Die Domäne der Visualisierung umfasst rechenbetonte, kognitive und mechanische Aspekte der Generierung, Organisation, Manipulation und Komprimierung solcher Repräsentationen. Repräsentationen können symbolisch, graphisch oder ikonisch gerendert werden und unterscheiden sich meistens von anderen Ausdrucksformen (Text, verbale Form, formelhaft) bezüglich des synoptischen Formats und der Gestalt (Bartelme & Rautz, Visualisierung von Geoinformation, 2014).

GIS und Visualisierung

Im Bereich GIS (Geographisches Informationssystem) und Visualisierung unterscheidet man ganz allgemein zwischen verschiedenen Modellen. Es gibt Geländehöhenmodelle (GM), Landschaftsmodelle (LM) und Kartographische Modelle (KM), um die wichtigsten zu nennen (Bartelme & Rautz, Visualisierung von Geoinformation, 2014).

Ein LM ist ein topographisches Modell, das verwendet wird, um beispielsweise Gewässer oder Verkehrsnetze darzustellen.

Ein GM, heutzutage spricht man meist von Digitalen Geländemodellen (DGM), beschreibt die Erdoberfläche ohne Vegetation und Bebauung, als eine in der Lage und Höhe bekannte Punktwolke. Diese liegt als regelmäßiges Gitter oder in Form von Laserpunkten vor. Verwendet werden DGM-Daten beispielsweise für die Berechnungen von Simulationen (Hochwasser, Lärmschutz, Lawinen, Funkwellenausbreitung), für Erdmassenberechnungen, zur Visualisierungen von Bauvorhaben, für die Kombination mit weiteren Geobasisdaten (z.B. Höhenlinien, digitales Orthofoto, digitale Karten) oder Fachdaten¹⁹.

Bezüglich KM kann gesagt werden, dass der hohe Informationsgehalt der staatlichen topographischen Landesaufnahme generalisiert und maßstabsabhängig, digital in Form von Kartographischen Modellen, geführt wird. Diese bilden die Grundlage für alle amtlichen kartographischen Produkte. Die kartographischen Modelle wurden ursprünglich durch Digitalisierung der analogen staatlichen Kartenwerke erstellt und seither in Form von Raster- und Vektormodellen weiterentwickelt. Kartographische Rastermodelle stellen den Karteninhalt in bildhafter Weise (quadratische Rasterzellen - Pixel) dar. Die Kartographischen Vektormodelle speichern die einzelnen Objekte in Form von Punkten, Linien oder Flächen und gewinnen durch die Zuordnung von verschiedenen Attributen an zusätzlichem Wert²⁰.

6.9.2 Übersichtskarte Wandernetz

Um einen Überblick über das Wandernetz zu erhalten, werden die Knoten und Kanten des Wandernetzes geplottet und dargestellt. In Abbildung 35 ist eine Übersichtskarte des Wandernetzes in Form von Knoten und Kanten dargestellt. Die grünen Punkte in der Abbildung symbolisieren Knotenpunkte. Diese Knotenpunkte können normale Kreuzungspunkte der Wanderwege, Hütten, Seen, Gipfel oder andere wichtige Punkte sein. Verbunden sind die Knoten durch Kanten (Wege), die durch schwarze Linien dargestellt sind. Das Wandergebiet erstreckt sich von 12.17° bis 12.52° in der Länge und von 46.85° bis 46.98° in der Breite. Bezogen sind diese Angaben auf geographische Koordinaten im System WGS-84. Flächenmäßig entspricht das einer Abdeckung von ca. 340 km².

¹⁹ Quelle: <http://www.ldbv.bayern.de/produkte/3dprodukte/gelaende.html>

²⁰ Quelle: <http://www.bev.gv.at>

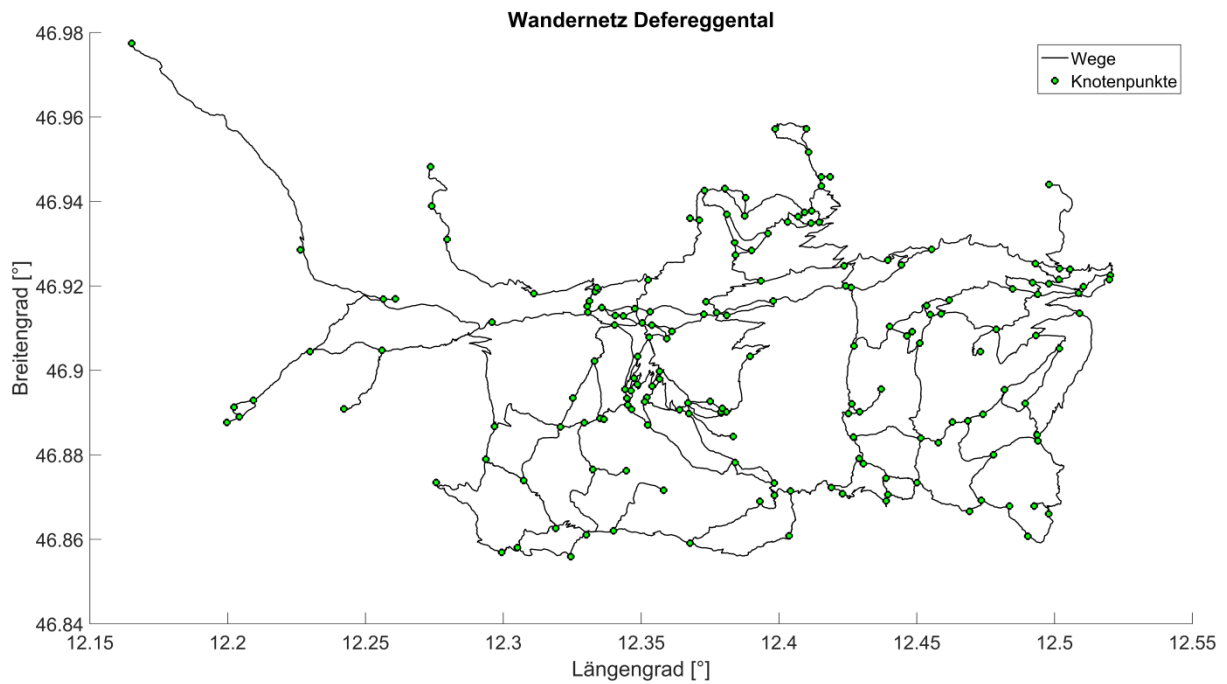


Abbildung 35: Übersichtskarte des Wandergebietes Defereggental

In Abbildung 36 ist eine Übersichtskarte mit einem Orthofotoausschnitt als Hintergrund dargestellt. Die Knoten sind wieder in grün und die Kanten (Wege) sind in blau abgebildet. In Abbildung 37 ist ebenfalls eine Übersichtskarte, die die Gemeinden des Defereggentals enthält, dargestellt. In Abbildung 38 ist die Lage des Wandergebietes innerhalb Osttirols abgebildet. In Abbildung 39 ist das Wandergebiet mit den Knotennummern dargestellt.

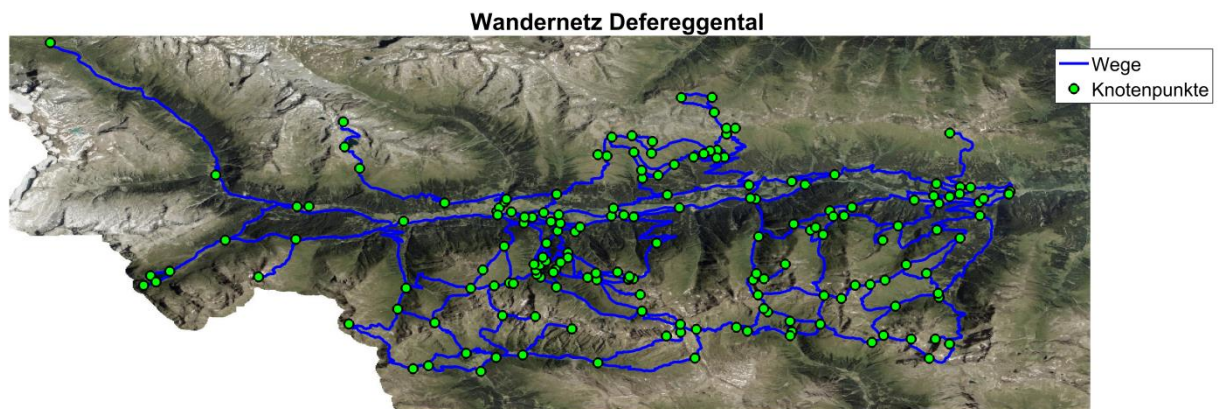


Abbildung 36: Übersichtskarte des Wandergebietes Defereggental (Orthofotoausschnitt als Hintergrund)

Wandernetz Defereggental

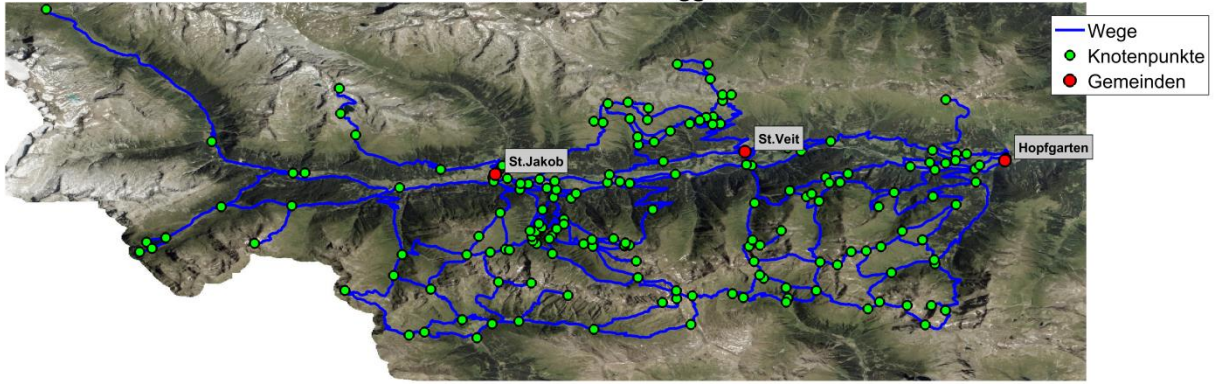


Abbildung 37: Übersichtskarte des Wandergebietes Defereggental mit Gemeinden (Orthofotoausschnitt als Hintergrund)

Wandernetz Defereggental

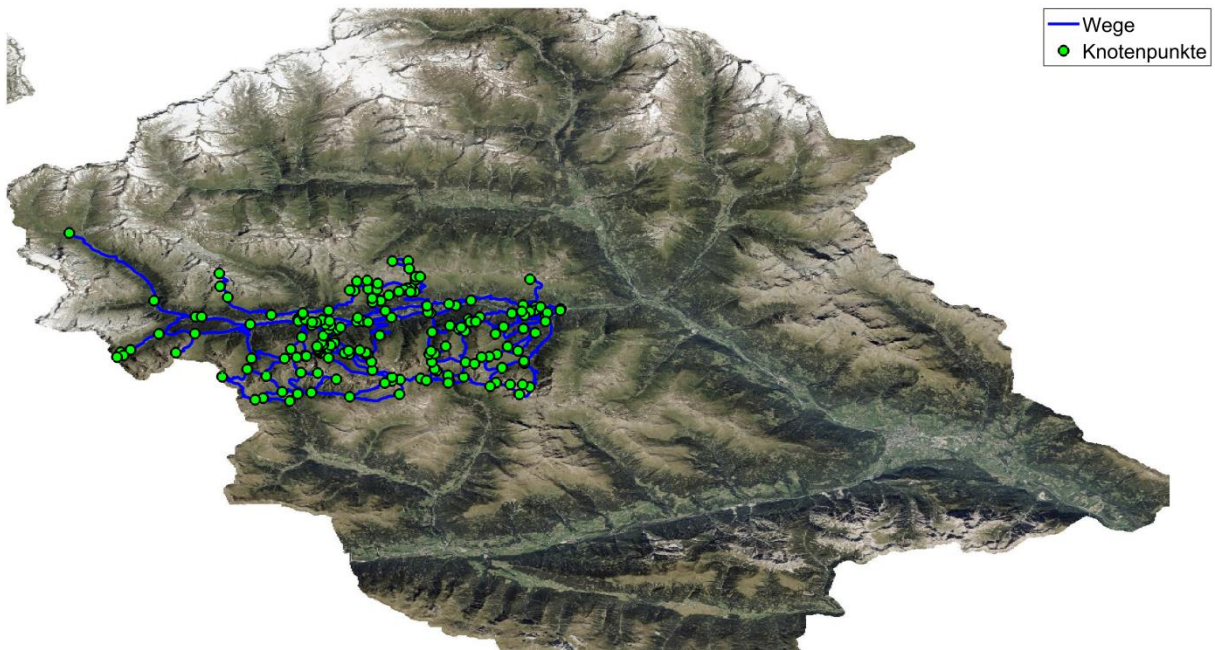


Abbildung 38: Lage des Wandergebietes innerhalb Osttirols

Wandernetz Defereggental

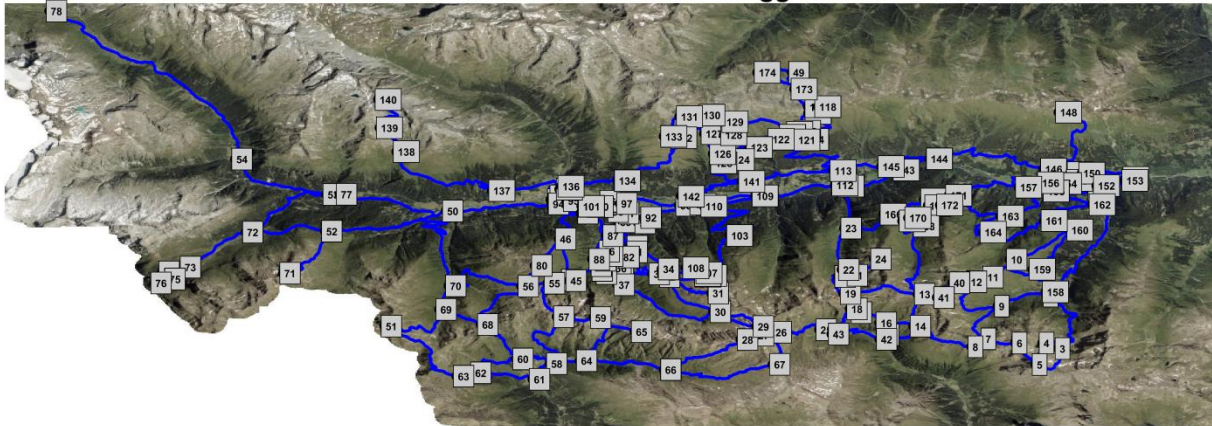


Abbildung 39: Übersichtskarte des Wandergebietes Defereggental mit Knotennummern

6.9.3 Web GIS

Ein Web-GIS ist ein GIS, dessen Funktionen teilweise auf Netzwerktechnologien (Internet, Intranet) basieren. Es kommunizieren mindestens zwei Rechner miteinander und tauschen Geodaten aus bzw. stellen GIS-Funktionalitäten bereit (Client-Server Prinzip). Die Client-Server Architektur schaut so aus, dass die Art des Netzzugangs über ein lokales Netz (Intranet) oder über das Internet stattfindet. Die Verbindung zwischen Client und Server wird über http (Hypertext Transfer Protocol) hergestellt. Ein Webserver überträgt Dokumente an Clients, wobei als Webserver Computer mit Websoftware oder nur die Websoftware bezeichnet werden. Es gibt verschiedene Webserver, freie, wie z.B. Apache²¹ oder auch proprietäre, wie z.B. Microsoft Internet Information Service (IIS)²². Ein Mapserver kann als Erweiterung eines Webserver gesehen werden und ist eine ausführbare Datei, die vom Webserver aufgerufen wird. Ein Weg-GIS besteht also zwingend aus einem Webserver und der Mapserver Erweiterung. Ein OGC konformer Mapserver wird als Web Map Server (WMS) bezeichnet (Bartelme, GIS Analysetechniken 1, 2014).

²¹ Quelle: www.apache.org

²² Quelle: www.iis.net

Web Map Server – Web Map Service

Die Client-Server Architektur eines WMS sieht so aus, dass durch das Senden eines HTTP Strings über einen Web Browser von einem Benutzer/einer Benutzerin, der die URL des Webservers und die aktuelle Anfrage enthält, ein Domain Nameserver (DNS) die URL dem richtigen Webserver zuordnet. Dann leitet der Webserver die Anfrage über CGI Schnittstellen an das Anwenderprogramm, den Mapserver, weiter. Der Mapserver wertet die Anfrage aus und erstellt eine Ergebniskarte. Der Mapserver kann dabei auf Geodaten verschiedenster Formatierungen zugreifen. Die Ergebniskarte wird dem Webserver übermittelt und der Webserver schickt diese, meist eingebettet in HTML (Hypertext Markup Language) oder in eine JAVA Applikation, an den Client. Dadurch ergeben sich verschiedenste Interaktionsmöglichkeiten bei der Verwendung der Karte. Das Navigieren in der Karte, das Hinzuladen verschiedener Ebenen durch Checkboxes und Funktionen zur Suchdatenabfrage, Objektselektion und Entfernungsmessungen werden ermöglicht. Unterstützt wird das ganze durch die AJAX (Asynchronous JavaScript und XML) Technologie. Das bedeutet, dass eine asynchrone Datenübertragung zwischen Server und Browser stattfindet, wobei der aktuelle Ausschnitt, die Zoomstufe und die Selektion bei einer Anfrage erhalten bleiben. Als Beispiele für solche Web Map Server können UMN Map Server, Degree, Mapnik, ArcGIS Server, Geoserver oder PostgreSQL/PostGIS genannt werden. Nach ISO (International Organization for Standardization) 19128 produziert ein Web Map Service dynamische Karten mit Hilfe georeferenzierter Daten. Dieser internationale Standard definiert eine Karte als eine Darstellung geographischer Informationen, als digitale Bilddatei, die für die Darstellung an einem Computerbildschirm geeignet ist. Eine Karte ist mehr als die dahintersteckenden Daten (Bartelme, GIS Analysetechniken 1, 2014).

Abbildung 40²³ zeigt den Verlauf wie ein Web Map Service (WMS) eine Karte aus vorliegenden Daten erzeugt.

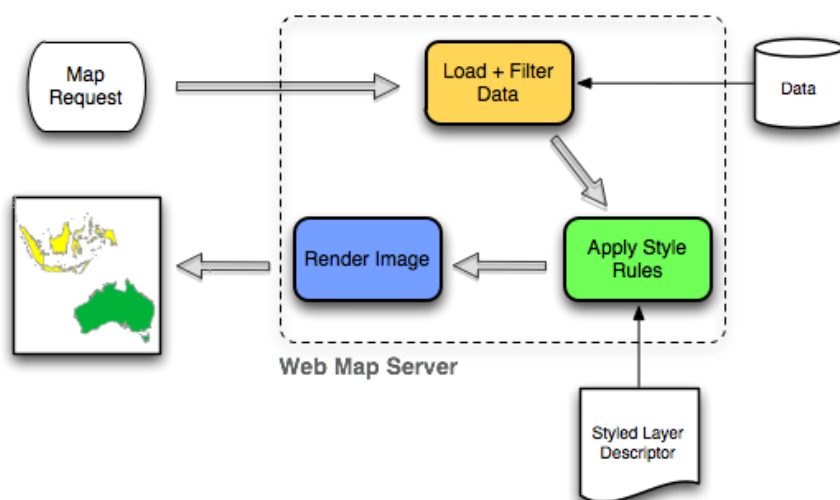


Abbildung 40: Ablauf eines WMS

²³ Bildquelle: http://presentations.openeo.org/2012_FOSSGIS/suiteintro/geoserver/wms.html

Im weiteren Verlauf der Arbeit wird ein WMS benutzt, um die Ergebnisse der multikriteriellen Routenplanung darzustellen. Die visualisierten Ergebnisse sind in Kapitel 8 dargestellt und beschrieben.

7 Anwendung des erweiterten Dijkstra Algorithmus auf das Wandergebiet

Wie bereits im Kapitel 4.5.1 beschrieben kann der erweiterte Dijkstra Algorithmus angewendet werden, um alle Pareto-optimalen Wege zu finden. Im Kapitel 4.5.1 wird der erweiterte Dijkstra Algorithmus auf einen Beispielgraph angewendet. Dieser erweiterte Algorithmus soll nun auf das erstellte Wandernetz (in Kapitel 6 beschrieben) angewendet werden. Die im Kapitel 6.8 erstellte Datenstruktur dient dabei als Ausgangsbasis. Es befinden sich 174 Knoten und 486 Bögen (243 Kanten) im Netzwerk. Im weiteren Verlauf werden nun die Kriterien, die einfließen, die Methoden der Optimierung und die Anfälligkeiten der Kriterien auf den Algorithmus beschrieben.

7.1 Kriterien für die multikriterielle Optimierung

Wie im Kapitel 6.7 beschrieben fließen geometrische und nicht-geometrische Kriterien in den Algorithmus ein. Folgend werden die geometrischen Kriterien, die in die Routenberechnung einfließen, aufgelistet:

- Weglänge (Horizontaldistanz)
- Weglänge (Schrägdistanz)
- Höhenunterschied
- Totaler Höhenunterschied
- Durchschnittliche Steigung
- Relative Steigung
- Durchschnittliche Höhe

Darüber hinaus fließen folgende nicht-geometrischen Kriterien in die Routenberechnung ein:

- Untergrundbeschaffenheit
- POI`s
- Gehzeit

Wie bereits im Kapitel 4.5.1 erwähnt enthält der Graph Kantenkostenvektoren und Labelvektoren in den Knoten. Die Kantenkostenvektoren enthalten die oben beschriebenen Kriterien und somit werden Labelvektoren in den Knoten erzeugt. Diese enthalten neben den Kriterien auch Informationen über den Vorgängerknoten, den betreffenden Labelvektor im Vorgängerknoten und die Position des Labelvektors des aktuellen Knoten. In Tabelle 20 wird der Aufbau der Labelvektoren dargestellt:

Tabelle 20: Aufbau eines Labelvektors des erstellten Algorithmus zur Routenplanung mit multikriterieller Optimierung

Gehzeit	Asphaltstraße	Sehenswürdigkeit
Weglänge (Horizontaldistanz)	Schotterweg	See
Höhenunterschied	steiniger Weg	schöner Weg
durchschnittliche Steigung	sandiger/erdiger Weg	schöne Aussicht
durchschnittliche Höhe	Wiesenweg	ruhig
relative Steigung	Waldpfad	über Waldgrenze
Weglänge (Schrägdistanz)	steiniger Pfad	im Wald
totaler Höhenunterschied	Standardpfad	am Bach
Vorgängerknoten	Pfad am Fels	Almhütte entlang des Weges
Labelnummer des Vorgängers	Wiesenpfad	Grad entlang
Labelnummer des aktuellen Knoten	-	Brücke/Überqueren eines Baches
-	-	leicht bewaldet

In der linken Spalte von Tabelle 20 befinden sich die Gehzeit und die restlichen geometrisch bestimmten Kriterien. In der mittleren Spalte werden die prozentuellen Anteile der Untergrundbeschaffenheit eingetragen. In der rechten Spalte wird die Anzahl der vorkommenden POI'S aufgelistet.

Interessant dabei ist, wie die Kriterien in den Labelvektoren berechnet werden. Normalerweise werden die einzelnen Kosten der Kanten aufsummiert, um den entsprechenden Wert im Labelvektor zu berechnen. Dies geschieht für Kriterien wie die Gehzeit, die für jede Kante einzeln bestimmt wird, für die Weglänge, für den Höhenunterschied und für die POI's. Um die durchschnittliche Höhe zu bestimmen wird der Wert aller betreffenden Kanten gemittelt. Um die Steigung zu bestimmen werden die Anteile der einzelnen Kanten betreffend der Weglänge betrachtet und die Steigungswerte so gemittelt, dass die Steigung betreffend des gesamten Weges korrekt ermittelt wird. Dies geschieht auch für die Untergrundbeschaffenheit. Beachtet werden muss allerdings, dass keine negativen Kriterien in den Algorithmus einfließen, sonst muss anstatt eines Label-setting ein Label-correcting Algorithmus angewendet werden. Der Höhenunterschied kann negative Werte einnehmen, da ein Weg vom Gipfel ins Tal führen kann. Dies wird so umgangen, indem der Höhenunterschied nur als einfacher Wert mitgeführt wird aber nicht als Kriterium ausgewählt wird. Wenn in weiterer Folge vom Höhenunterschied als Kriterium gesprochen wird, dann bezieht sich dies immer auf den totalen Höhenunterschied, der positiv angesetzt wird. Der Höhenunterschied ist der tatsächliche Höhenunterschied zwischen Startpunkt und Endpunkt einer Kante, wobei der totale Höhenunterschied der Unterschied des kleinsten und des größten Höhenwertes einer Kante ist. Das Problem bezüglich negativer Werte betrifft auch die Steigung. Bei der Steigung werden nur positive Werte angenommen. Wenn jetzt ein Weg eigentlich eine negative Steigung hätte, weil eine Kante absteigend verläuft, dann wird die Steigung als positiver Wert dargestellt. Aber aufgrund der Information des Höhenunterschiedes weiß der Algorithmus, dass die Kante absteigend verläuft und dadurch kann die Steigung wieder korrekt berechnet werden. Die relative Steigung arbeitet mit negativen Werten und dient so auch in gewisser Weise als

Vergleich zur durchschnittlichen Steigung. Wenn in weiterer Folge von der Steigung als Kriterium gesprochen wird, dann bezieht sich dies immer auf die durchschnittliche Steigung. Die relative Steigung bezieht sich immer auf die Gesamtlänge und den Gesamthöhenunterschied einer Kante, wobei sich die durchschnittliche Steigung auf die einzelnen Abschnitte innerhalb einer Kante bezieht.

Für einen Beispiel-Labelvektor des Knoten 150, während einer Berechnung des Algorithmus, sieht dies wie in Tabelle 21 aus:

Tabelle 21: Aufbau eines Beispiel-Labelvektors des Knoten 150 während der Berechnung des Algorithmus

Kriterien		Untergrund	[%]	POI`s	Anzahl
Gehzeit	1.5 h	Asphaltstraße	2	Sehenswürdigkeit	0
Weglänge (horizontal)	3.046 km	Schotterweg	4	See	0
Höhenunterschied	318 m	steiniger Weg	0	schöner Weg	0
durchschnittliche Steigung	12.5 %	sandiger/erdiger Weg	45	schöne Aussicht	0
durchschnittliche Höhe	1284 m	Wiesenweg	34	ruhig	2
relative Steigung	11.8 %	Waldpfad	0	über Waldgrenze	0
Weglänge (schräg)	3.123 km	steiniger Pfad	0	im Wald	2
totaler Höhenunterschied	320 m	Standardpfad	0	am Bach	0
Vorgängerknoten	149	Pfad am Fels	0	Almhütte entlang des Weges	0
Labelnummer des Vorgängers	1	Wiesenpfad	15	Grad entlang	0
Labelnummer des aktuellen Knoten	2	-		Brücke/Überqueren eines Baches	1
-		-		leicht bewaldet	3

7.2 Methoden der Optimierung

Wie in Kapitel 4.2 beschrieben werden durch die Einführung von Effizienz und Dominanz die besten Pareto-optimalen Lösungen gefunden. Mit Hilfe von Skalarisierungsmethoden oder anderen Methoden können Pareto-optimale Lösungen in polynomischer Laufzeit ermittelt werden, da nur eine Teilmenge aller Lösungen, durch die Anwendung einer Approximation, produziert wird.

7.2.1 Anwendung von Skalarisierungsmethoden auf den Algorithmus

Die WSM (Weighted Sum Method) ist eine Skalarisierungsmethode, bei der das multikriterielle Problem in ein „single-criterion“ Problem, indem positive Gewichte an den

Kriterien angebracht werden, transformiert wird. Dies wurde für das Wandernetz mit verschiedenen Einstellungen bezüglich der Gewichte getestet. Allerdings sind die Ergebnisse nicht wirklich aussagekräftig, da je nachdem mit welchen beliebig gewählten Gewichten gerechnet wird, unterschiedlichste Ergebnisse entstehen. Darüber hinaus wird durch die Verwendung solcher Skalarisierungs Methoden wieder von der multikriteriellen Optimierung abgewichen und es handelt sich nur mehr um eine „single-criterion“ Optimierung. Da das Hauptaugenmerk dieser Arbeit auf der multikriteriellen Optimierung liegt, wird die WSM nicht weiter behandelt. In weiterer Folge wird auf andere „Nicht-Skalarisierungs“ Methoden zurückgegriffen, um aussagekräftige Ergebnisse zu erzielen.

7.2.2 Anwendung von Nicht-Skalarisierungsmethoden auf den Algorithmus

Wie in Kapitel 4.3.2 beschrieben ist die lexikographische Ordnung eine geeignete Methode um multikriterielle Problemstellungen zu lösen. Dabei wird eine Reihenfolge bezüglich der Kriterien festgelegt, das heißt es gibt ein Hauptkriterium und mindestens ein Nebenkriterium. Bezüglich des erstellten Wandernetzes und der erzeugten Kriterien bedeutet das, dass wenn die lexikographische Ordnung für die Optimierung verwendet wird, ein Kriterium aus den oben genannten (geometrische Kriterien, Gehzeit, Untergrund, POI's) als Hauptkriterium gesetzt wird. Daneben können Nebenkriterien in unterschiedlichster Reihenfolge festgelegt werden. Dies ist entscheidend für den Algorithmus, da während dieser durchläuft, jedes Mal der kleinste Labelvektor aus T gefunden und ausgewählt werden muss. Der kleinste Labelvektor ist somit der, der lexikographisch gesehen am kleinsten ist. In weiterer Folge vergleicht der Algorithmus die Labelvektoren in den Knoten und streicht Lösungen, die dominiert werden. Wenn man das Beispiel vom Kapitel 4.5.1 betrachtet, dann sieht man, dass bei diesem alle Pareto-optimalen Wege berechnet werden. Es werden dabei die Lösungen (Labelvektoren) in den Knoten verglichen und mit Hilfe der Definition der Dominanz (4.2.2) wird bestimmt, ob eine Lösung gestrichen wird oder nicht. Darüber hinaus könnte man auch die Definition der strikten Dominanz (4.2.3) anwenden. In weiterer Folge werden nun verschiedene Varianten, die einerseits versuchen alle Pareto-optimalen Lösungen und andererseits eine Teilmenge aller Lösungen oder Einzellösungen zu bestimmen, beschrieben und diskutiert.

Wie bereits erwähnt kann die Suche nach allen Pareto-optimalen Lösungen im Normalfall nicht in polynomischer Zeit abgeschlossen werden. Deshalb kann eine Begrenzung der Laufzeit, die Kombination von Einzellösungen (Kapitel 5.3.1) oder die Dominanz mit Begrenzung (Kapitel 5.3.2) eingeführt werden, um eine Teilmenge aller Lösungen zu produzieren.

Darüber hinaus wird in dieser Arbeit versucht, die Dominanz mit Hilfe der lexikographischen Optimalität zu bestimmen, so dass, ähnlich wie bei der Bestimmung von Einzellösungen, lexikographisch dominierte Lösungen gestrichen werden. Dabei werden die Labelvektoren,

wenn sie lexikographisch dominiert werden, aus T gestrichen, allerdings bleiben sie in P wenn sie in P aufgenommen wurden (siehe Punkt 3). Dies kann dazu führen, dass (schwache) Pareto-optimale Lösungen oder alternative Lösungen produziert werden.

Eine weitere Möglichkeit eine Teilmenge aller Lösungen zu bestimmen ist jene, dass die Definition der Dominanz mit Hilfe der lexikographischen Optimalität verändert werden kann, (siehe Punkt 4) um auch (schwache) Pareto-optimale Lösungen zu erhalten. Dabei werden im Vergleich zu Punkt 3 tatsächlich korrekte Lösungen (nach Definition) erhalten.

Wenn man eine Einzellösung produzieren will, so kann dies wie im Kapitel 5.2.1 durch die Einführung der lexikographische Optimalität erreicht werden. So wird jene Lösung gestrichen, die lexikographisch gesehen dominiert wird. In weiterer Folge werden nun verschiedene Varianten und die dazugehörigen Ergebnisse diskutiert.

Für alle weiteren Fälle und Ergebnisse dieses Kapitels wird die Gehzeit auf eine Viertelstunde gerundet. Diese Annahme gilt solange bis diese aufgehoben wird.

Punkt 1: Bestimmung einer Einzellösung mit lexikographischer Optimalität für 2 Kriterien (Gehzeit, durchschnittliche Steigung), Startpunkt 1 und Zielpunkt 8 im Wandernetz

Es wird die lexikographische Ordnung für die Suche nach dem kleinsten Labelvektor aus T verwendet. Weiteres wird die lexikographische Ordnung auch verwendet, um die Labelvektoren in den Knoten zu bestimmen. Dazu wird die Gehzeit (auf eine Viertelstunde gerundet) und die durchschnittliche Steigung als Nebenkriterium verwendet. Definition 6 aus dem Kapitel 4.3.2 beschreibt das lexikographische Minimum für zwei Kriterien:

$$(f_1, f_2) <_{lex} (f'_1, f'_2) \text{ if } (f_1 < f'_1) \text{ or } (f_1 = f'_1 \text{ and } f_2 < f'_2)$$

Die Routenberechnung produziert somit folgendes Ergebnis vom Startknoten 1 bis zum Zielknoten 8:

Tabelle 22: Ergebnis Einzellösung

Gehzeit [h]	durchschnittliche Steigung [%]
5.75	14

Punkt 2: Bestimmung einer Teilmenge aller Lösungen durch Kombination von Einzellösungen mit lexikographischer Optimalität für 2 Kriterien (durchschnittliche Steigung, Gehzeit), Startpunkt 1 und Zielpunkt 8 im Wandernetz

Die Vorgehensweise ist dieselbe wie oben beschrieben (Punkt 1), außer, dass sich die Reihenfolge der Kriterien ändert (Hauptkriterium: durchschnittliche Steigung, Nebenkriterium: Gehzeit). Nun entsteht eine zusätzliche Lösung, die durch Kombination mit der oben genannten Lösung (Unterschied: Priorität der Kriterien) wie folgt aussieht:

Tabelle 23: Ergebnis einer Teilmenge (1)

Gehzeit [h]	durchschnittliche Steigung [%]
5.75	14
7	8

Der zweite Eintrag weist eine geringere Steigung aber eine längere Gehzeit auf.

Punkt 3: Bestimmung einer Teilmenge aller Lösungen durch Dominanz mit lexikographischer Begrenzung für 2 Kriterien (Gehzeit, durchschnittliche Steigung), Startpunkt 1 und Zielpunkt 164 im Wandernetz

Es wird die lexikographische Ordnung für die Suche nach dem kleinsten Labelvektor aus T angewendet. Weiteres wird die lexikographische Ordnung auch verwendet, um zu bestimmen, ob ein Labelvektor in einem Knoten dominiert wird oder nicht. Dazu werden die Gehzeit (auf eine Viertelstunde gerundet) als Hauptkriterium und die durchschnittliche Steigung als Nebenkriterium verwendet. Definition 6 aus dem Kapitel 4.3.2 beschreibt das lexikographische Minimum für zwei Kriterien:

$$(f_1, f_2) <_{lex} (f'_1, f'_2) \text{ if } (f_1 < f'_1) \text{ or } (f_1 = f'_1 \text{ and } f_2 < f'_2)$$

Mit Hilfe dieser Definition werden im Laufe der Berechnungen Lösungen, die dominiert werden, gestrichen. Darüber hinaus werden Lösungen, die in P aufgenommen werden, behalten, wodurch Ergebnisse entstehen, die nicht unbedingt der Definition des lexikographischen Minimums mit der eingestellten Kriterienreihenfolge genügen. Allerdings entstehen dadurch weitere effiziente Lösungen. Folgend wird das Ergebnis zum Zielknoten dargestellt:

Tabelle 24: Ergebnis einer Teilmenge (2)

Gehzeit [h]	durchschnittliche Steigung [%]
4.5	19
4.5	13

Wie man sieht wird die erste Lösung laut Definition 6 eigentlich lexikographisch dominiert. Da die erste Lösung allerdings als erstes in P aufgenommen wurde, bleibt sie in P . Somit sind eine schwache und eine strikte Pareto-optimale Lösung entstanden.

Punkt 4: Bestimmung einer Teilmenge aller Lösungen durch Dominanz mit veränderter lexikographischer Begrenzung für 2 Kriterien (Gehzeit, durchschnittliche Steigung), Startpunkt 1 und Zielpunkt 164 im Wandernetz

Es wird die lexikographische Ordnung für die Suche nach dem kleinsten Labelvektor aus T laut Definition 6 angewendet. Weiteres wird die lexikographische Ordnung auch verwendet,

um zu bestimmen, ob ein Labelvektor in einem Knoten dominiert wird oder nicht, allerdings wird hierzu die Definition geändert.

Definition 10. *Das veränderte lexikographische Minimum für einen Zwei-Kriterien Fall ist wie folgt definiert:*

$$(f_1, f_2) <_{lex} (f'_1, f'_2) \text{ if } (f_1 < f'_1)$$

Somit bleiben Lösungen, bei denen das Hauptkriterium gleich groß ist, erhalten. Somit können (schwache) Pareto-optimale Lösungen als Ergebnisse produziert werden. Die Gehzeit (auf eine Viertelstunde gerundet) und die durchschnittliche Steigung werden wieder als Kriterien verwendet. Folgend wird das Ergebnis dargestellt:

Tabelle 25: Ergebnis einer Teilmenge (3)

Gehzeit [h]	durchschnittliche Steigung [%]
4.5	19
4.5	13

Wie man sieht, erhält man das gleiche Ergebnis wie oben (Punkt 3).

Punkt 5: Bestimmung aller Pareto-optimalen Lösungen für 2 Kriterien (Gehzeit, totaler Höhenunterschied), Startpunkt 1 und Zielpunkt 169 im Wandernetz

Es wird die lexikographische Ordnung für die Suche nach dem kleinsten Labelvektor aus T laut Definition 6 angewendet. Nun werden nur mehr ineffiziente Lösungen in den Knoten gestrichen, um den Suchraum zu minimieren (wie beim Beispiel im Kapitel 4.5.1). Das Streichen ineffizienter Lösungen erfolgt nach den Definitionen 2 bis 4 (Kapitel 4.2.2).

Folgende Ergebnisse werden produziert:

Tabelle 26: Ergebnis aller Pareto-optimalen Lösungen

Gehzeit [h]	durchschnittliche Steigung [%]
4.5	9.4
4.5	9.2
4.75	9.1

Wie bereits erwähnt können alle Pareto-optimalen Lösungen, ohne gewisse Einschränkungen, normalerweise nicht in polynomischer Laufzeit berechnet werden. Nach der Untersuchung der Ergebnisse, die der entwickelte Algorithmus zur multikriteriellen Optimierung für die Routenplanung in dem erstellten Wandernetz produziert, kann gesagt werden, dass eine Berechnung in einer angebrachten Berechnungszeit nur für gewisse Einstellungen und Varianten möglich ist. Möglich ist die Berechnung für die oben genannten Fälle (Punkt 1 bis Punkt 4). Darüber hinaus können auch mehrere Kriterien in die Berechnung einfließen, die Berechnungszeit erhöht sich zwar, aber die Berechnung ist in polynomischer Zeit möglich. Für die Berechnung aller Pareto-optimalen Lösungen wie im Punkt 5 gilt, dass

dies nur mit gewissen Einstellungen (Reihenfolge und Auswahl der Kriterien) möglich ist. Welche Kriterien geeignet sind, welche Kriterienreihenfolge dabei verwendet wird und welche Ergebnisse damit erzielt werden, wird im nächsten Kapitel (7.3) genauer beschrieben.

7.3 Anfälligkeiten des Algorithmus bezüglich der Kriterien

Der verwendete Algorithmus erzeugt je nach Variante, das heißt welche Optimierungsmethode und welche Kriterien in welcher Reihenfolge verwendet werden, unterschiedliche Ergebnisse. Ziel dieser Arbeit ist es unter anderem herauszufinden, welche Optimierungsmethoden und welche Kriterien sinnvolle und interessante Ergebnisse liefern. Darauf wird nun im weiteren Verlauf dieses Kapitels eingegangen. Zuerst, sei noch erwähnt, dass, wie oben genannt (Punkt 5), darauf geachtet werden muss, welche Varianten überhaupt Ergebnisse produzieren.

7.3.1 Anfälligkeiten des Algorithmus bezüglich der Kriterien für die Berechnung einer Einzellösung

Wie in den Unterpunkten 1 bis 5 oben genannt, wird mit Hilfe der lexikographischen Optimalität laut Definition 6 der kleinste Labelvektor aus T ausgewählt. Dieser Vorgang bleibt nun für die weiteren beschriebenen Varianten und Fälle bestehen. Die Bestimmung der Labelvektoren erfolgt ebenfalls mit Hilfe der lexikographischen Optimalität. Es werden nun unterschiedliche Kriterien als Hauptkriterium und Nebenkriterium verwendet und die Ergebnisse untersucht. Als Startpunkt wird der Knoten 103 verwendet.

Gehzeit als Hauptkriterium

Die Gehzeit ist ein geeignetes Hauptkriterium, da die Zeitkomponente mit zunehmenden Wegen größer wird. Somit lässt sich im Normalfall ausschließen, dass der Algorithmus zwischen zwei Knoten hin und her rechnet. Da Wege in beide Richtungen begehbar sind, ist dies theoretisch möglich. Obwohl der Algorithmus als Label-setting Algorithmus definiert ist, können, da es sich um den erweiterten Dijkstra Algorithmus handelt, mehrere Label in einem Knoten vorhanden sein. Somit kann es sein, dass der Algorithmus zwischen zwei Knoten hin und her rechnet, wenn iterativ gesehen, jeweils ein Label zweier betreffender Knoten das kleinste ist. Durch das Setzen der Gehzeit als Hauptkriterium wird die schnellste Route berechnet. Welches der anderen Kriterien als Nebenkriterium verwendet wird spielt nur eine untergeordnete Rolle, da sich an den Ergebnissen nur dann etwas ändert, wenn die Gehzeit zweier Lösungen ident ist und der Algorithmus dann das zweite Kriterium betrachtet. Als

Beispiel wird die Lösung zum Zielpunkt 34 betrachtet, wobei als Nebenkriterium die Weglänge (horizontal) angenommen wird.

Tabelle 27: Ergebnis (Labelvektor) einer Einzellösung (Gehzeit, Weglänge (horizontal)) - 103 bis 34

2.25	0	0
3546.9	0	1
431	0	1
9.3	0.05	2
2348.2	0	3
7.5	0.15	2
3651.4	0.20	0
597	0.35	0
105	0	2
1	0.25	0
1	-	1
-	-	1

Die Gehzeit beträgt 2.25 *h*. Die Weglänge (horizontal) beträgt 3546.9 *m*. Die Weglänge (schräg) beträgt 3651.4 *m*.

Weglänge als Hauptkriterium

Die Weglänge ist ebenfalls ein geeignetes Hauptkriterium. Hier wird der kürzeste Weg berechnet. Im Vergleich zur Gehzeit als Hauptkriterium ändern sich die Ergebnisse zwar teilweise, aber die Unterschiede sind gering. Im Zielknoten 34 lässt sich ein Unterschied ausmachen und das Ergebnis sieht wie folgt aus:

Tabelle 28: Ergebnis (Labelvektor) einer Einzellösung (Weglänge (schräg), Weglänge (horizontal)) - 103 bis 34

2.5	0	0
3403.5	0.09	1
431	0	1
9.3	0	3
2379.2	0	4
9.0	0.17	3
3508.9	0.29	0
589	0.28	0
106	0	2
1	0.17	0
2	-	1
-	-	1

Die Gehzeit beträgt 2.5 *h*. Die Weglänge (horizontal) beträgt 3403.5 *m*. Die Weglänge (schräg) beträgt 3508.9 *m*. Der Unterschied zum vorherigen Ergebnis lässt sich durch die unterschiedliche Reihenfolge der Kriterien erklären.

Höhenunterschied als Hauptkriterium

Setzt man den totalen Höhenunterschied als Hauptkriterium an, so ergibt sich zum Knoten 34 dieselbe Lösung, wie in Tabelle 28 dargestellt. Die Gehzeit beträgt 2.5 h. Die Weglänge (horizontal) beträgt 3403.5 m. Die Weglänge (schräg) beträgt 3508.9 m. Der totale Höhenunterschied beträgt 589 m. Der Höhenunterschied ist ebenfalls ein geeignetes Hauptkriterium. Hier wird die Lösung mit dem geringsten Höhenunterschied dargestellt.

Steigung als Hauptkriterium

Die Steigung ist als Hauptkriterium nicht geeignet. Das Problem ist, wenn die Steigung das Hauptkriterium ist, dass es Wege mit sehr geringer Steigung gibt, die dem Algorithmus Probleme bereiten. Stellen wir uns vor, die Steigung zwischen zwei Knoten v_2 und v_3 beträgt 0.5 % und die Steigung von v_2 zu allen anderen Nachbarknoten beträgt für jeden Nachbarknoten über 5 %. Die Wege sind in beide Richtungen begehbar. Wenn der Algorithmus nun durchläuft, dann rechnet er sehr lange zwischen den beiden Knoten v_2 und v_3 , da sich die berechnete Steigung um 0 einpendelt und sich nicht großartig ändert und einer der beiden Knoten immer das kleinste Label besitzt.

Gewisser Untergrund als Hauptkriterium

Die Untergrundbeschaffenheit ist im Normalfall auch nicht als Hauptkriterium geeignet, da es viele Wege gibt, die beispielsweise keine Asphaltstraße beinhalten. Darüber hinaus kann es auch wieder sein, dass ein Weg zu 99 % Prozent aus einer Asphaltstraße besteht und der Algorithmus wieder zwischen zwei Knoten hin und her rechnet.

Gewisser POI als Hauptkriterium

POI's sind als Hauptkriterium im Normalfall nicht geeignet, da die Anzahl an POI's betrachtet wird und je größer die Anzahl von einem gewissen POI für einen Weg desto besser. Somit kann es sein, dass der Algorithmus zwischen ein paar Knoten hin und her rechnet und sich die Anzahl der POI'S immer weiter erhöht.

7.3.2 Anfälligkeiten des Algorithmus bezüglich der Kriterien für die Berechnung einer Teilmenge aller Lösungen durch Dominanz mit veränderter lexikographischer Begrenzung

Die Vorgehensweise ist dieselbe wie oben erwähnt (vierter Unterpunkt in 7.2.2). Das Streichen dominierter Lösungen erfolgt mit Hilfe der veränderten lexikographischen Optimalität. Es werden nun wieder unterschiedliche Kriterien als Hauptkriterium und Nebenkriterium verwendet und die Ergebnisse untersucht. Als Startpunkt wird der Knoten 103 verwendet.

Die Anfälligkeiten des Algorithmus bezüglich der Kriterien verändern sich nicht. Durch die Einführung der Dominanz mit veränderter lexikographischer Begrenzung wird eine Teilmenge aller Lösungen produziert, wobei die Auswahl der Kriterien als Hauptkriterium und Nebenkriterium die gleichen Auswirkungen hat wie oben erwähnt (7.3.1). Der Unterschied ist nur, dass jetzt mehrere Lösungen entstehen können.

7.3.3 Anfälligkeiten des Algorithmus bezüglich der Kriterien für die Berechnung aller Pareto-optimalen Lösungen

Nur die Auswahl bestimmter Kriterien führt zu einer Lösung in polynomischer Laufzeit. Es werden alle Varianten, die zwei Kriterien im Optimierungsverlauf berücksichtigen, genauer betrachtet. Beispielsweise werden die Gehzeit und die Weglänge und die Gehzeit und die Steigung als Kriterien betrachtet. Als Startpunkt wird wieder der Knoten 103 verwendet.

Gehzeit und Weglänge als Kriterien

Wählt man die Gehzeit und die Weglänge als die beiden relevanten Kriterien aus, so produziert der Algorithmus alle Pareto-optimalen Lösungen in einer angemessenen Laufzeit.

Gehzeit und Steigung als Kriterien

Es kann keine Berechnung in polynomischer Laufzeit durchgeführt werden.

Um alle weiteren Fälle übersichtlicher darzustellen, werden die möglichen Kombinationen in der Tabelle 29 aufgelistet:

Tabelle 29: Kombinationen von Kriterien, ob alle Pareto-optimalen Wege berechnet werden können

	Gehzeit	Weglänge	Höhenunterschied	Steigung	Untergrund	POI
Gehzeit	-	ja	ja	nein	teilweise	teil.
Weglänge	ja	-	ja	nein	teilweise	teil.
Höhenunterschied	ja	ja	-	nein	teilweise	teil.
Steigung	nein	nein	nein	-	nein	nein
Untergrund	teilweise	teilweise	teilweise	nein	-	nein
POI	teilweise	teilweise	teilweise	nein	nein	-

Somit kann festgehalten werden, dass alle Pareto-optimalen Lösungen nur bei der Verwendung der Kombinationen Gehzeit & Weglänge, Gehzeit & Höhenunterschied und Weglänge & Höhenunterschied berechnet werden können.

7.3.4 Auswirkungen von gerundeten Kriterien auf die Ergebnisse des Algorithmus

Je nachdem wie die Werte der Kriterien gerundet werden, werden unterschiedliche Ergebnisse produziert. Als Standardannahme galt bisher, die Gehzeit wird auf eine Viertelstunde gerundet, die Weglänge, der Höhenunterschied und die durchschnittliche Höhe werden auf den Dezimeter genau angegeben und die Steigung und Untergrundbeschaffenheiten werden auf die erste Nachkommastelle von Prozentzahlen gerundet. Um die Auswirkungen dieser Annahmen zu untersuchen, werden diese Annahmen leicht verändert und die Ergebnisse betrachtet.

Als erstes wird die Rundung der Gehzeit genauer betrachtet. Als Beispiel wird die Bestimmung aller Pareto-optimalen Wege mit den Kriterien Gehzeit und Weglänge durchgeführt. Nachfolgend werden die unterschiedlichen Ergebnisse vom Knoten 103 zum Zielknoten 88 dargestellt und diskutiert.

Tabelle 30: Auswirkung der Rundung der Gehzeit auf die Ergebnisse

	Rundung auf Achtelstunde	Rundung auf Viertelstunde	Rundung auf halbe Stunde
Anzahl an effizienten Lösungen	6	11	Algorithmus bricht ab, zu viele Lösungen

Je höher der Rundungsfaktor, desto mehr Lösungen werden produziert.

Nun wird die Rundung der Weglänge genauer betrachtet. Als Beispiel wird die Bestimmung einer Teilmenge durch Dominanz mit veränderter lexikographischer Begrenzung, mit den Kriterien Weglänge und dem POI „Almhütte“, untersucht. Nachfolgend werden die unterschiedlichen Ergebnisse vom Knoten 103 zum Zielknoten 88 dargestellt und diskutiert.

Tabelle 31: Auswirkung der Rundung der Weglänge auf die Ergebnisse

	Rundung auf Meter	Rundung auf 100 Meter	Rundung auf 1 Kilometer
Anzahl an effizienten Lösungen	1	2	4

Das bedeutet, je höher der Rundungsfaktor, desto mehr Lösungen werden produziert. Dies gilt für alle Kriterien gleichermaßen. Für alle weiteren dargestellten Ergebnisse wird folgende Annahme getroffen:

Die Gehzeit wird auf eine Viertelstunde gerundet. Die Weglänge wird auf 100 Meter gerundet. Der Höhenunterschied wird auf 1 Meter gerundet. Die Steigung und die

Untergrundbeschaffenheit werden auf ganze Prozentzahlen gerundet. Die durchschnittliche Höhe wird auf 100 Meter gerundet.

7.3.5 Anfälligkeiten des Algorithmus bezüglich der Kriterien für mehr als zwei Kriterien

Es können auch drei oder vier Kriterien in den Routenplanungsalgorithmus aufgenommen werden. Dies erfolgt durch eine einfache Erweiterung der Komponenten im Algorithmus. An den oben beschriebenen Anfälligkeiten auf den Algorithmus ändert sich dabei nichts. Interessant hierbei ist wie ineffiziente Lösungen gestrichen werden. Es können weiterhin Einzellösungen produziert werden, indem, falls beide Kriterien den gleichen Wert aufweisen, einfach das dritte oder vierte Kriterium herangezogen wird, um die beste Lösung zu erhalten. Auch bei der Bestimmung einer Teilmenge aller Lösungen wird das dritte und eventuell vierte Kriterium, beim Streichen ineffizienter Lösungen, mitberücksichtigt. Bei der Bestimmung aller Pareto-optimalen Lösungen werden dann, wenn es um das Streichen ineffizienter Lösungen geht, alle Kriterien gleichermaßen behandelt. Definition 2 wird angewendet, um ineffiziente Lösungen zu streichen.

7.3.6 Anzahl an effizienten Lösungen

Je nachdem welche Optimierungsmethode gewählt wird, welche Kriterien festgelegt werden und welcher Startknoten im Wandernetz ausgewählt wird, produziert der Algorithmus unterschiedliche Lösungen. Interessant ist, wie viele Lösungen durchschnittlich pro Zielknoten entstehen. Dies wurde für einige Fälle untersucht und die Ergebnisse werden nachfolgend betrachtet.

Bestimmung einer Teilmenge aller Ergebnisse durch die Verwendung der veränderten lexikographischen Optimalität

Als Startknoten wird der Knoten 100 ausgewählt. Das Hauptkriterium ist die Gehzeit und das Nebenkriterium wird mit der Weglänge (Schrägdistanz) festgelegt.

Durchschnittlich werden 4.1 effiziente Lösungen pro Zielknoten produziert. In Abbildung 41 ist ein Ausschnitt aus *Matlab R2016a* dargestellt. In der Matrix *effizientewege* werden die optimalen Lösungen für jeden Knoten gespeichert. Die Anzahl der Lösungen (Anzahl der Zeilen mit Eintrag) ist für jeden Knoten (Spalte) dargestellt.

	22	23	24	25	26	27	28	29	30	31	32
1	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double
2	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double
3	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double
4	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double			
5				14x3 double	14x3 double	14x3 double	14x3 double	14x3 double			
6				14x3 double	14x3 double	14x3 double	14x3 double	14x3 double			
7											
8											
9											
10											
11											
12											
13											
14											
15											
16											
17											

Abbildung 41: Ausschnitt aus Matlab – Anzahl effizienter Wege (Startknoten 100, Gehzeit & Weglänge, Approximation)

Nun wird wieder der Knoten 100 als Startknoten ausgewählt, dieser bleibt für die restlichen Fälle bestehen. Das Hauptkriterium ist nun die Weglänge (Schrägdistanz) und die Gehzeit wird als Nebenkriterium festgelegt. Somit ändern sich die Ergebnisse. Durchschnittlich werden nun 1.7 Lösungen pro Zielknoten produziert. Dies ist dadurch zu erklären, dass die Weglänge auf 100 Meter gerundet, weniger idente Lösungen produziert, als dies bei der Gehzeit als Hauptkriterium der Fall ist. In Abbildung 42 ist die Anzahl der effizienten Lösungen für einige Zielknoten dargestellt.

	22	23	24	25	26	27	28	29	30	31	32
1	14x3 double	14x3 double	14x3 double		14x3 double	14x3 double	14x3 double		14x3 double	14x3 double	14x3 double
2					14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double
3					14x3 double	14x3 double	14x3 double		14x3 double	14x3 double	14x3 double
4				14x3 double				14x3 double			
5											14x3 double
6								14x3 double			14x3 double
7											
8											
9											
10											
11											
12											
13											
14											
15											
16											
17											

Abbildung 42: Ausschnitt aus Matlab – Anzahl effizienter Wege (Startknoten 100, Weglänge & Gehzeit, Approximation)

Wählt man den totalen Höhenunterschied als Hauptkriterium und die Weglänge (Schrägdistanz) als Nebenkriterium, so erhält man durchschnittlich 1.2 Lösungen pro

Zielknoten. Der Höhenunterschied wird auf 1 Meter gerundet, somit ist es logisch, dass weniger effiziente Lösungen produziert werden.

Bestimmung aller Pareto-optimalen Lösungen

Als Startknoten wird wieder der Knoten 100 ausgewählt. Die Kriterien werden mit der Gehzeit und der Weglänge (Schrägdistanz) festgelegt.

Nun werden durchschnittlich 6.1 effiziente Lösungen pro Zielknoten produziert. In Abbildung 43 ist wieder ein Ausschnitt der erhaltenen Lösungen dargestellt.

	22	23	24	25	26	27	28	29	30	31	32	
1	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double
2	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double
3	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double
4	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double
5	[]	[]	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double
6	[]	[]	[]	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double
7	[]	[]	[]	[]	[]	[]	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double
8	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]
9	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]
10	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]
11	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]
12	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]
13	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]
14	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]
15	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]
16	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]
17	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]

Abbildung 43: Ausschnitt aus Matlab – Anzahl effizienter Wege (Startknoten 100, Gehzeit & Weglänge, alle Lösungen)

Wählt man die Gehzeit und den totalen Höhenunterschied so entstehen 9.2 Lösungen pro Zielknoten (Abbildung 44).

	22	23	24	25	26	27	28	29	30	31	32	
1	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	[]	14x3 double	14x3 double	14x3 double	[]
2	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	[]
3	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	[]	14x3 double	14x3 double	14x3 double	[]
4	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	[]
5	[]	[]	[]	14x3 double	14x3 double	14x3 double	14x3 double	[]	14x3 double	14x3 double	14x3 double	14x3 double
6	[]	[]	[]	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	[]	14x3 double	14x3 double	14x3 double
7	[]	[]	[]	14x3 double	14x3 double	14x3 double	14x3 double	[]	[]	14x3 double	[]	14x3 double
8	[]	[]	[]	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	[]	[]	14x3 double
9	[]	[]	[]	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	[]	14x3 double
10	[]	[]	[]	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	[]	14x3 double
11	[]	[]	[]	14x3 double	14x3 double	14x3 double	14x3 double	[]	14x3 double	[]	[]	[]
12	[]	[]	[]	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	[]	[]	[]	[]
13	[]	[]	[]	14x3 double	[]	[]	[]	[]	[]	[]	14x3 double	[]
14	[]	[]	[]	14x3 double	[]	[]	[]	14x3 double	[]	[]	[]	[]
15	[]	[]	[]	14x3 double	[]	[]	[]	14x3 double	[]	[]	[]	[]
16	[]	[]	[]	14x3 double	[]	[]	[]	14x3 double	[]	[]	[]	[]
17	[]	[]	[]	[]	[]	[]	[]	14x3 double	[]	[]	[]	[]

Abbildung 44: Ausschnitt aus Matlab – Anzahl effizienter Wege (Startknoten 100, Gehzeit & Höhenunterschied, alle Lösungen)

Wendet man die Kriterien Weglänge (Schrägdistanz) und totaler Höhenunterschied an so werden durchschnittlich 4.3 Lösungen für jeden Zielknoten produziert (Abbildung 45).

	22	23	24	25	26	27	28	29	30	31	32	
1	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	[]	14x3 double	14x3 double	14x3 double	14x3 double
2	[]	[]	[]	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double
3	[]	[]	[]	14x3 double	14x3 double	14x3 double	14x3 double	[]	14x3 double	14x3 double	[]	[]
4	[]	[]	[]	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	[]	14x3 double
5	[]	[]	[]	14x3 double	14x3 double	[]	14x3 double	[]	14x3 double	14x3 double	14x3 double	[]
6	[]	[]	[]	14x3 double	[]	14x3 double	[]	14x3 double	[]	14x3 double	14x3 double	[]
7	[]	[]	[]	[]	[]	[]	[]	[]	[]	14x3 double	[]	[]
8	[]	[]	[]	[]	[]	[]	[]	14x3 double	[]	[]	[]	[]
9	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	14x3 double
10	[]	[]	[]	[]	[]	[]	[]	14x3 double	[]	[]	[]	[]
11	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	14x3 double	[]
12	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]
13	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]
14	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]
15	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	14x3 double	14x3 double
16	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]
17	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	14x3 double

Abbildung 45: Ausschnitt aus Matlab – Anzahl effizienter Wege (Startknoten 100, Höhenunterschied & Weglänge, alle Lösungen)

Wenn man drei Kriterien in die Berechnung einfließen lässt, die Gehzeit, die Weglänge (Schrägdistanz) und den totalen Höhenunterschied, so ergeben sich durchschnittlich 1.3 effiziente Lösungen pro Zielknoten (Abbildung 46).

	22	23	24	25	26	27	28	29	30	31	32	
1	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	[]	[]	14x3 double	14x3 double	14x3 double
2	[]	[]	[]	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	[]
3	[]	[]	[]	[]	[]	[]	[]	[]	14x3 double	14x3 double	[]	[]
4	[]	[]	[]	[]	[]	[]	[]	14x3 double	[]	[]	[]	[]
5	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]
6	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]
7	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]
8	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]
9	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]
10	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]
11	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]
12	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]
13	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]
14	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]
15	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]
16	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]
17	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]

Abbildung 46: Ausschnitt aus Matlab – Anzahl effizienter Wege (Startknoten 100, Höhenunterschied & Weglänge & Gehzeit, alle Lösungen)

Werden die drei Kriterien Weglänge (Schrägdistanz), Gehzeit und der Untergrund „Standardpfad“ verwendet, so werden durchschnittlich 2.0 Lösungen produziert (Abbildung 47).

	22	23	24	25	26	27	28	29	30	31	32	
1	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	[]	14x3 double	14x3 double	14x3 double
2	[]	[]	[]	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	14x3 double	[]	[]	[]
3	[]	14x3 double	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]
4	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]
5	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]
6	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]
7	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]
8	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]
9	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]
10	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]
11	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]
12	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]
13	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]
14	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]
15	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]
16	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]
17	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]

Abbildung 47: Ausschnitt aus Matlab – Anzahl effizienter Wege (Startknoten 100, Standardpfad & Weglänge & Gehzeit, alle Lösungen)

Je nachdem welche Optimierungsmethode gewählt ist und welche Kriterien einfließen, produziert der Algorithmus unterschiedliche Ergebnisse.

7.4 Statistik und Analyse

Betrachtet man das vorhandene Wandernetz, so besteht dieses aus 174 Knoten. Daraus ergibt sich die Frage: „Wie viele Wegemöglichkeiten, das heißt wie viele Kombinationen von Start- und Zielknoten, gibt es im Wandernetz?“ Die resultierende Anzahl an Möglichkeiten beträgt 30102. Interessant ist nun zu wissen, bei wie vielen dieser 30102 Möglichkeiten, mit bestimmten Einstellungen, es mehr als eine effiziente Lösung gibt. Dies ist deshalb interessant, weil dadurch sich abschätzen lässt, ob sich die Anwendung eines multikriteriellen Routenplanungsalgorithmus wirklich auszahlt.

Um dies zu beantworten, wird als erstes die Gehzeit (auf eine Viertelstunde gerundet) als Hauptkriterium angewendet und es werden alle Pareto-optimalen Wege von allen Startknoten zu allen Zielknoten gerechnet. Die Ergebnisse sind in Tabelle 32 dargestellt. Es gibt von jedem Startknoten 173 Möglichkeiten, wobei die Anzahl, bei wie vielen der 173 Möglichkeiten es mehr als eine effiziente Lösung gibt, dargestellt wird.

Tabelle 32: Anzahl der Möglichkeiten, die mehr als eine effiziente Lösung aufweisen - Gehzeit als Hauptkriterium

Startknoten	Anzahl (max. 173 möglich)	Startknoten	Anzahl (max. 173 möglich)
1	131	88	98
2	88	89	125
3	80	90	123
4	79	91	128
5	73	92	105
6	64	93	116
7	65	94	96
8	73	95	99
9	74	96	90
10	75	97	98
11	65	98	96
12	64	99	107
13	66	100	124
14	66	101	121
15	93	102	101
16	70	103	97
17	71	104	109
18	70	105	99
19	101	106	112
20	59	107	144
21	56	108	98
22	128	109	58
23	67	110	60
24	42	111	55
25	98	112	56
26	108	113	116

27	99	114	99
28	102	115	67
29	42	116	69
30	50	117	66
31	108	118	67
32	94	119	52
33	146	120	47
34	96	121	51
35	98	122	48
36	101	123	50
37	102	124	62
38	117	125	63
39	88	126	65
40	63	127	58
41	124	128	53
42	96	129	51
43	98	130	91
44	52	131	57
45	69	132	56
46	82	133	53
47	108	134	88
48	106	135	110
49	50	136	106
50	90	137	111
51	52	138	110
52	81	139	114
53	80	140	109
54	82	141	59
55	126	142	79
56	95	143	48
57	90	144	103
58	83	145	105
59	89	146	102
60	66	147	128
61	84	148	122
62	65	149	119
63	33	150	130
64	79	151	131
65	78	152	126
66	84	153	113
67	100	154	111
68	72	155	109
69	75	156	112
70	53	157	128
71	80	158	97
72	78	159	118

73	80	160	116
74	81	161	134
75	79	162	90
76	82	163	117
77	80	164	112
78	80	165	57
79	118	166	65
80	84	167	54
81	117	168	58
82	123	169	47
83	97	170	44
84	119	171	69
85	107	172	121
86	129	173	67
87	116	174	66

Insgesamt ergibt das 15285 Möglichkeiten, bei denen es mehr als eine effiziente Lösung gibt. Somit ergibt sich, dass ca. 51 % aller Lösungsmöglichkeiten, für die getroffenen Einstellungen, mehr als eine eindeutige Lösung aufweisen. Durchschnittlich ergeben sich pro Startknoten ca. 88 Lösungen, bei denen es mehr als eine effiziente Lösung gibt.

Nun wird die Weglänge (Schrägdistanz) als Hauptkriterium verwendet und es werden wieder alle Pareto-optimalen Wege von allen Startknoten zu allen Zielknoten gerechnet. Die Ergebnisse sind in Tabelle 33 dargestellt.

Tabelle 33: Anzahl der Möglichkeiten, die mehr als eine effiziente Lösung aufweisen - Weglänge als Hauptkriterium

Startknoten	Anzahl (max. 173 möglich)	Startknoten	Anzahl (max. 173 möglich)
1	26	88	1
2	25	89	1
3	24	90	34
4	26	91	2
5	29	92	2
6	25	93	68
7	27	94	1
8	27	95	2
9	26	96	1
10	24	97	1
11	23	98	21
12	23	99	69
13	31	100	1
14	32	101	1
15	30	102	21
16	33	103	21
17	29	104	0
18	29	105	0

19	28	106	0
20	20	107	0
21	20	108	0
22	20	109	27
23	21	110	21
24	20	111	21
25	8	112	22
26	8	113	20
27	9	114	15
28	8	115	2
29	8	116	2
30	7	117	2
31	1	118	2
32	1	119	3
33	0	120	13
34	0	121	14
35	1	122	15
36	2	123	15
37	0	124	16
38	1	125	20
39	1	126	15
40	27	127	0
41	30	128	0
42	30	129	0
43	8	130	1
44	1	131	1
45	1	132	0
46	59	133	0
47	1	134	1
48	4	135	1
49	2	136	1
50	2	137	1
51	0	138	1
52	2	139	1
53	2	140	1
54	2	141	20
55	1	142	21
56	1	143	20
57	1	144	20
58	5	145	20
59	1	146	29
60	3	147	29
61	5	148	29
62	15	149	39
63	3	150	37
64	6	151	39

65	6	152	41
66	8	153	40
67	8	154	41
68	1	155	43
69	3	156	44
70	2	157	46
71	2	158	22
72	2	159	21
73	2	160	22
74	3	161	20
75	3	162	61
76	2	163	21
77	2	164	21
78	2	165	21
79	1	166	22
80	55	167	30
81	31	168	30
82	1	169	20
83	1	170	21
84	1	171	51
85	1	172	21
86	5	173	2
87	11	174	2

Insgesamt ergibt das 2482 Möglichkeiten, bei denen es mehr als eine effiziente Lösung gibt. Somit ergibt sich, dass ca. 8 % aller Lösungsmöglichkeiten, für die getroffenen Einstellungen, mehr als eine eindeutige Lösung aufweisen. Durchschnittlich ergeben sich pro Startknoten ca. 14 Lösungen, bei denen es mehr als eine effiziente Lösung gibt.

Der Unterschied der Ergebnisse der beiden Anwendungen lässt sich dadurch erklären, dass die Weglänge, die auf 100 Meter genau gerundet wird, weniger idente Lösungen mit sich bringt, als die Gehzeit, die auf eine Viertelstunde gerundet wird. Man könnte diese Anwendung auch für weitere Kriterien durchführen, um so auch den Unterschied der Anfälligkeiten des Algorithmus bezüglich der Kriterien zu beobachten. Wie bereits oben erwähnt, sollte durch diese Untersuchung aber in erster Linie geklärt werden, ob sich die Anwendung eines multikriteriellen Routenplanungsalgorithmus, in dem vorliegenden Wandernetz, auszahlt. Wenn die Anzahl an Möglichkeiten, bei denen es mehr als eine effiziente Lösung gibt, sehr gering wäre (z.B. < 5 %), so würde sich die Frage nach der Sinnhaftigkeit, eine multikriterielle Optimierung in die Routenplanung zu inkludieren, stellen. Deshalb kann die Annahme getroffen werden, dass die Anwendung eines multikriteriellen Routenplanungsalgorithmus, in dem vorliegenden Wandernetz, als sinnvoll zu betrachten ist.

Darüber hinaus spielt auch die Performance des Algorithmus, bei der Anwendung dieses Routenplanungsalgorithmus, eine wesentliche Rolle. Je nachdem, welche Einstellungen

getroffen und welcher Startknoten ausgewählt ist, rechnet das Programm unterschiedlich lang. Für das angeführte Beispiel alle Möglichkeiten zu berechnen, wobei die Gehzeit als Hauptkriterium angesetzt wird, rechnet das Programm pro Startknoten durchschnittlich 330 Sekunden. Das ergibt eine Gesamtberechnungszeit für alle 30102 Möglichkeiten von ca. 60000 Sekunden, das entspricht ca. 1000 Minuten oder ca. 16.7 Stunden. Für das zweite Beispiel, bei dem die Weglänge als Hauptkriterium verwendet wird, rechnet das Programm pro Startknoten durchschnittlich 210 Sekunden. Das ergibt eine Gesamtberechnungszeit für alle 30102 Möglichkeiten von ca. 37000 Sekunden, das entspricht ca. 620 Minuten oder ca. 10.3 Stunden. Je nachdem welche Einstellungen getroffen sind, rechnet das Programm für einen ausgewählten Startknoten zwischen 150 und 1000 Sekunden.

8 Fallbeispiele

Nun werden verschiedene Fallbeispiele dargestellt, um eine Übersicht über die Möglichkeiten, die der entwickelte Routenplanungsalgorithmus mit sich bringt, zu veranschaulichen.

8.1 Fallbeispiel 1

Ausgangslage: Wir wollen vom Startpunkt St. Veit (Knotennummer 113) den Speikbodengipfel (Knotennummer 118) erreichen. Als Voraussetzungen definieren wir, dass wir den schnellsten Weg und unter allen schnellsten Wegen denjenigen suchen, bei dem man bei einer Hütte vorbeikommt. Das bedeutet, wir verwenden als Optimierungsmethode die lexikographische Optimalität. Dann wird folgendes Ergebnis berechnet:

Es gibt zwei effiziente Wege. Lösung Nr. 1 und Lösung Nr. 2:

Lösung Nr. 1

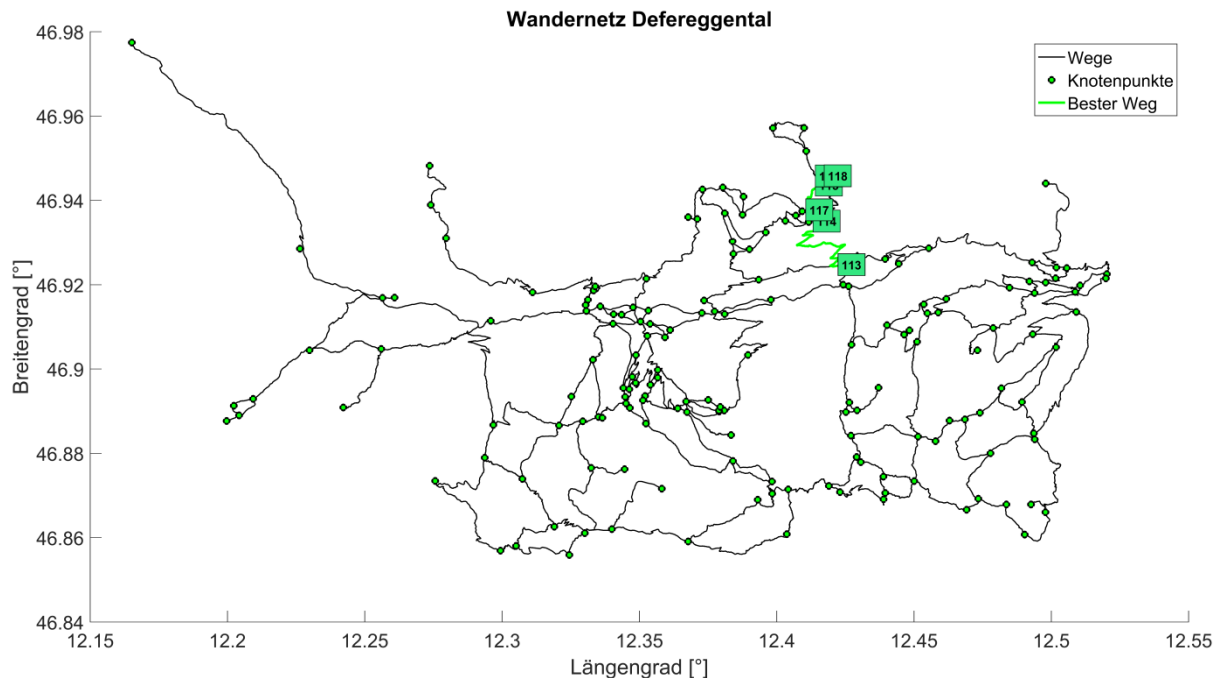


Abbildung 48: Übersichtskarte - optimale Route Nr.1 (St.Veit-Speikboden)

Die optimale Route Nr. 1 verläuft vom Startknoten St. Veit (113) über die Speikbodenhütte (114), die Zischkealm (117), die Abzweigung Froezalm-Speikboden (115) und über den Knoten Speikboden (116) auf den Gipfel des Speikboden (118). In Abbildung 48 ist eine Übersichtskarte dargestellt, in Abbildung 49 ein vergrößerter Ausschnitt davon.

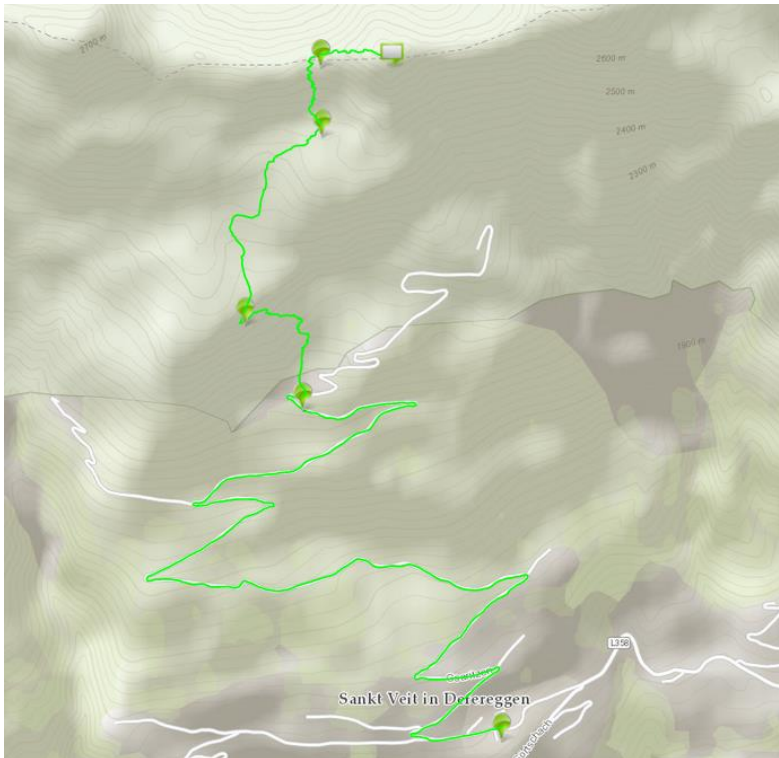


Abbildung 51: Web Map Display (World Topographic Map) - optimale Route Nr.1 (St.Veit-Speikboden)

In Abbildung 51 ist als Layer die World Topographic Map gewählt.

Tabelle 34: Ergebnisse (Gehzeit & geometrische Kriterien) der optimalen Route Nr.1 (St.Veit-Speikboden)

Knoten	113	114	117	115	116	118
Gehzeit [h]	0	2	2.75	3.5	4	4.25
Weglänge (horizontal) [m]	0	5200	5900	6800	7200	7500
Höhenunterschied [m]	0	556	760	960	1083	1158
durchschnittliche Steigung [%]	0	11	14	15	16	16
durchschnittliche Höhe [m]	1495	1800	2000	2200	2400	2500
relative Steigung [%]	0	11	14	15	16	16
Weglänge (schräg) [m]	0	5300	6100	7100	7600	7900
totaler Höhenunterschied [m]	0	556	760	962	1085	1160

Als Ergebnis (Tabellen 34, 35) erhält man eine Gehzeit von 4.25 h, einen Höhenunterschied von 1158 m, eine Weglänge (schräg) von 7.9 km, eine durchschnittliche Steigung von 16 % und die durchschnittliche Höhe dieser Route beträgt 2500 m. Die Anzahl der Hütten beträgt 1. Dies ist die Speikbodenhütte beim Knoten 114. Als weiteres Ergebnis sind noch die Untergrundbeschaffenheit und POI's zu nennen.

Tabelle 35: Ergebnisse (Untergrund & POI's) der optimalen Route Nr.1 (St.Veit-Speikboden)

Untergrund	[%]	POI's	Anzahl
Asphaltstraße	22	Sehenswürdigkeit	1
Schotterweg	44	See	0
steiniger Weg	0	schöner Weg	4
sandiger/erdiger Weg	0	schöne Aussicht	5
Wiesenweg	0	ruhig	4
Waldpfad	0	über Waldgrenze	3
steiniger Pfad	4	im Wald	1
Standardpfad	27	am Bach	0
Pfad am Fels	0	Almhütte entlang des Weges	1
Wiesenspfad	3	Grad entlang	1
-		Brücke/Überqueren eines Baches	0
-		leicht bewaldet	1

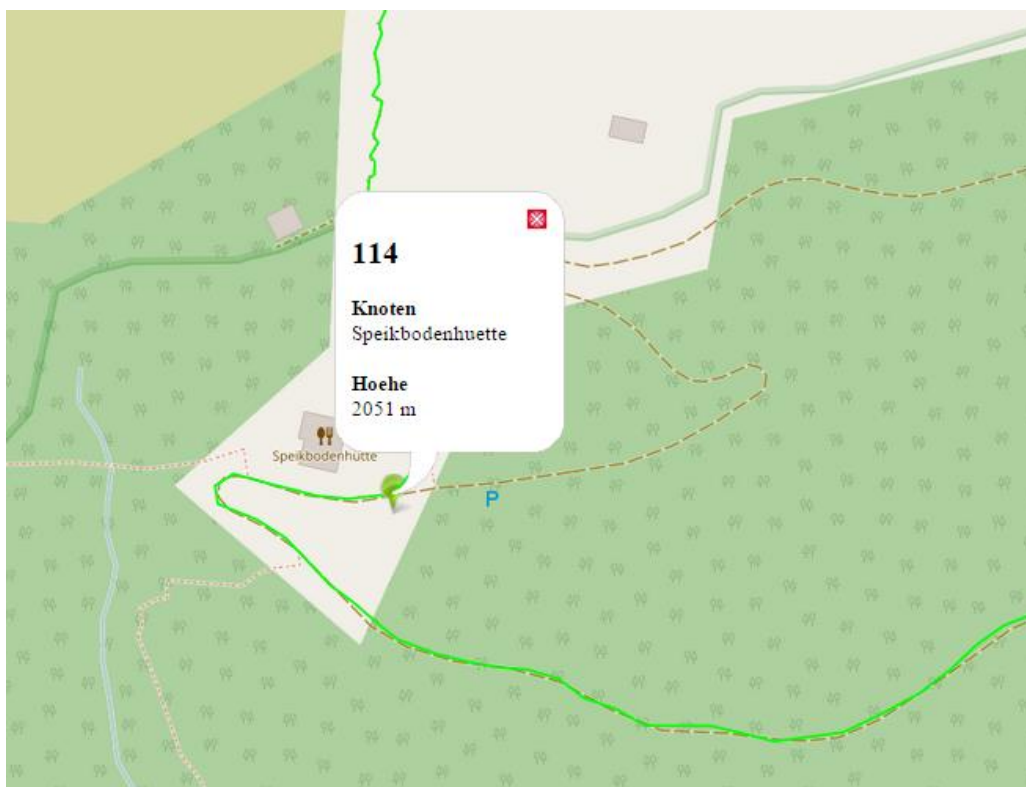


Abbildung 52: Web Map Display (Open Street Map) - Hütte entlang des Weges (Speikbodenhütte)

In Abbildung 52 ist ein Ausschnitt aus dem WMS abgebildet, die die einzige Hütte (Speikbodenhütte) entlang des Weges zeigt.

Lösung Nr. 2

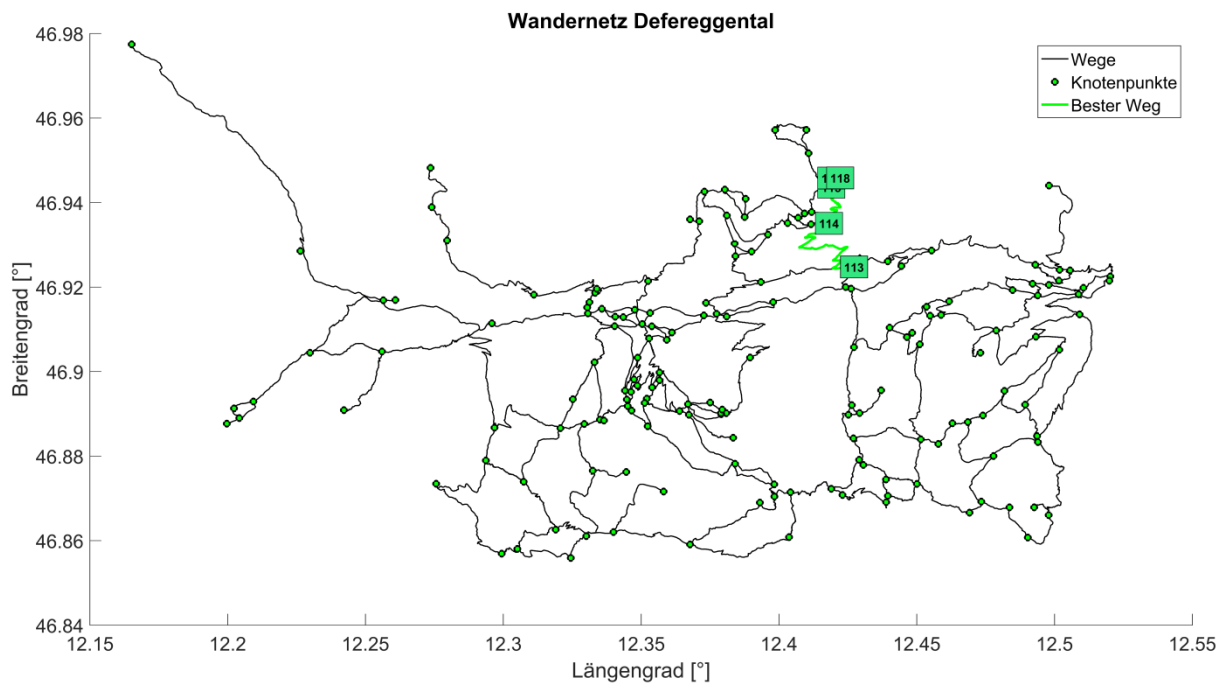


Abbildung 53: Übersichtskarte - optimale Route Nr.2 (St.Veit-Speikboden)

Die optimale Route Nr. 2 verläuft vom Startknoten St. Veit (113) über die Speikbodenhütte (114), die Abzweigung Froezalm-Speikboden (115) und über den Knoten Speikboden (116) auf den Gipfel des Speikbodens (118). In Abbildung 53 ist eine Übersichtskarte dargestellt, in Abbildung 54 ein vergrößerter Ausschnitt davon.

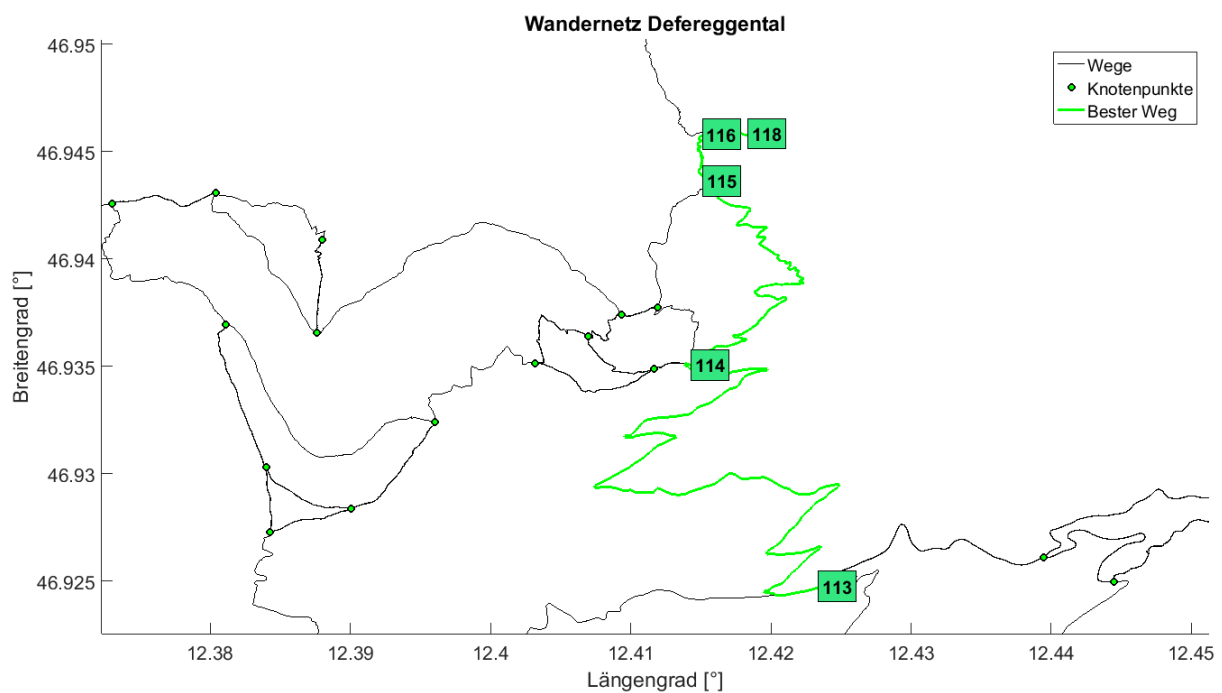


Abbildung 54: Ausschnitt aus Übersichtskarte - optimale Route Nr.2 (St.Veit-Speikboden)

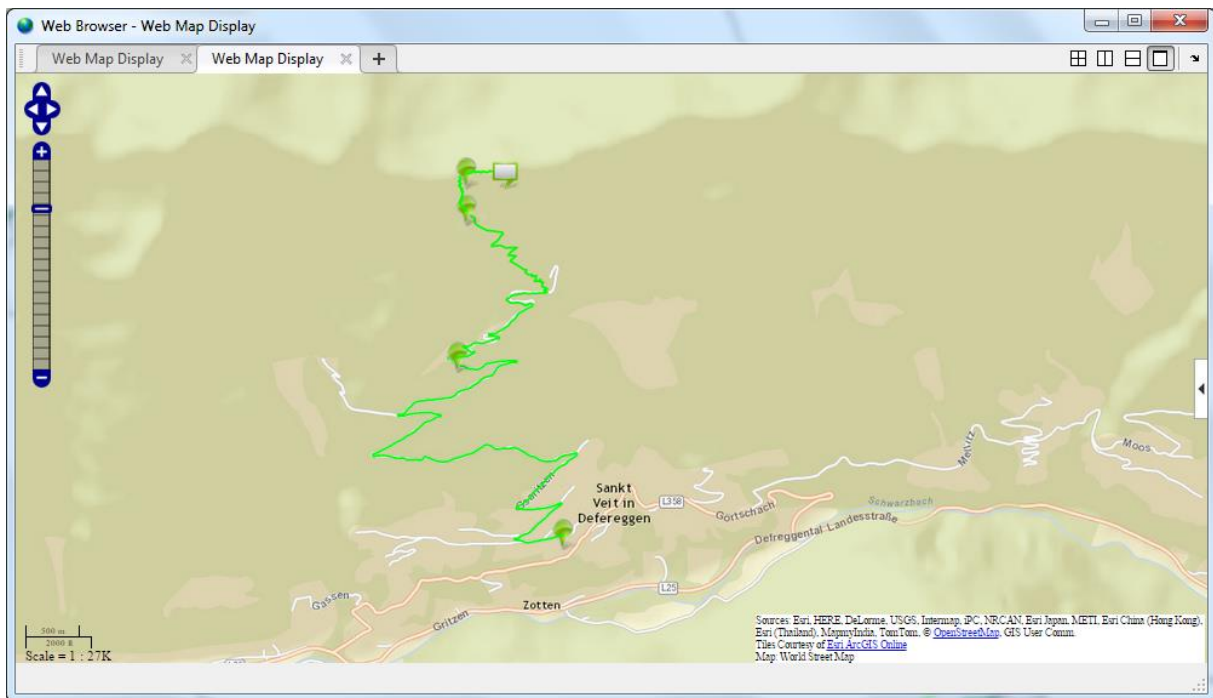


Abbildung 55: Web Map Display (World Street Map) - optimale Route Nr.2 (St.Veit-Speikboden)

In Abbildung 55 ist ein Ausschnitt aus dem erstellten WMS dargestellt. Der Baselayer ist die World Street Map.

Tabelle 36: Ergebnisse (Gehzeit & geometrische Kriterien) der optimalen Route Nr.2 (St.Veit-Speikboden)

Knoten	113	114	115	116	118
Gehzeit [h]	0	2	3.5	4	4.25
Weglänge (horizontal) [m]	0	5200	7800	8200	8500
Höhenunterschied [m]	0	556	960	1083	1158
durchschnittliche Steigung [%]	0	11	14	15	15
durchschnittliche Höhe [m]	1495	1800	2000	2300	2500
relative Steigung [%]	0	11	13	14	14
Weglänge (schräg) [m]	0	5300	8000	8500	8800
totaler Höhenunterschied [m]	0	556	960	1083	1158

Als Ergebnis (Tabellen 36, 37) erhält man eine Gehzeit von 4.25 h, einen Höhenunterschied von 1158 m, eine Weglänge (schräg) von 8.8 km, eine durchschnittliche Steigung von 15 % und die durchschnittliche Höhe dieser Route beträgt 2500 m. Die Anzahl der Hütten beträgt ebenfalls 1. Dies ist die Speikbodenhütte beim Knoten 114. Als weitere Ergebnisse sind noch die Untergrundbeschaffenheit und die POI's zu nennen.

Tabelle 37: Ergebnisse (Untergrund & POI's) der optimalen Route Nr.1 (St.Veit-Speikboden)

Untergrund	[%]	POI's	Anzahl
Asphaltstraße	15	Sehenswürdigkeit	1
Schotterweg	53	See	0
steiniger Weg	0	schöner Weg	2
sandiger/erdiger Weg	0	schöne Aussicht	4
Wiesenweg	0	ruhig	3
Waldpfad	0	über Waldgrenze	2
steiniger Pfad	4	im Wald	1
Standardpfad	28	am Bach	0
Pfad am Fels	0	Almhütte entlang des Weges	1
Wiesenspfad	0	Grad entlang	1
-		Brücke/Überqueren eines Baches	0
-		leicht bewaldet	1

Wie man sieht, entstehen zwei idente Lösungen, wenn man die Kriterien Gehzeit und POI „Almhütte“ anwendet. Beide Wege beinhalten eine Hütte (Speikbodenhütte beim Knoten 114) und die Gehzeit beträgt in beiden Fällen 4.25 Stunden. Falls man im Nachhinein nur eine optimale Lösung will, so kann ein drittes Kriterium berücksichtigt werden und aufgrund dessen, die optimale Lösung gewählt werden.

8.2 Fallbeispiel 2

Ausgangslage: Wir wollen vom Startpunkt Leppetalm (Knotennummer 103) die Mooseralm (Knotennummer 34) erreichen. Als Voraussetzungen definieren wir, dass wir den kürzesten Weg und unter allen kürzesten Wegen denjenigen suchen, mit der bevorzugten Untergrundbeschaffenheit Standardpfad. Das bedeutet, wir verwenden als Optimierungsmethode die lexikographische Optimalität. Dann wird folgendes Ergebnis berechnet:

Es gibt 2 Lösungen mit identer Weglänge, doch eine dieser beiden Lösungen weist zu einem größeren Prozentsatz den Untergrund Standardpfad auf. Diese Lösung wird folgend dargestellt.

Lösung

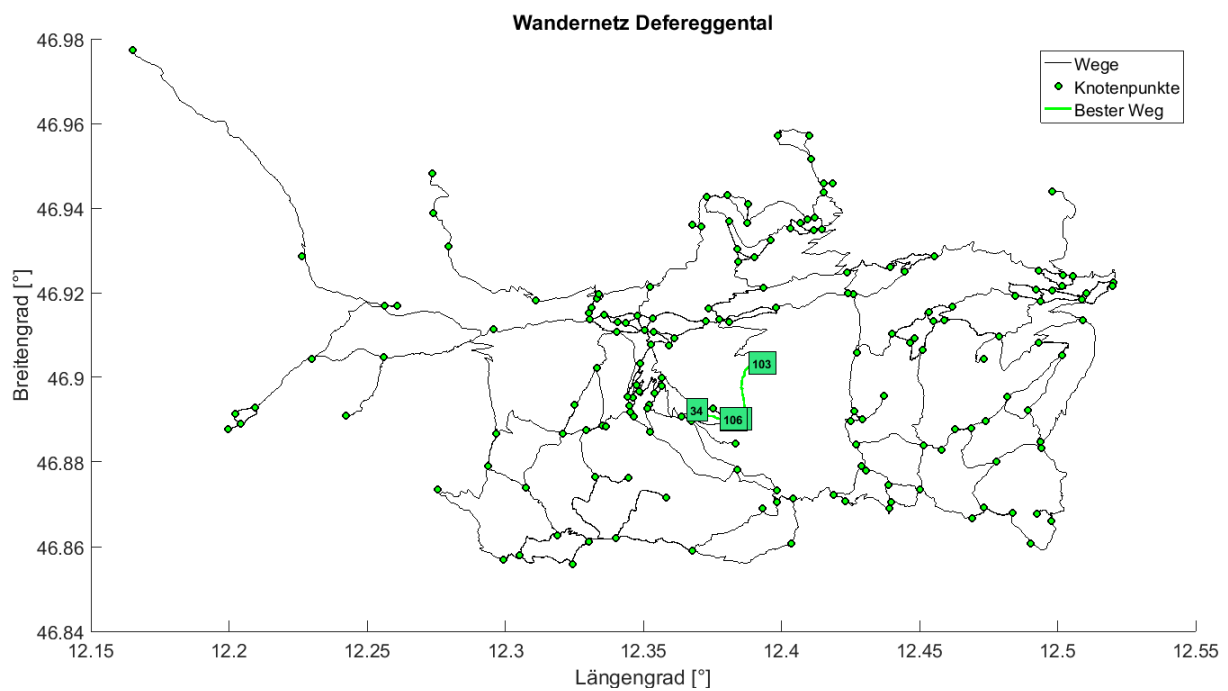


Abbildung 56: Übersichtskarte - optimale Route (Leppetalalm-Mooseralm)

In den Abbildungen 56 und 57 ist die optimale Route dargestellt. Der Weg startet bei der Leppetalalm (103) und verläuft über den Knoten Leppetalalm (104), die Abzweigung Ochsenlacke-Leppetalalm (105) und den Knoten Ochsenlacke (106) zur Mooseralm (34).

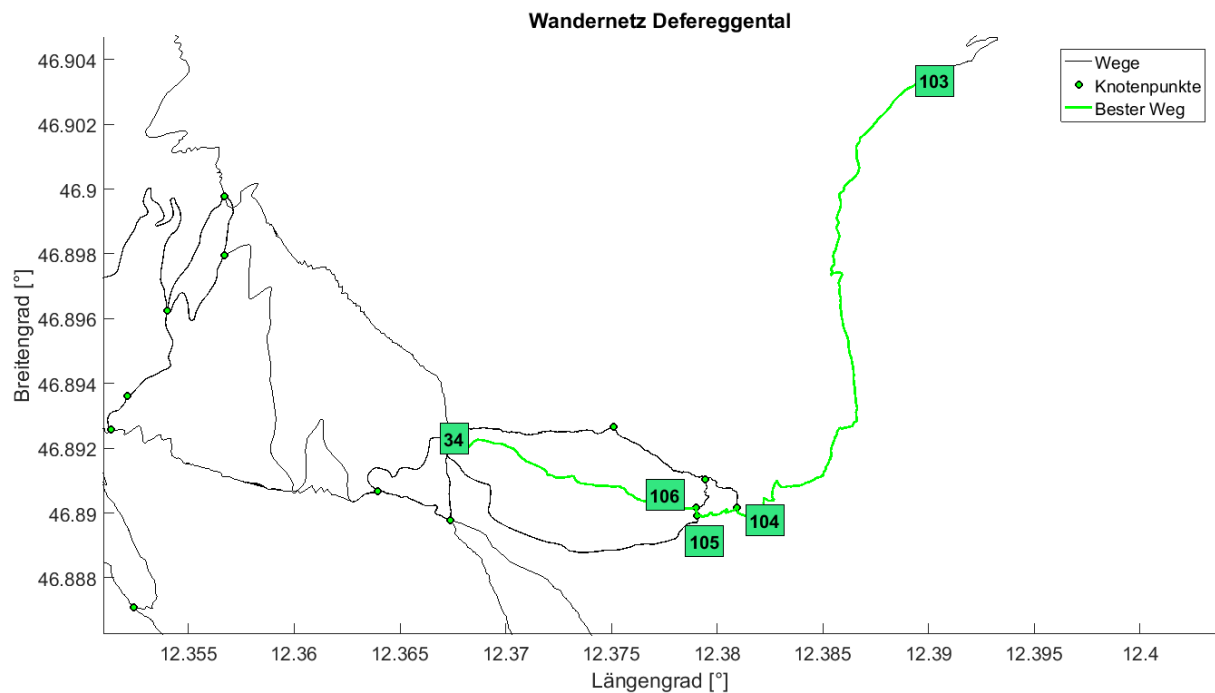


Abbildung 57: Ausschnitt aus Übersichtskarte - optimale Route (Leppetalalm-Mooseralm)

Tabelle 38: Ergebnisse (Gehzeit & geometrische Kriterien) der optimalen Route (Leppetalm-Mooseralm)

Knoten	103	104	105	106	34
Gehzeit [h]	0	1.5	1.75	2	2.25
Weglänge (horizontal) [m]	0	2200	2400	2500	3400
Höhenunterschied [m]	0	463	506	507	581
durchschnittliche Steigung [%]	0	23	23	23	17
durchschnittliche Höhe [m]	1930	2200	2300	2400	2400
relative Steigung [%]	0	21	21	21	15
Weglänge (schräg) [m]	0	2300	2500	2600	3600
totaler Höhenunterschied [m]	0	469	512	513	589

Als Ergebnis erhält man eine Gehzeit von 2.25 h, einen Höhenunterschied von 581 m, eine Weglänge (schräg) von 3.6 km, eine durchschnittliche Steigung von 17 % und die durchschnittliche Höhe dieser Route beträgt 2400 m. Der Anteil der Untergrundbeschaffenheit Standardpfad beträgt 28 %. Die Ergebnisse sind in den Tabellen 38 und 39 dargestellt.

Tabelle 39: Ergebnisse (Untergrund & POI's) der optimalen Route (Leppetalm-Mooseralm)

Untergrund	[%]	POI's	Anzahl
Asphaltstraße	0	Sehenswürdigkeit	0
Schotterweg	8	See	1
steiniger Weg	0	schöner Weg	1
sandiger/erdiger Weg	0	schöne Aussicht	3
Wiesenweg	0	ruhig	4
Waldpfad	18	über Waldgrenze	3
steiniger Pfad	28	im Wald	0
Standardpfad	28	am Bach	0
Pfad am Fels	0	Almhütte entlang des Weges	2
Wiesenspfad	18	Grad entlang	0
-		Brücke/Überqueren eines Baches	1
-		leicht bewaldet	1

Mit den oben genannten Einstellungen erhält man diese eine Pareto-optimale Lösung.

8.3 Fallbeispiel 3

Ausgangslage: Wir wollen vom Startpunkt Start Zwenewaldweg (Knotennummer 1) auf die Leisacher Alm (Knotennummer 14). Wir wollen in erster Linie einen schnellen Weg und verwenden die Steigung als zweites Kriterium. Die Steigung wird, nachdem die schnellsten Lösungen produziert wurden, betrachtet. Wir setzen die veränderte lexikographische

Optimalität ein, um zu bestimmen, ob Lösungen dominiert werden. So werden vier Lösungen mit schnellster Gehzeit produziert.

Zum Vergleich werden die unterschiedlichen Routenverläufe betrachtet. In den Abbildungen 60 bis 63 sind mit Hilfe der Übersichtskarte die unterschiedlichen Routenverläufe dargestellt.

Routenverläufe der 4 Lösungen

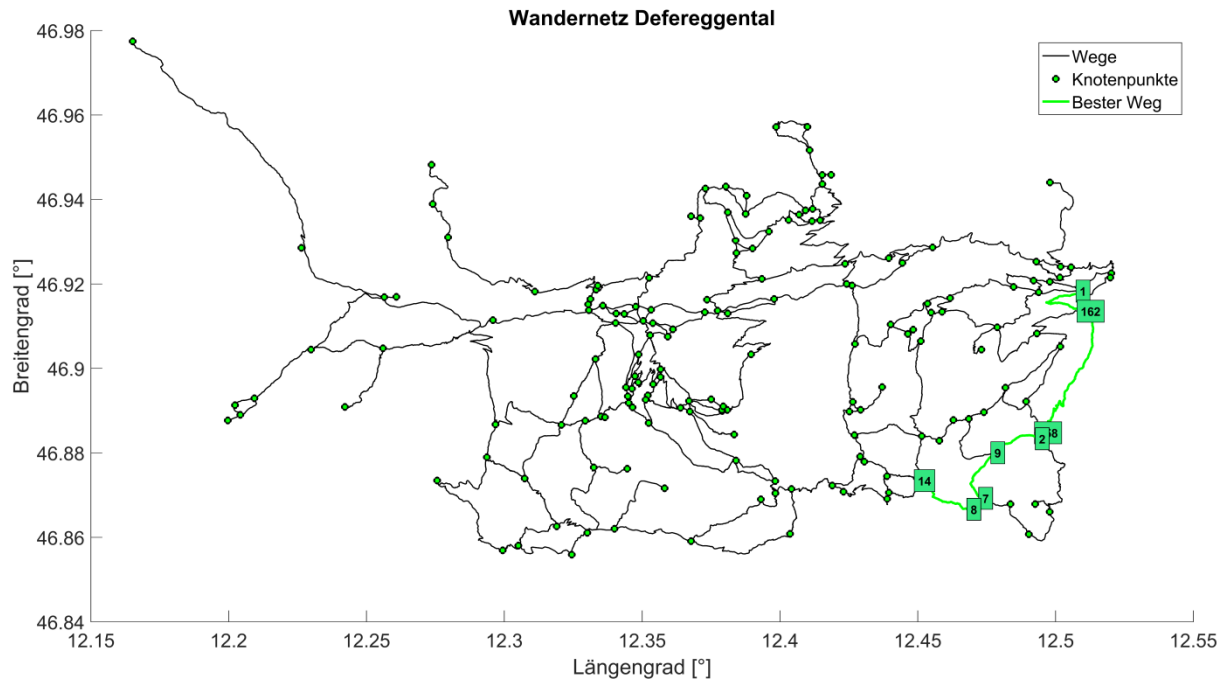


Abbildung 60: Übersichtskarte - optimale Route Nr.1 (Start Zwenewaldweg-Leisacher Alm)

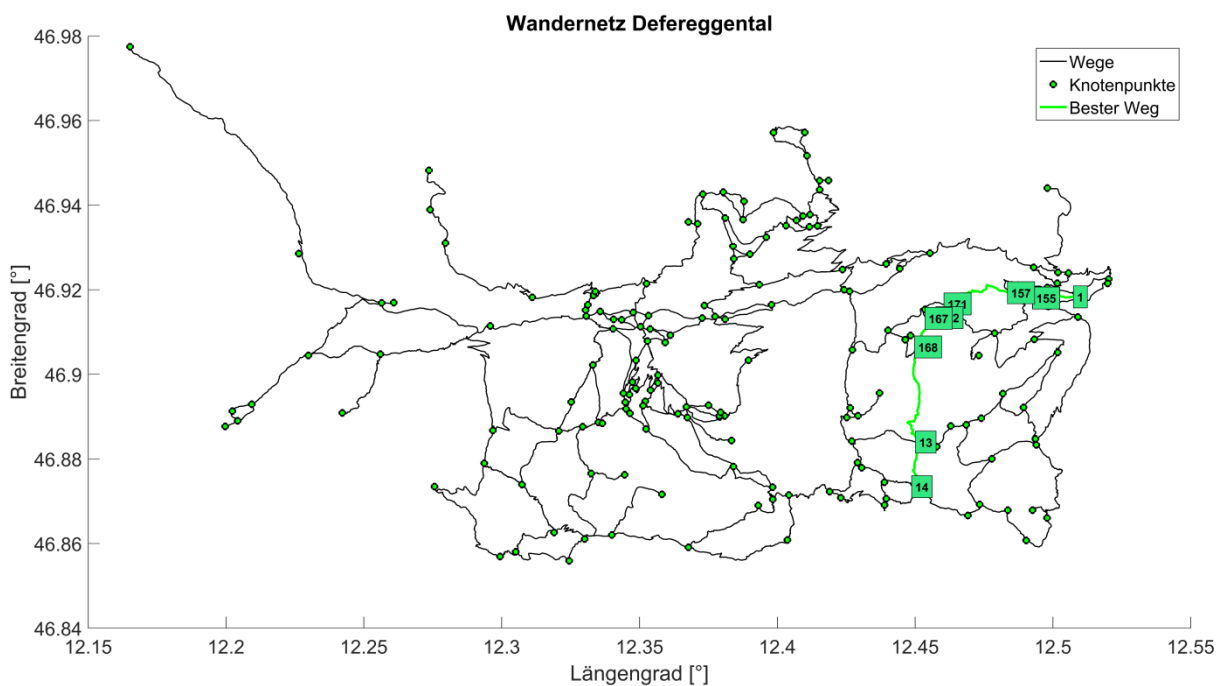


Abbildung 61: Übersichtskarte - optimale Route Nr.2 (Start Zwenewaldweg-Leisacher Alm)

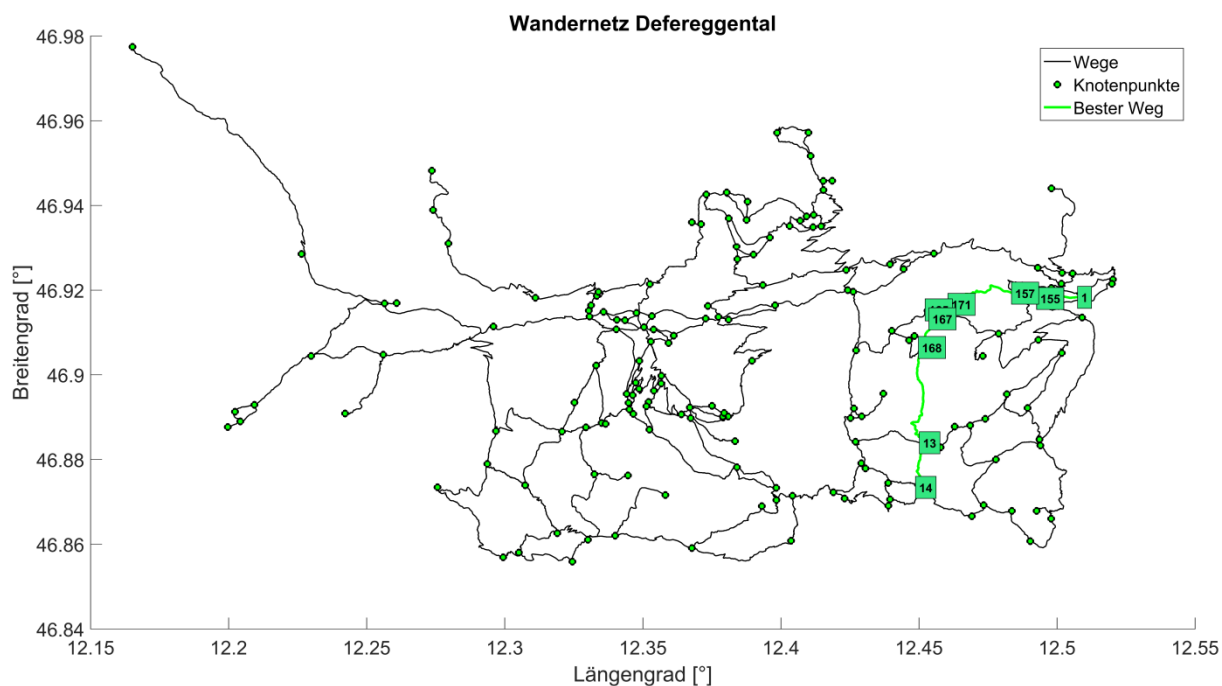


Abbildung 62: Übersichtskarte - optimale Route Nr.3 (Start Zwenewaldweg-Leisacher Alm)

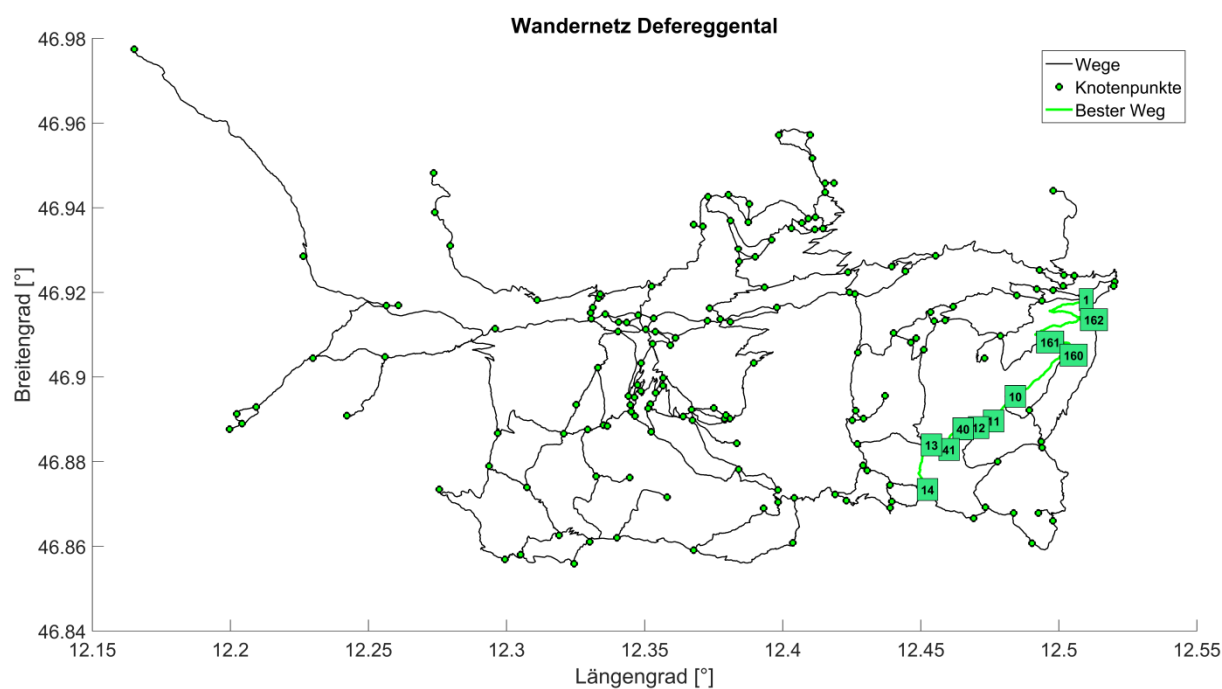


Abbildung 63: Übersichtskarte - optimale Route Nr.4 (Start Zwenewaldweg-Leisacher Alm)

Wie man sieht, verlaufen alle vier Routen unterschiedlich. Um dies besser darzustellen, werden Ausschnitte aus den Übersichtskarten betrachtet (siehe Abbildungen 64-67).

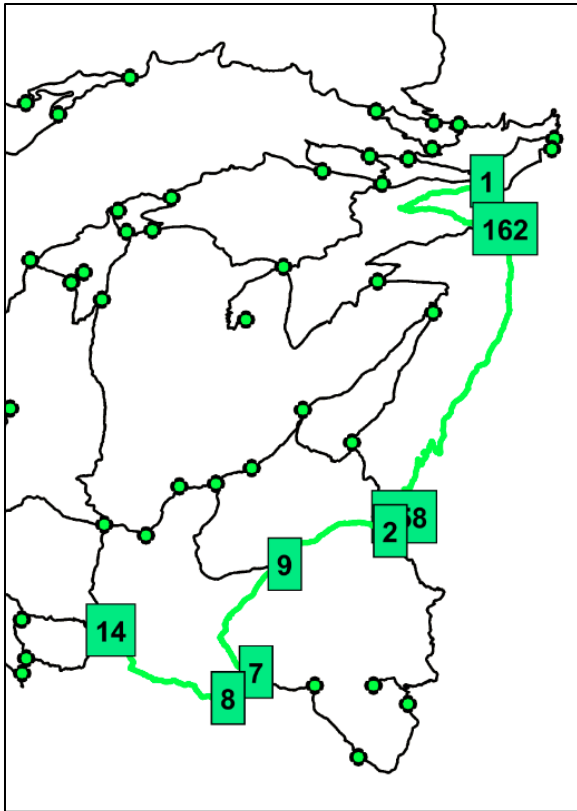


Abbildung 64: optimale Route Nr.1

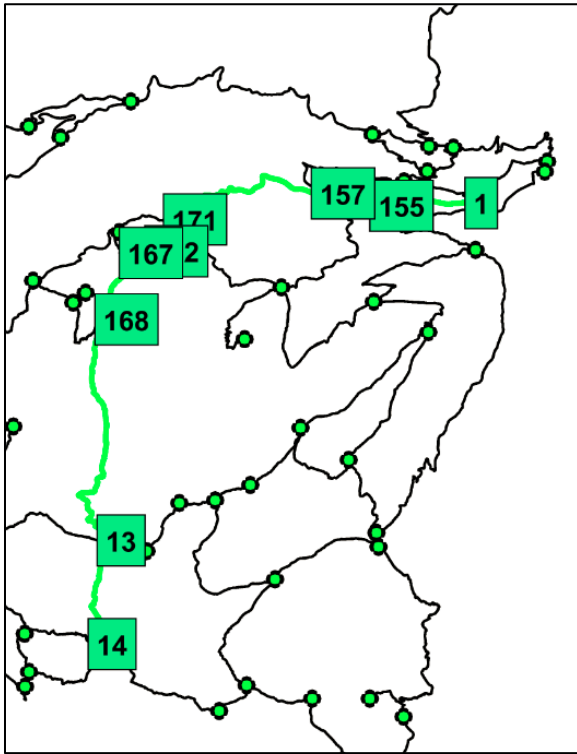


Abbildung 65: optimale Route Nr.2

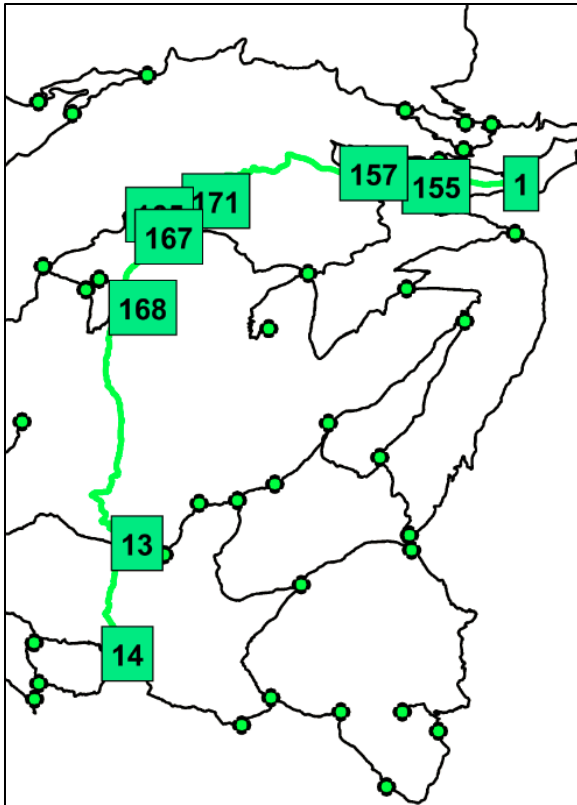


Abbildung 66: optimale Route Nr.3

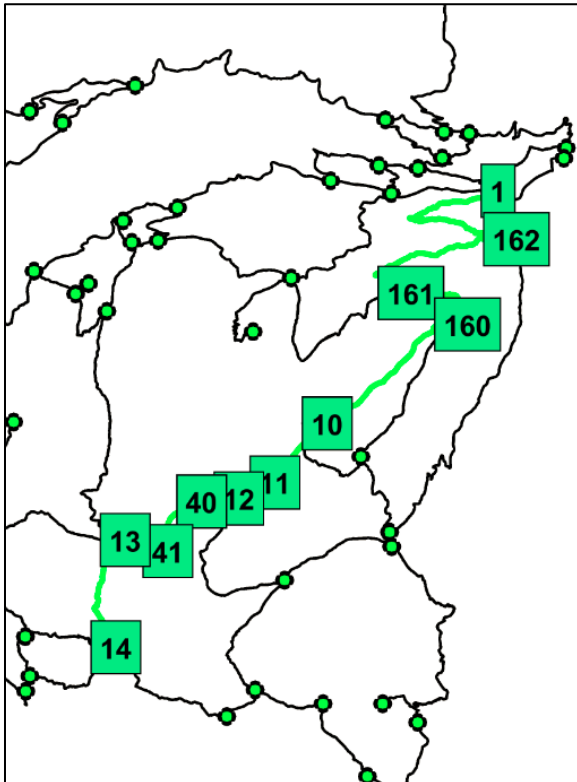


Abbildung 67: optimale Route Nr.4

In weiterer Folge werden die unterschiedlichen Routenverläufe noch mit Hilfe des erstellten WMS (Open Street Map als Baselayer) dargestellt (Abbildungen 68-71).

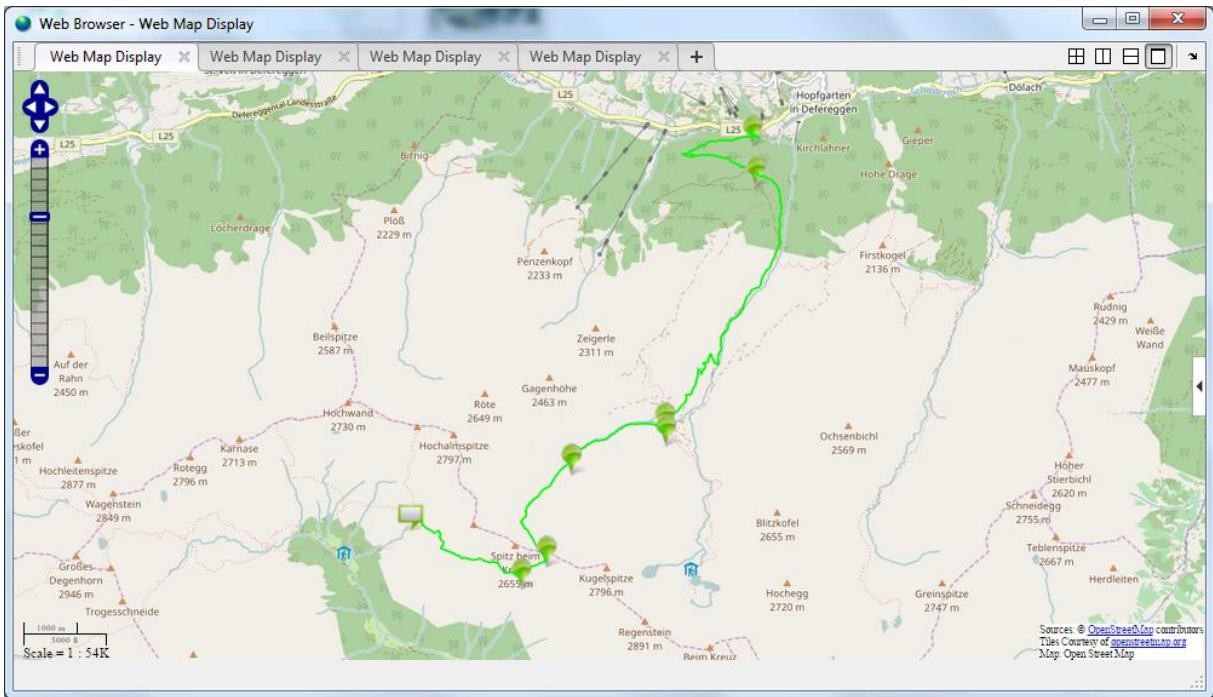


Abbildung 68: WMS Ausschnitt (Open Street Map) - optimale Route Nr.1

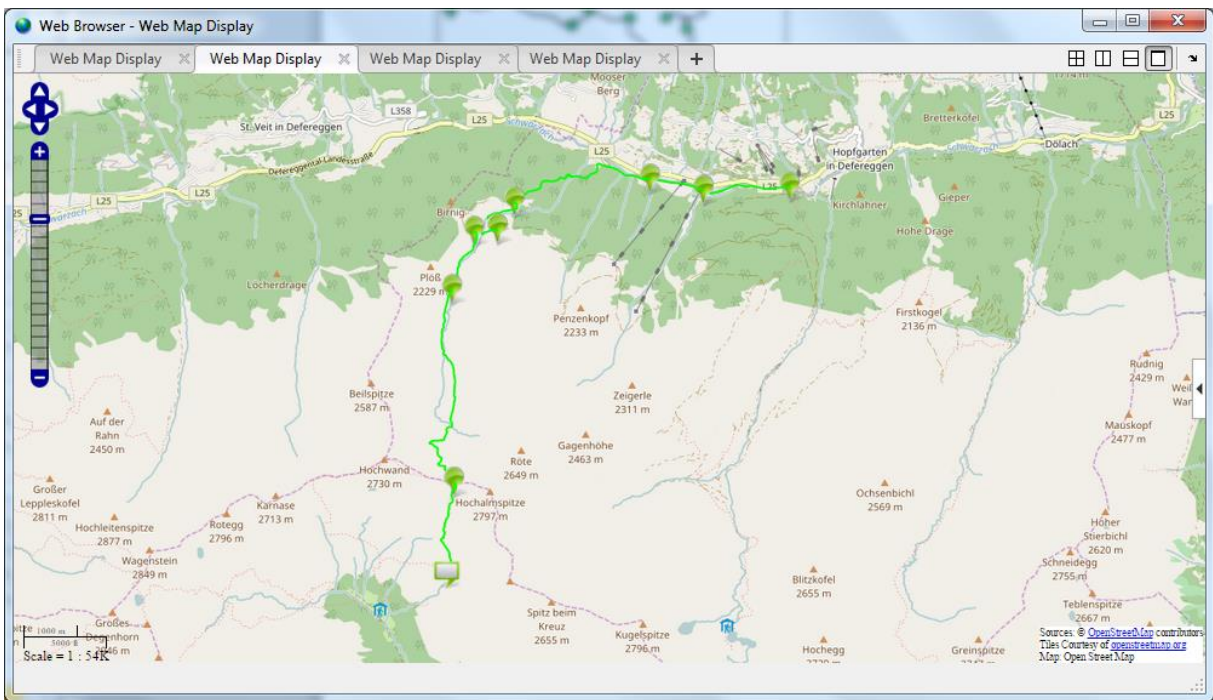


Abbildung 69: WMS Ausschnitt (Open Street Map) - optimale Route Nr.2

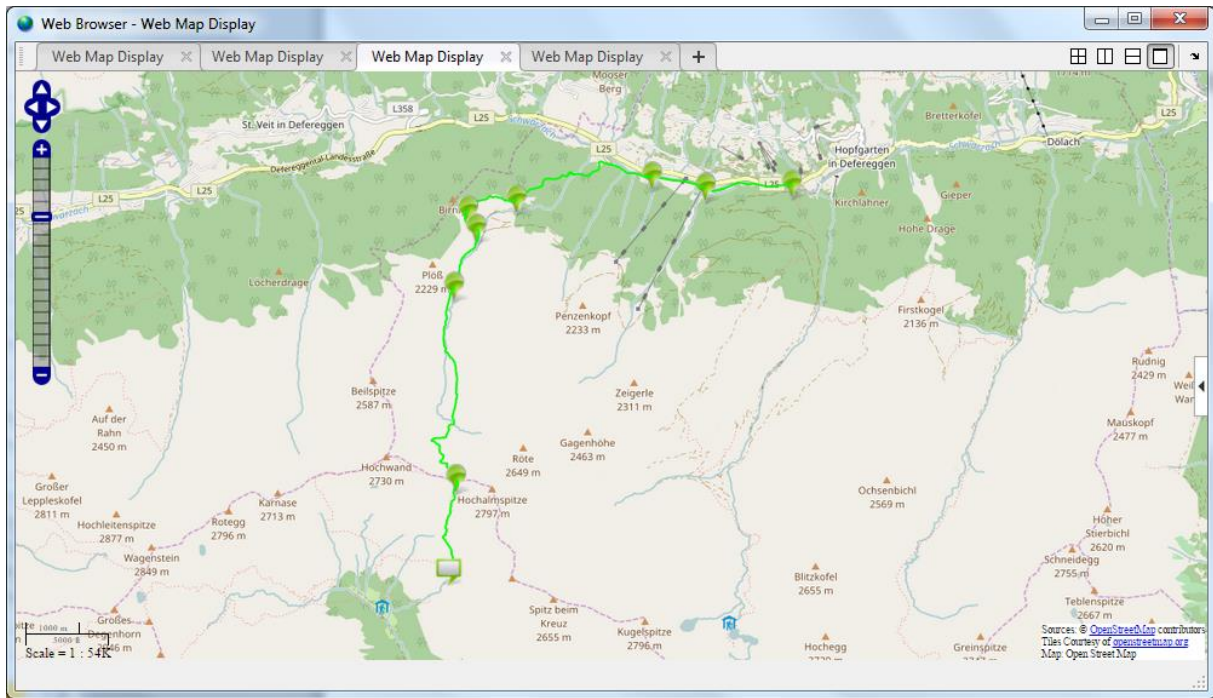


Abbildung 70: WMS Ausschnitt (Open Street Map) - optimale Route Nr.3

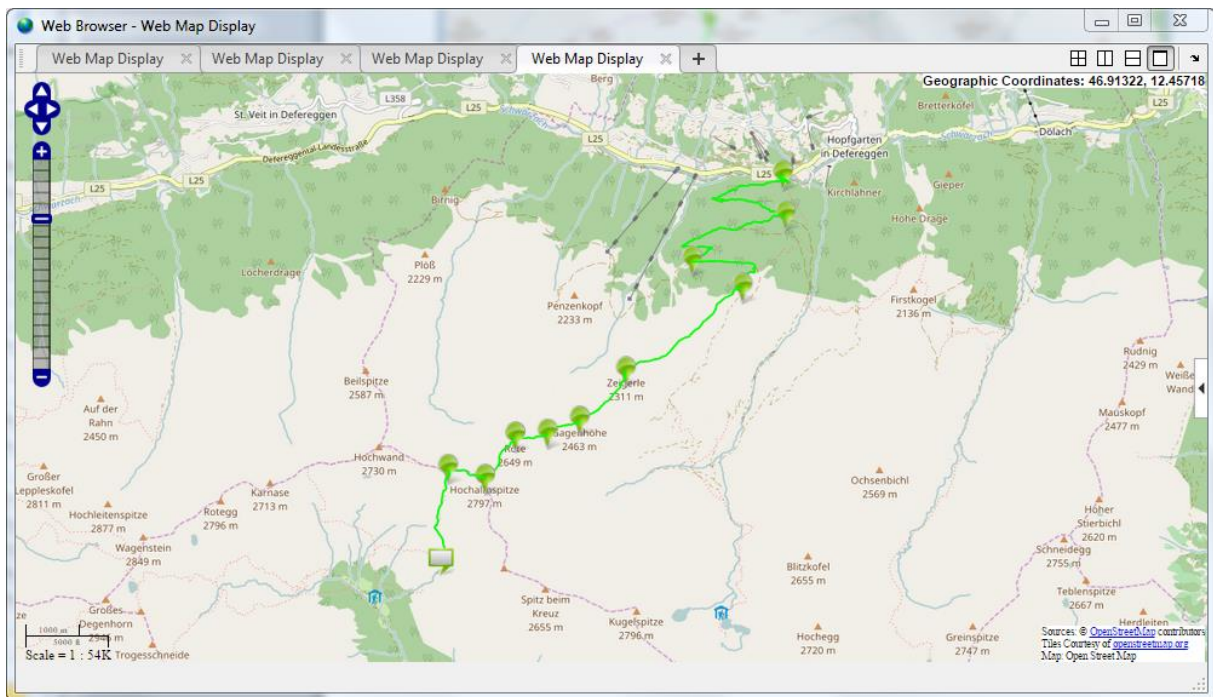


Abbildung 71: WMS Ausschnitt (Open Street Map) - optimale Route Nr.4

Betrachtet man die Routenverläufe der vier Lösungen mit den zugehörigen Kriterien Gehzeit und durchschnittliche Steigung so ergeben sich folgende Ergebnisse (Tabelle 40):

Tabelle 40: Routenverläufe der 4 Lösungen mit Gehzeit & Steigung

Wege	Route Nr. 1	Route Nr. 2	Route Nr. 3	Route Nr. 4
Vom Knoten	1	1	1	1
	162	155	155	162
	158	157	157	161
	2	171	171	160
	9	172	165	10
	7	167	167	11
	8	168	168	12
Bis Knoten	14	13	13	40
Bis Knoten		14	14	41
				13
Bis Knoten				14
Gehzeit [h]	5.75	5.75	5.75	5.75
Steigung [%]	11	10	10	11

Es entstehen vier schwache Pareto-optimale Lösungen. Wollte man unter allen schnellsten Wegen denjenigen mit der geringsten durchschnittlichen Steigung, so würden die Lösungen Nr. 2 und Nr. 3 gewählt werden. Man kann dieses Ergebnis nun nach Abschluss der Berechnungen aber auch auf weitere Kriterien untersuchen. Man könnte zusätzlich die relative Steigung betrachten, da diese mit Hilfe des totalen Höhenunterschiedes berechnet wird. Betrachtet man die Ergebnisse der Lösungen Nr. 2 und Nr. 3 bezüglich der relativen Steigung, so ergibt sich, dass die Lösung Nr. 2 mit einer relativen Steigung von 8 % eine geringere, relative Steigung als die Lösung Nr. 4 mit 9 % aufweist. Da der Unterschied der durchschnittlichen Steigung aller Lösungen sehr gering ist (innerhalb von 1 %), kann auch auf zusätzliche Kriterien geschaut werden, um die optimale Route zu wählen. Dazu werden die Ergebnisse der vier Pareto-optimalen Wege nun genauer betrachtet (Tabellen 41 bis 44).

Tabelle 41: Ergebnis Labelvektor Lösung Nr.1 (Zwenewaldweg-Leisacher Alm)

Kriterien		Untergrund	[%]	POI's	Anzahl
Gehzeit [h]	5.75	Asphaltstraße	0	Sehenswürdigkeit	0
Weglänge (horizontal) [m]	12700	Schotterweg	8	See	0
Höhenunterschied [m]	1230	steiniger Weg	43	schöner Weg	1
durchschnittliche Steigung [%]	11	sandiger/erdiger Weg	0	schöne Aussicht	0
durchschnittliche Höhe [m]	2400	Wiesenweg	3	ruhig	5
relative Steigung [%]	11	Waldpfad	0	über Waldgrenze	3
Weglänge (schräg) [m]	13300	steiniger Pfad	13	im Wald	2

totaler Höhenunterschied [m]	1772	Standardpfad	17	am Bach	1
Vorgängerknoten	8	Pfad am Fels	0	Almhütte entlang des Weges	3
Labelnummer des Vorgängers	1	Wiesenpfad	16	Grad entlang	0
Labelnummer des aktuellen Knoten	1	-		Brücke/Überqueren eines Baches	3
-		-		leicht bewaldet	2

Tabelle 42: Ergebnis Labelvektor Lösung Nr.2 (Zwenewaldweg-Leisacher Alm)

Kriterien		Untergrund	[%]	POI's	Anzahl
Gehzeit [h]	5.75	Asphaltstraße	0	Sehenswürdigkeit	1
Weglänge (horizontal) [m]	10700	Schotterweg	23	See	0
Höhenunterschied [m]	1230	steiniger Weg	6	schöner Weg	1
durchschnittliche Steigung [%]	10	sandiger/erdiger Weg	0	schöne Aussicht	2
durchschnittliche Höhe [m]	2300	Wiesenweg	2	ruhig	6
relative Steigung [%]	8	Waldpfad	5	über Waldgrenze	2
Weglänge (schräg) [m]	11100	steiniger Pfad	18	im Wald	2
totaler Höhenunterschied [m]	1817	Standardpfad	29	am Bach	5
Vorgängerknoten	13	Pfad am Fels	0	Almhütte entlang des Weges	5
Labelnummer des Vorgängers	1	Wiesenpfad	17	Grad entlang	0
Labelnummer des aktuellen Knoten	2	-		Brücke/Überqueren eines Baches	4
-		-		leicht bewaldet	6

Tabelle 43: Ergebnis Labelvektor Lösung Nr.3 (Zwenewaldweg-Leisacher Alm)

Kriterien		Untergrund	[%]	POI's	Anzahl
Gehzeit [h]	5.75	Asphaltstraße	0	Sehenswürdigkeit	1
Weglänge (horizontal) [m]	10600	Schotterweg	23	See	0
Höhenunterschied [m]	1230	steiniger Weg	6	schöner Weg	1
durchschnittliche Steigung [%]	10	sandiger/erdiger Weg	0	schöne Aussicht	3

durchschnittliche Höhe [m]	2300	Wiesenweg	0	ruhig	6
relative Steigung [%]	9	Waldpfad	5	über Waldgrenze	2
Weglänge (schräg) [m]	11100	steiniger Pfad	18	im Wald	2
totaler Höhenunterschied [m]	1821	Standardpfad	29	am Bach	5
Vorgängerknoten	13	Pfad am Fels	0	Almhütte entlang des Weges	5
Labelnummer des Vorgängers	2	Wiesenpfad	19	Grad entlang	0
Labelnummer des aktuellen Knoten	3	-		Brücke/Überqueren eines Baches	3
-		-		leicht bewaldet	6

Tabelle 44: Ergebnis Labelvektor Lösung Nr.4 (Zwenewaldweg-Leisacher Alm)

Kriterien		Untergrund	[%]	POI's	Anzahl
Gehzeit [h]	5.75	Asphaltstraße	0	Sehenswürdigkeit	0
Weglänge (horizontal) [m]	12300	Schotterweg	0	See	0
Höhenunterschied [m]	1230	steiniger Weg	41	schöner Weg	0
durchschnittliche Steigung [%]	11	sandiger/erdiger Weg	0	schöne Aussicht	6
durchschnittliche Höhe [m]	2600	Wiesenweg	0	ruhig	10
relative Steigung [%]	10	Waldpfad	0	über Waldgrenze	7
Weglänge (schräg) [m]	13100	steiniger Pfad	23	im Wald	3
totaler Höhenunterschied [m]	2197	Standardpfad	27	am Bach	0
Vorgängerknoten	13	Pfad am Fels	2	Almhütte entlang des Weges	3
Labelnummer des Vorgängers	6	Wiesenpfad	7	Grad entlang	4
Labelnummer des aktuellen Knoten	7	-		Brücke/Überqueren eines Baches	1
-		-		leicht bewaldet	1

Es werden in diesem Beispiel alle vier Pareto-optimalen Wege als effiziente Lösungen angesehen. Man könnte nun noch weitere Parameter berücksichtigen. Beispielsweise könnte man nun aus den vier Lösungen jene Lösung wählen, bei der die Bewertung des POI „über Waldgrenze“ am besten ist. Für diesen Fall würde die Lösung Nr. 4 gewählt werden.

Dies kann für alle weiteren Kriterien durchgespielt werden und somit ergeben sich unterschiedliche, optimale Lösungen.

8.4 Fallbeispiel 4

Ausgangslage: Wir wollen vom Startpunkt St. Jakob Schule (Knotennummer 47) auf die Weiße Spitze (Knotennummer 59). Wir verwenden als Kriterien die Weglänge (schräg), den totalen Höhenunterschied und die Gehzeit und wollen alle Pareto-optimalen Lösungen finden. Zur Bestimmung effizienter Lösungen wird die Definition 2 (Kapitel 4.2.2) verwendet. Somit entstehen zwei Pareto-optimale Lösungen.

Lösung 1

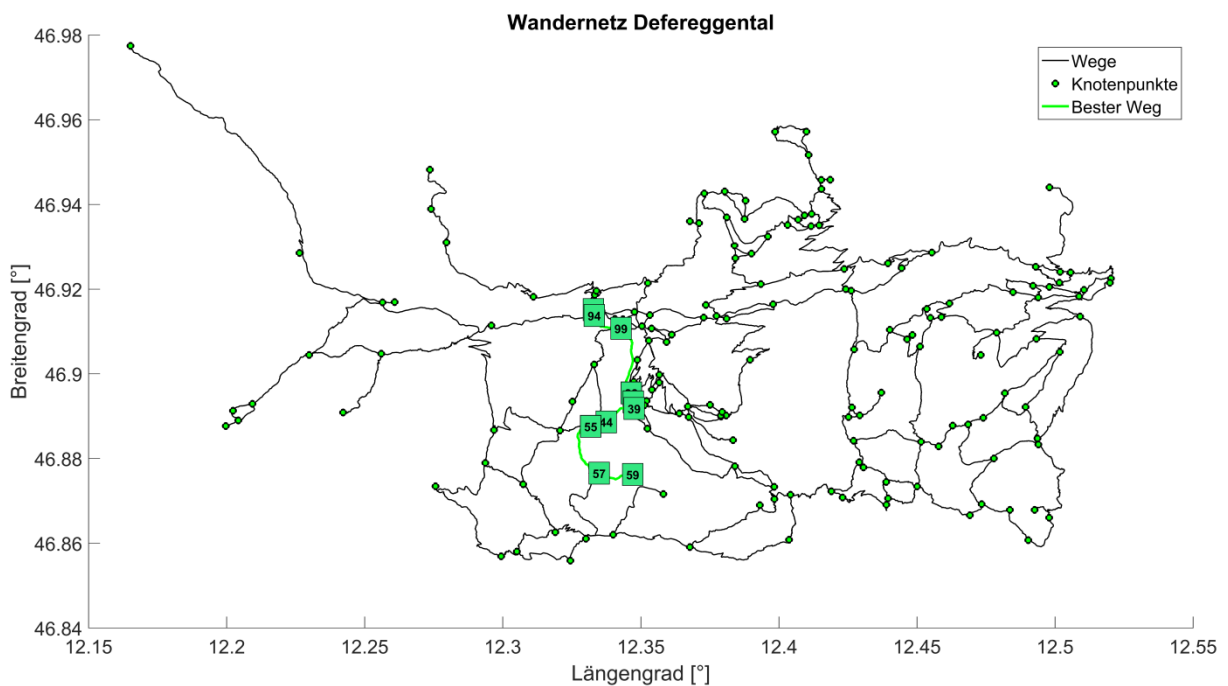


Abbildung 72: Übersichtskarte - optimale Route Nr.1 (St. Jakob Schule-Weiße Spitze)

In den Abbildungen 72 und 73 sind die Routenverläufe dargestellt.

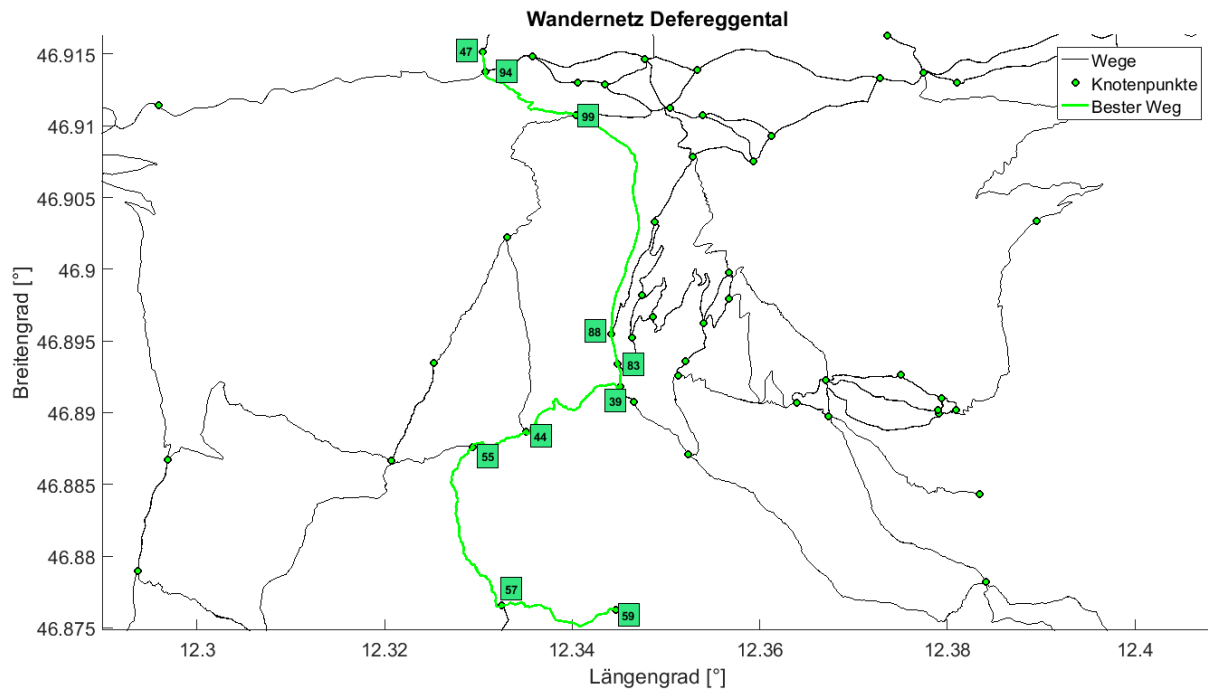


Abbildung 73: Ausschnitt aus Übersichtskarte - optimale Route Nr.1 (St. Jakob Schule-Weiße Spitze)

Die optimale Route Nr. 1 verläuft ausgehend von der Schule St. Jakob über den Knoten Wassererlebnisweg (48), den Knoten Chr.-Erlsbacher-Steig (99), den Knoten Erlsbachersteig (88), den Knoten Abzweigung Eggenalm (83), den Knoten Brugger Alm (39), den Knoten Ragoetzlalm (44), den Knoten Ragoetzlenke (55) und den Knoten Weiße Spitze (57) auf die Weiße Spitze (59).

Tabelle 45: Ergebnis Labelvektor Lösung Nr.1 (St. Jakob Schule-Weiße Spitze)

Kriterien		Untergrund	[%]	POI`s	Anzahl
Gehzeit [h]	6	Asphaltstraße	0	Sehenswürdigkeit	0
Weglänge (horizontal) [m]	8700	Schotterweg	7	See	1
Höhenunterschied [m]	1572	steiniger Weg	0	schöner Weg	2
durchschnittliche Steigung [%]	22	sandiger/erdiger Weg	0	schöne Aussicht	1
durchschnittliche Höhe [m]	2600	Wiesenweg	3	ruhig	8
relative Steigung [%]	20	Waldpfad	21	über Waldgrenze	3
Weglänge (schräg) [m]	9000	steiniger Pfad	21	im Wald	2
totaler Höhenunterschied [m]	1592	Standardpfad	24	am Bach	1
Vorgängerknoten	57	Pfad am Fels	10	Almhütte entlang des Weges	0
Labelnummer des Vorgängers	1	Wiesenpfad	14	Grad entlang	1
Labelnummer des aktuellen Knoten	1	-		Brücke/Überqueren eines Baches	4
-		-		leicht bewaldet	2

Als Ergebnis (Tabelle 45) erhält man eine Gehzeit von 6 h, einen Höhenunterschied von 1572 m, einen totalen Höhenunterschied von 1592 m, eine Weglänge (schräg) von 9.0 km, eine durchschnittliche Steigung von 22 % und die durchschnittliche Höhe dieser Route beträgt 2600 m.

Lösung 2

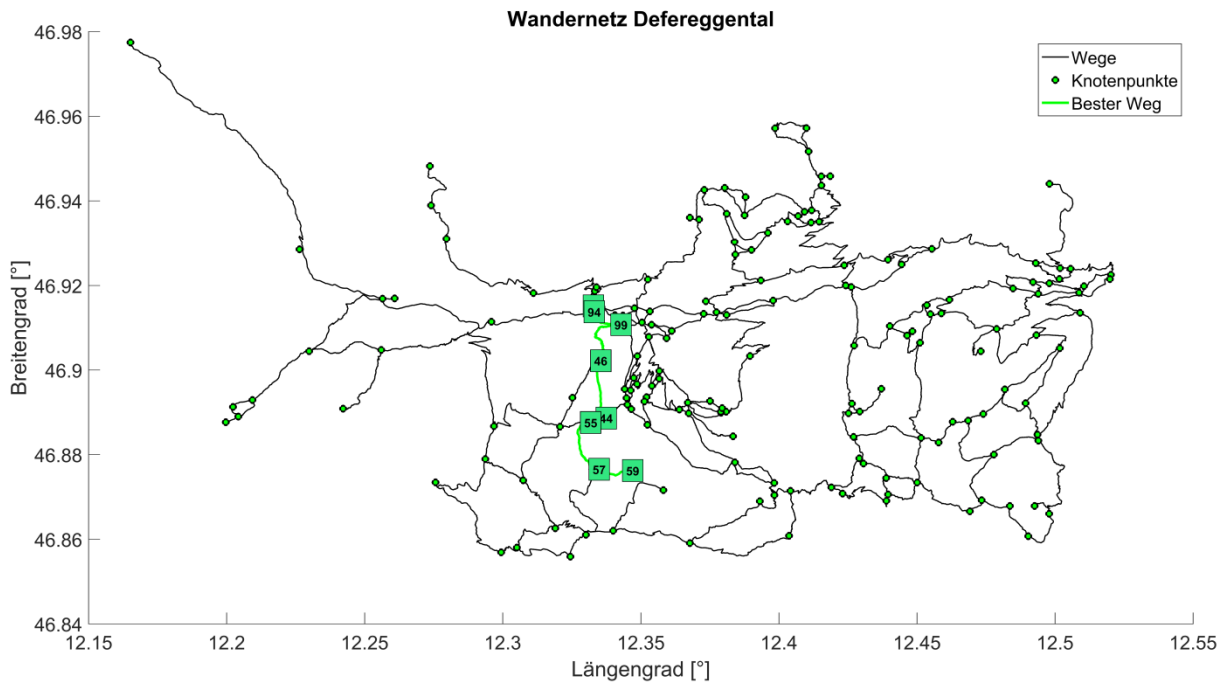


Abbildung 74: Übersichtskarte - optimale Route Nr.2 (St. Jakob Schule-Weiße Spitze)

In den Abbildungen 74 und 75 sind die Routenverläufe dargestellt.

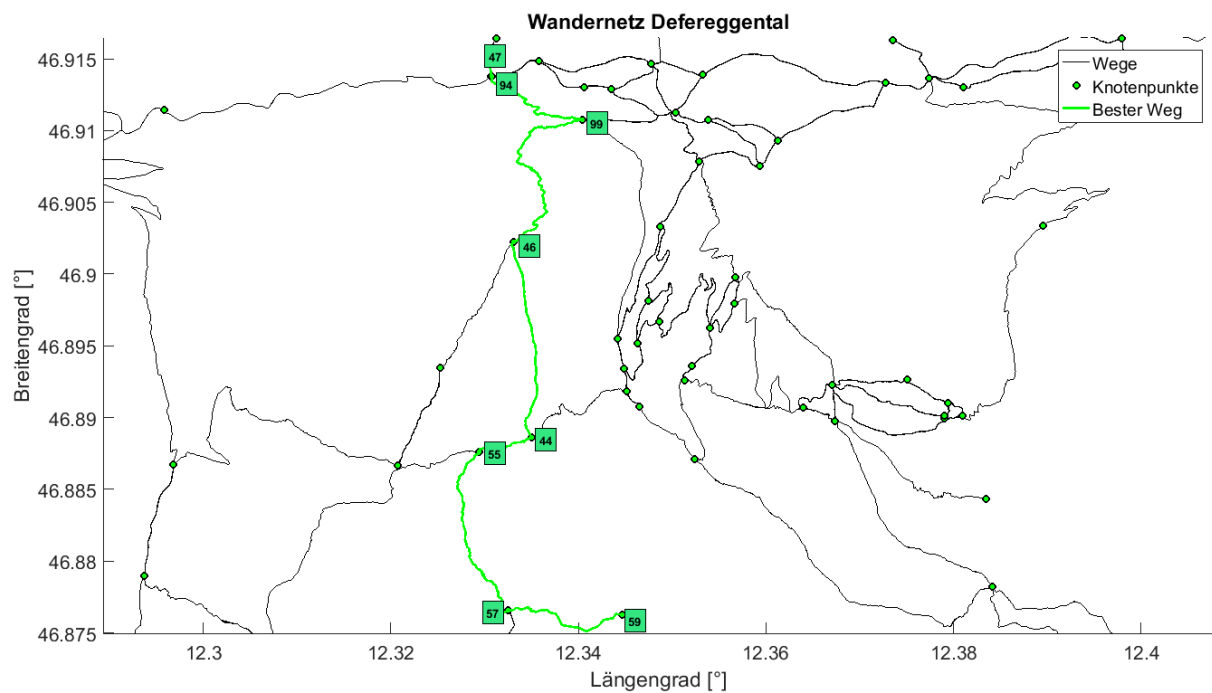


Abbildung 75: Ausschnitt aus Übersichtskarte - optimale Route Nr.2 (St. Jakob Schule-Weiße Spitze)

Die optimale Route Nr. 2 verläuft ausgehend von der Schule St. Jakob über den Knoten Wassererlebnisweg (48), den Knoten Chr.-Erlsbacher-Steig (99), den Knoten Wetterkreuz (46), den Knoten Ragoetzalm (44), den Knoten Ragoetzlenke (55) und den Knoten Weiße Spitze (57) auf die Weiße Spitze (59).

Tabelle 46: Ergebnis Labelvektor Lösung Nr.2 (St. Jakob Schule-Weiße Spitze)

Kriterien		Untergrund	[%]	POI`s	Anzahl
Gehzeit [h]	6	Asphaltstraße	0	Sehenswürdigkeit	0
Weglänge (horizontal) [m]	8500	Schotterweg	3	See	1
Höhenunterschied [m]	1572	steiniger Weg	0	schöner Weg	2
durchschnittliche Steigung [%]	18	sandiger/erdiger Weg	0	schöne Aussicht	1
durchschnittliche Höhe [m]	2600	Wiesenweg	0	ruhig	6
relative Steigung [%]	16	Waldpfad	18	über Waldgrenze	4
Weglänge (schräg) [m]	8900	steiniger Pfad	21	im Wald	2
totaler Höhenunterschied [m]	1809	Standardpfad	33	am Bach	0
Vorgängerknoten	57	Pfad am Fels	11	Almhütte entlang des Weges	0
Labelnummer des Vorgängers	2	Wiesenspfad	14	Grad entlang	1
Labelnummer des aktuellen Knoten	2	-		Brücke/Überqueren eines Baches	2
-		-		leicht bewaldet	0

Als Ergebnis (Tabelle 46) erhält man eine Gehzeit von 6 h, einen Höhenunterschied von 1572 m, einen totalen Höhenunterschied von 1809 m, eine Weglänge (schräg) von 8.9 km, eine durchschnittliche Steigung von 18 % und die durchschnittliche Höhe dieser Route beträgt 2600 m.

Vergleich der beiden Routenverläufe

In den Abbildungen 76 bis 79 werden die Routenverläufe gegenübergestellt. Es wird jeweils ein anderer Baselayer verwendet.

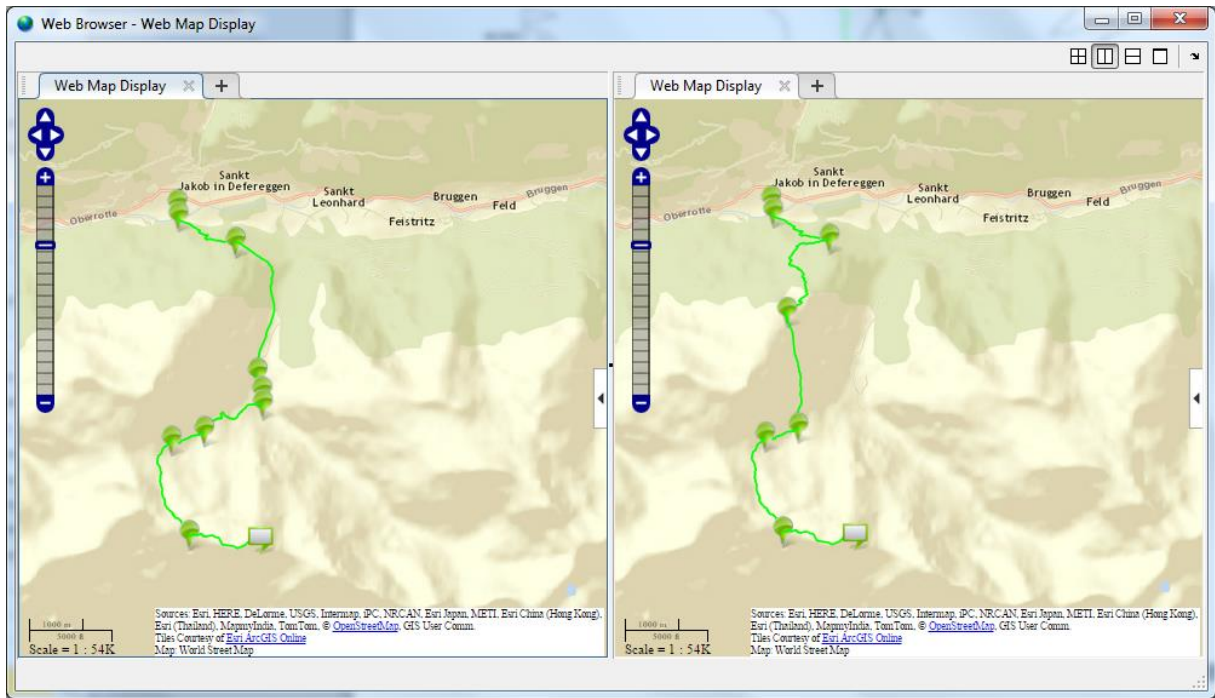


Abbildung 76: WMS Ausschnitt (World Street Map) - Vergleich der optimalen Routen

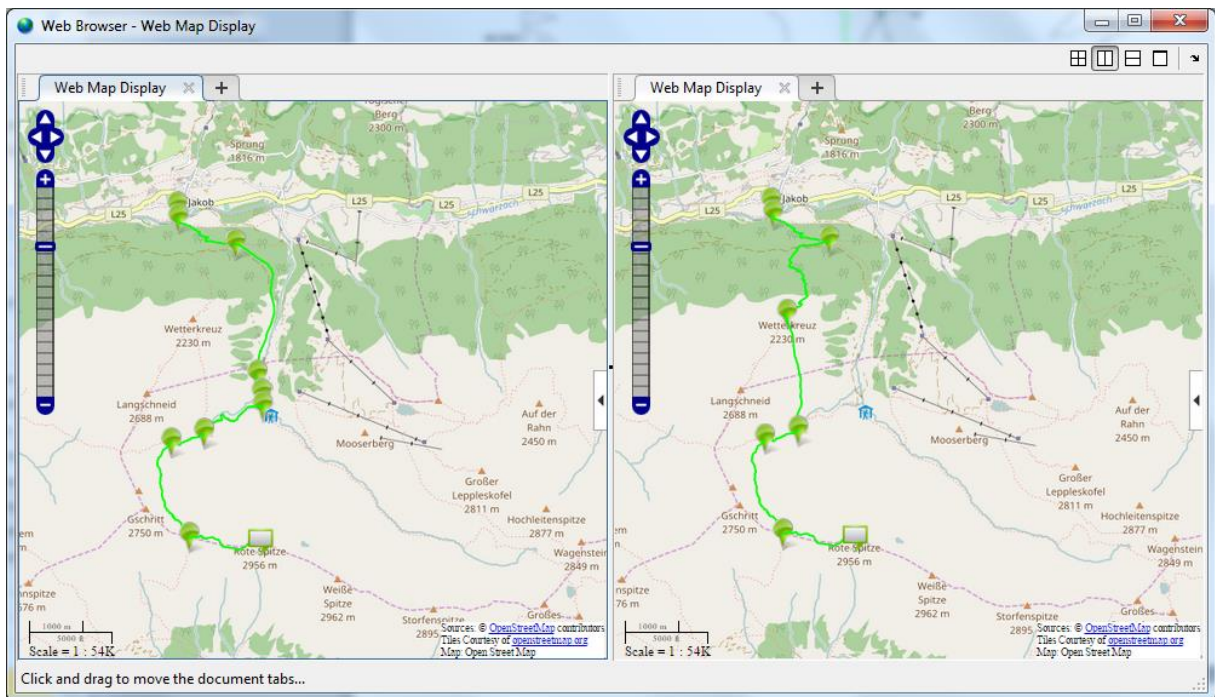


Abbildung 77: WMS Ausschnitt (Open Street Map) - Vergleich der optimalen Routen

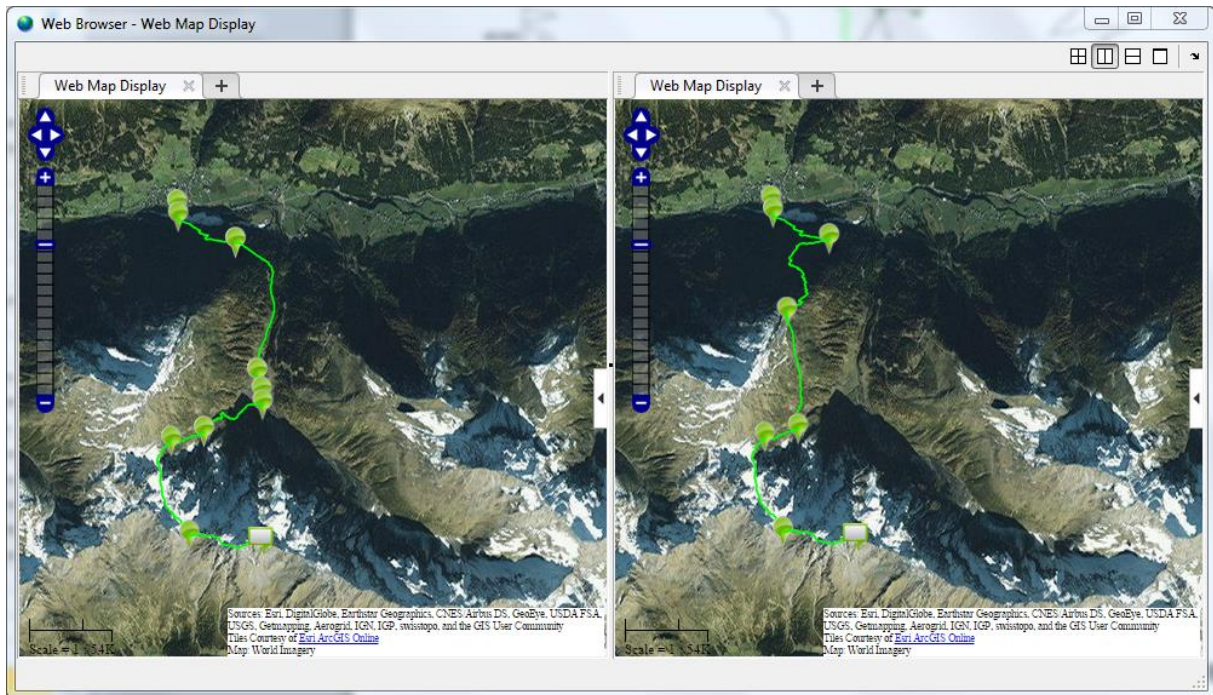


Abbildung 78: WMS Ausschnitt (World Imagery) - Vergleich der optimalen Routen

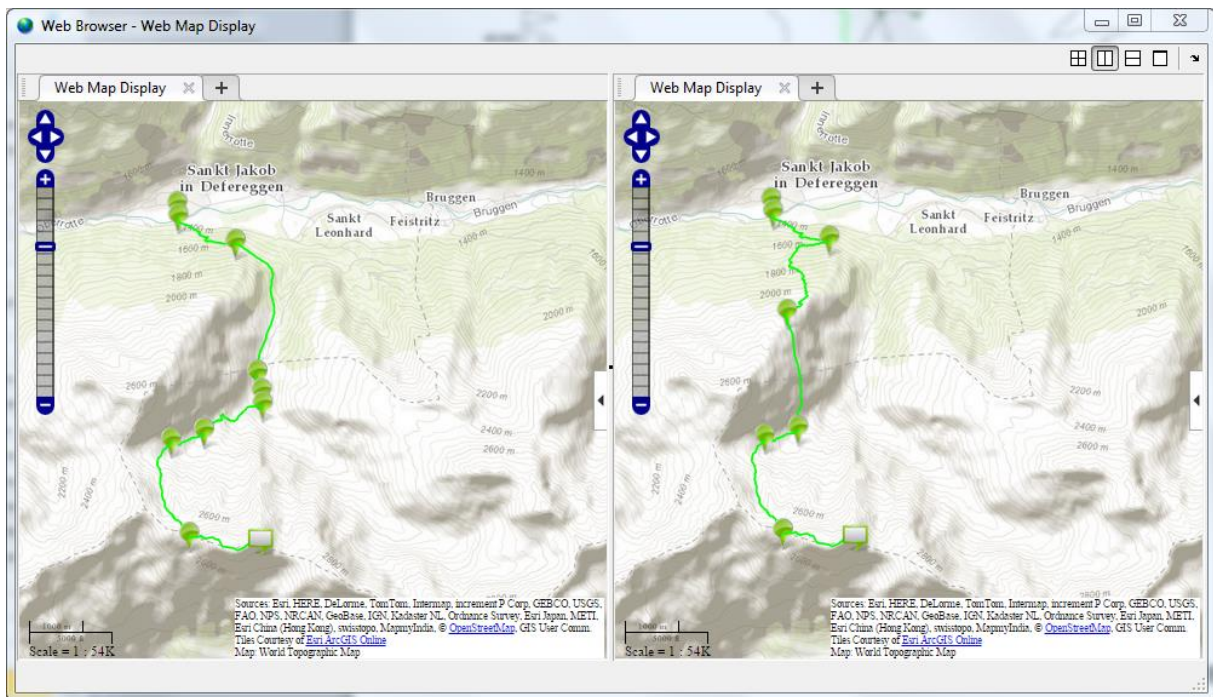


Abbildung 79: WMS Ausschnitt (World Topographic Map) - Vergleich der optimalen Routen

8.5 Fallbeispiel 5

Ausgangslage: Wir wollen von der Talstation Brunnalm (Knotennummer 93) alle Wanderwege, mit einer Gehzeit zwischen 3 und 4 Stunden. Es wird also die Gehzeit als Hauptkriterium verwendet, als Nebenkriterium die Weglänge und es werden alle Wege, die eine Gehzeit zwischen 3 und 4 Stunden aufweisen, angezeigt.

Ergebnis: Dies ergibt 45 Lösungen. Aus diesen Wegen wählen wir nun als erstes denjenigen, mit den meisten Hütten entlang des Weges. Dies ergibt folgende Lösung:

Der Weg verläuft vom Startknoten Brunnalm Talstation (93) bis zum Zielknoten Knoten Großer Leppleskofl (32). Der Weg führt ausgehend von der Talstation (93) über den Knoten Alte Abfahrt (98), den Knoten Brunnalm (79) und den Knoten Mooseralm (34) zum Knoten Großer Leppleskofel (32).

In den Abbildungen 80 und 81 sind eine Übersichtskarte und ein Ausschnitt aus der Übersichtskarte dargestellt.

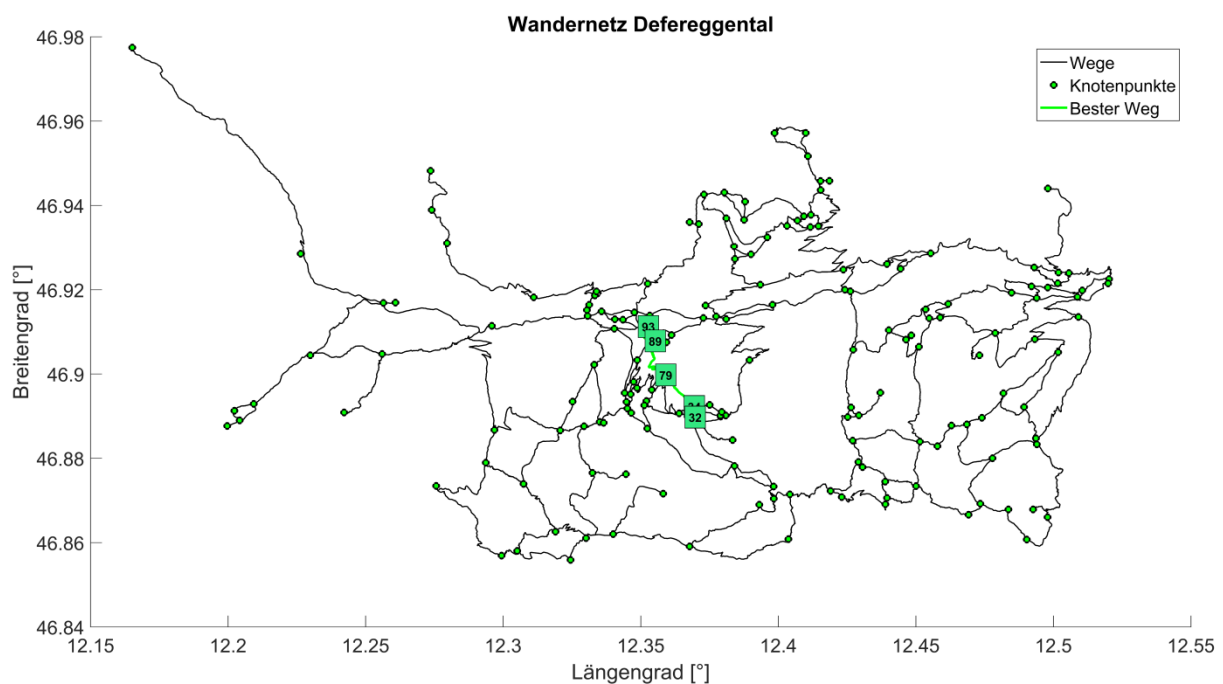


Abbildung 80: Übersichtskarte - optimale Route Nr.1 (Brunnalm Talstation-Knoten Großer Leppleskofl)

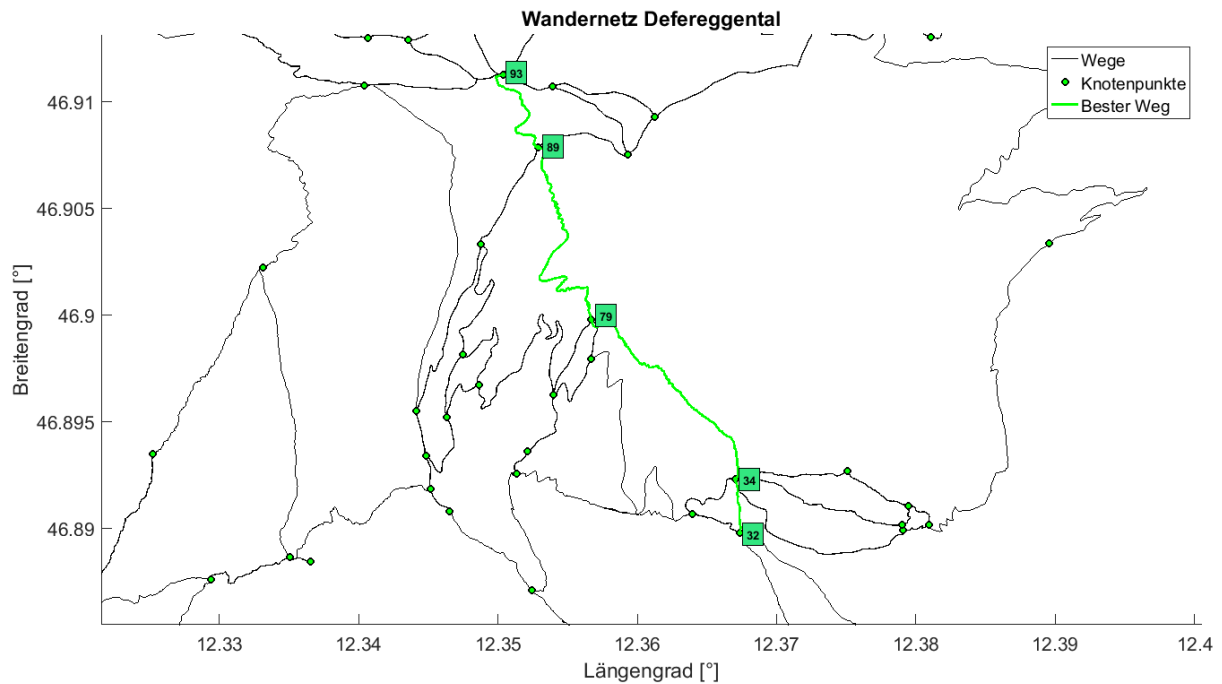


Abbildung 81: Ausschnitt aus Übersichtskarte - optimale Route Nr.1 (Brunnalm Talstation-Knoten Großer Leppleskofl)

Tabelle 47: Ergebnisse (Gehzeit & geometrische Kriterien) der optimalen Route Nr.1 (Brunnalm Talstation-Knoten Großer Leppleskofl)

Knoten	93	89	79	34	32
Gehzeit [<i>h</i>]	0	0.75	2.25	3.25	3.5
Weglänge (horizontal) [<i>m</i>]	0	700	2500	4200	4500
Höhenunterschied [<i>m</i>]	0	206	655	961	968
durchschnittliche Steigung [%]	0	30	28	21	20
durchschnittliche Höhe [<i>m</i>]	1400	1500	1700	2000	2200
relative Steigung [%]	0	28	26	21	20
Weglänge (schräg) [<i>m</i>]	0	800	2700	4400	4700
totaler Höhenunterschied [<i>m</i>]	0	206	657	970	985

In den Tabelle 47 und 48 sind die Ergebnisse dargestellt. Als Ergebnis erhält man eine Gehzeit von 3.5 h, einen Höhenunterschied von 968 m, eine Weglänge (schräg) von 4.7 km, eine durchschnittliche Steigung von 20 % und die durchschnittliche Höhe dieser Route beträgt 2200 m. Es gibt zwei Hütten entlang dieses Weges und deshalb ist diese Lösung die „beste“.

Tabelle 48: Ergebnisse (Untergrund & POI's) der optimalen Route Nr.1 (Brunnalm Talstation-Knoten Großer Leppleskofl)

Untergrund	[%]	POI's	Anzahl
Asphaltstraße	1	Sehenswürdigkeit	0
Schotterweg	1	See	0
steiniger Weg	0	schöner Weg	0
sandiger/erdiger Weg	0	schöne Aussicht	1
Wiesenweg	6	ruhig	3
Waldpfad	10	über Waldgrenze	1
steiniger Pfad	0	im Wald	1
Standardpfad	60	am Bach	0
Pfad am Fels	0	Almhütte entlang des Weges	2
Wiesenzpfad	22	Grad entlang	1
-		Brücke/Überqueren eines Baches	0
-		leicht bewaldet	2

In Abbildung 82 ist ein Ausschnitt aus dem WMS mit dem Baselayer World Imagery dargestellt.

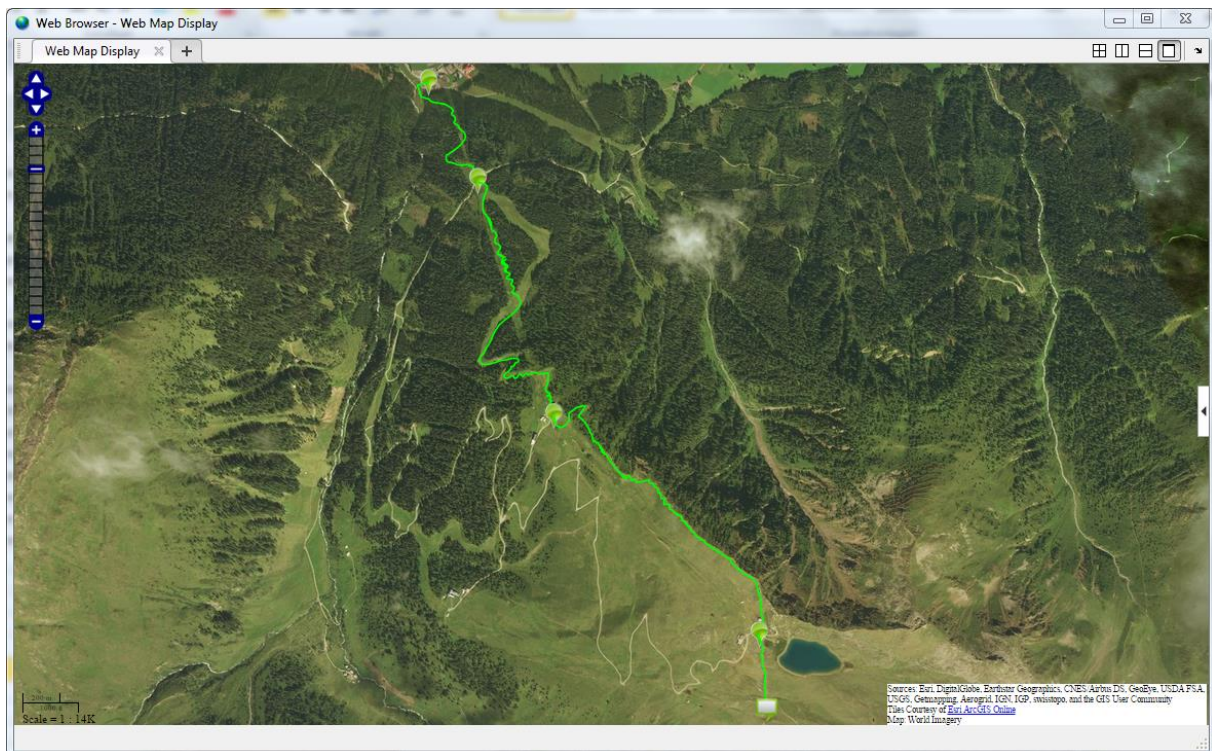


Abbildung 82: Web Map Display (World Imagery) - optimale Route Nr.1 (Brunnalm Talstation-Knoten Großer Leppleskofl)

Wählt man aus den zuvor berechneten 45 Lösungen, die eine Gehzeit zwischen 3 und 4 Stunden aufweisen, diejenige Lösung, die den größten Höhenunterschied aufweist, so ergibt dies folgende Lösung:

Der Weg verläuft vom Startknoten Brunnalm Talstation (93) bis zum Zielknoten Langschneid (80). Der Weg führt ausgehend von der Talstation (93) über den Knoten Chr.-Erlsbacher-Steig

(99) und den Knoten Wetterkreuz (46) zum Knoten Langschneid (80). Dieser Verlauf ist in den Abbildungen 83, 84 und 85 dargestellt.

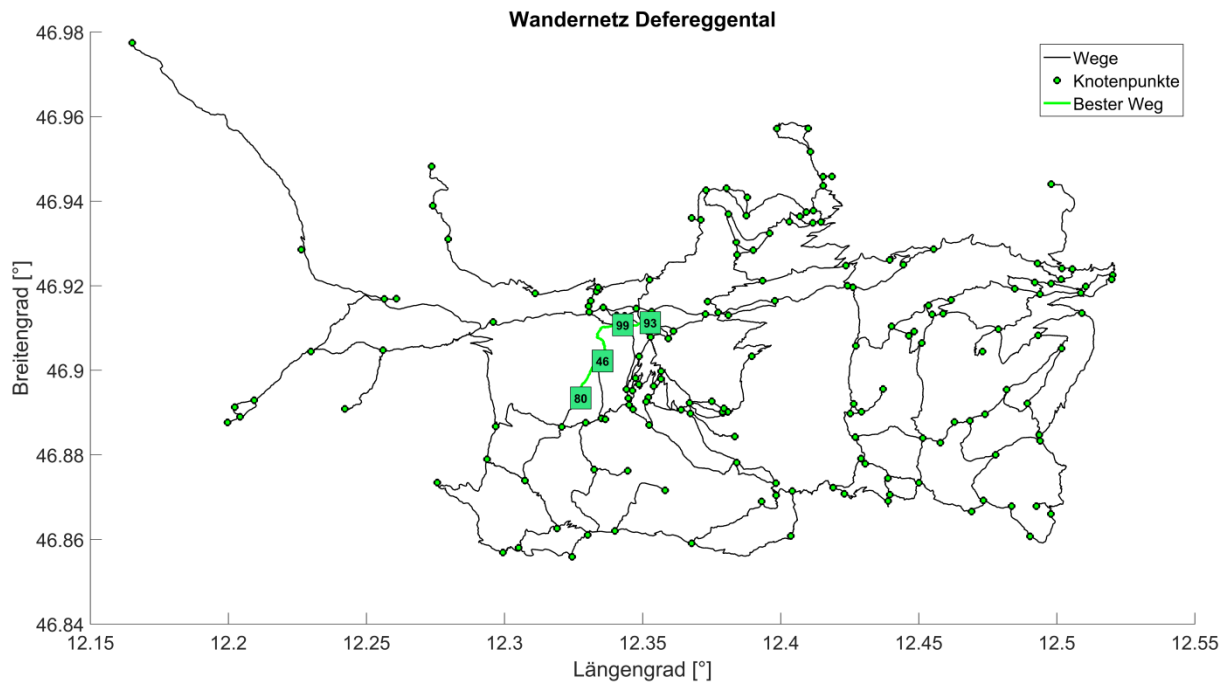


Abbildung 83: Übersichtskarte - optimale Route Nr.2 (Brunnalm Talstation-Langschneid)

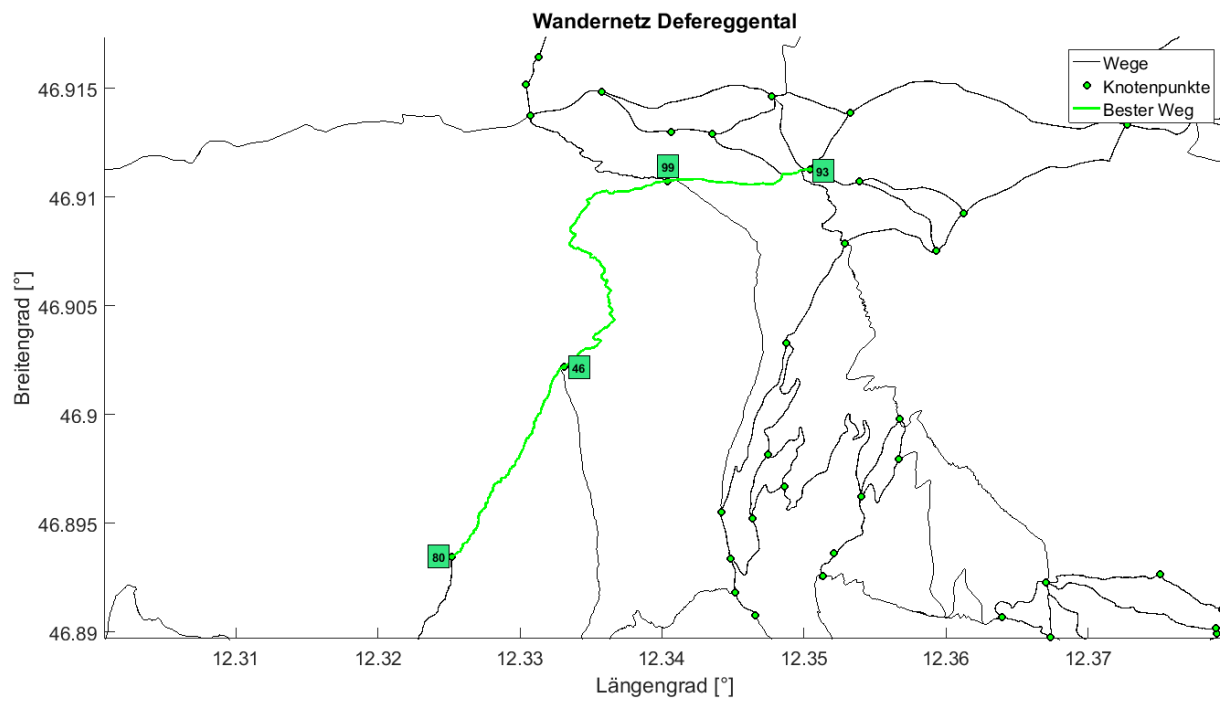


Abbildung 84: Ausschnitt aus Übersichtskarte - optimale Route Nr.2 (Brunnalm Talstation-Langschneid)

Tabelle 49: Ergebnisse (Gehzeit & geometrische Kriterien) der optimalen Route Nr.2 (Brunnalm Talstation-Langschneid)

Knoten	93	99	46	80
Gehzeit [h]	0	0.5	2.5	4
Weglänge (horizontal) [m]	0	800	2700	4000
Höhenunterschied [m]	0	181	830	1288
durchschnittliche Steigung [%]	0	22	29	33
durchschnittliche Höhe [m]	1400	1500	1700	2100
relative Steigung [%]	0	23	29	32
Weglänge (schräg) [m]	0	800	2900	4400
totaler Höhenunterschied [m]	0	181	830	1290

In Tabelle 49 und 50 sind die Ergebnisse dargestellt. Als Ergebnis erhält man eine Gehzeit von 4 h, einen Höhenunterschied von 1288 m, eine Weglänge (schräg) von 4.4 km, eine durchschnittliche Steigung von 33 % und die durchschnittliche Höhe dieser Route beträgt 2100 Höhenmeter. Der Höhenunterschied ist mit 1288 m, der größte Höhenunterschied unter allen Wegen, die eine Gehzeit zwischen 3 und 4 Stunden aufweisen.

Tabelle 50: Ergebnisse (Untergrund & POI's) der optimalen Route Nr.2 (Brunnalm Talstation-Langschneid)

Untergrund	[%]	POI's	Anzahl
Asphaltstraße	1	Sehenswürdigkeit	0
Schotterweg	1	See	0
steiniger Weg	0	schöner Weg	1
sandiger/erdiger Weg	0	schöne Aussicht	0
Wiesenweg	0	ruhig	2
Waldpfad	47	über Waldgrenze	1
steiniger Pfad	16	im Wald	2
Standardpfad	32	am Bach	0
Pfad am Fels	0	Almhütte entlang des Weges	0
Wiesenpfad	3	Grad entlang	0
-		Brücke/Überqueren eines Baches	1
-		leicht bewaldet	0

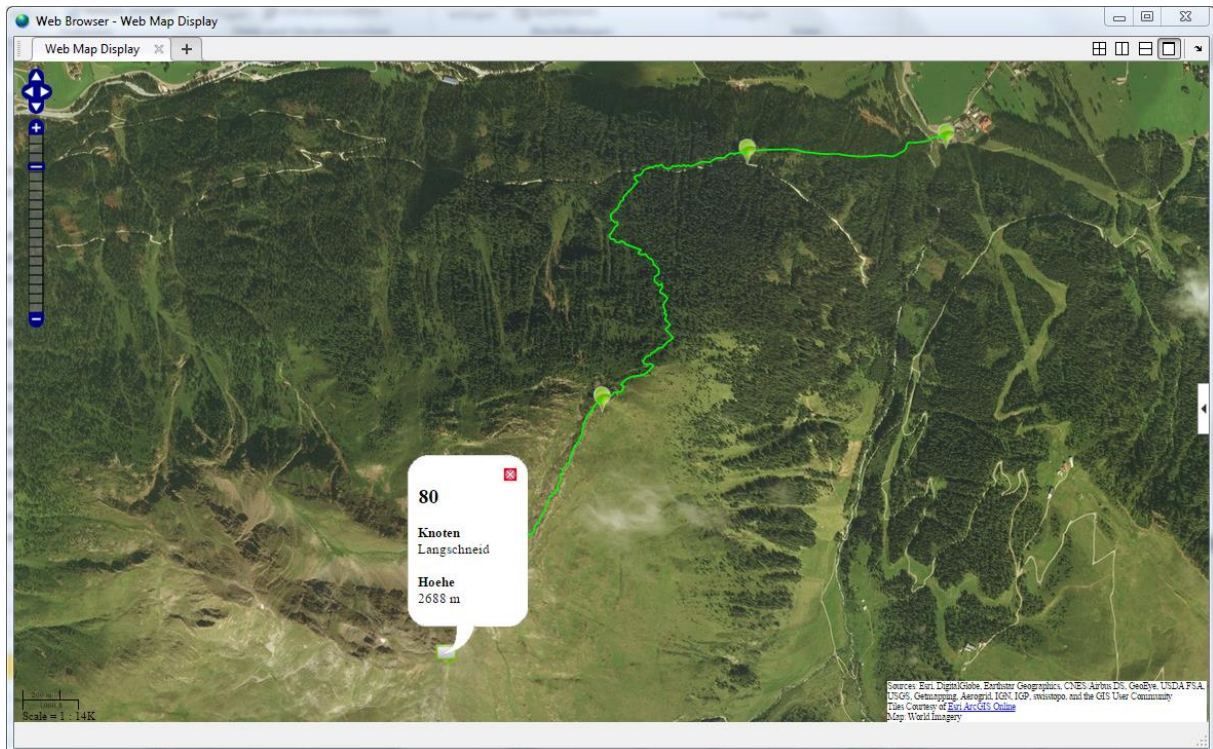


Abbildung 85: Web Map Display (World Imagery) - optimale Route Nr.2 (Brunnalm Talstation-Langschneid)

Dieses Beispiel soll verdeutlichen, welche Lösungsalternativen durch die unterschiedliche Wahl der Kriterien entstehen. Man könnte auch zum Beispiel unter allen Wegen mit einer gewissen Gehzeit, einer gewissen Weglänge oder einem gewissen Höhenunterschied, jene Wege auswählen, die gewisse POI's aufweisen, einen bestimmten Untergrund aufweisen oder anderes.

9 Anwendung

Die Software, mit der die Durchführung dieses Projektes hauptsächlich realisiert wurde, ist *Matlab R2016a*. Um den Routenplanungsalgorithmus nutzen zu können, muss die Software *Matlab R2016a* vorhanden sein. Es wurden 32 Matlab-Files erstellt, um die Durchführung dieses Projekts zu realisieren. Auf einen Teil davon greift der Routenplanungsalgorithmus zu, um die Berechnungen zu absolvieren. Darüber hinaus wird für die Visualisierung ein WMS (Web Map Service) verwendet. Um diesen Service nutzen zu können, muss eine Internetverbindung vorhanden sein. Durch eine graphische Benutzeroberfläche in Matlab können Einstellungen getroffen und Daten eingegeben werden. Als Ergebnisausgabe dient neben den Visualisierungen eine Matlab-Datei und eine Textdatei.

9.1 Software Anforderungen

64-Bit MATLAB, Simulink, and Polyspace Product Families				
Operating Systems	Processors	Disk Space	RAM	Graphics
Windows 10	Any Intel or AMD x86-64 processor	2 GB for MATLAB only, 4–6 GB for a typical installation	2 GB	No specific graphics card is required.
Windows 8.1	AVX2 instruction set support is recommended		With Simulink, 4 GB is required	
Windows 8	With Polyspace, 4 cores is recommended		With Polyspace, 4 GB per core is recommended	Hardware accelerated graphics card supporting OpenGL 3.3 with 1GB GPU memory is recommended.
Windows 7 Service Pack 1				
Windows Server 2016				
Windows Server 2012				
Windows Server 2008 R2 Service Pack 1				

Abbildung 86: Software Anforderungen für Matlab R2016a als 64-Bit Version (für Windows)

In Abbildung 86 sind die Software Anforderungen für *Matlab R2016a* als 64-Bit Version für Windows dargestellt.

64-Bit MATLAB, Simulink, and Polyspace Product Families				
Operating Systems	Processors	Disk Space	RAM	Graphics
macOS Sierra 10.12*	Any Intel x86-64 processor	2 GB for MATLAB only, 4–6 GB for a typical installation	2 GB	No specific graphics card is required.
macOS El Capitan 10.11	AVX2 instruction set support is recommended		With Simulink, 4 GB is required	
macOS Yosemite 10.10	With Polyspace, 4 cores is recommended		With Polyspace, 4 GB per core is recommended	Hardware accelerated graphics card supporting OpenGL 3.3 with 1GB GPU memory is recommended.

Abbildung 87: Software Anforderungen für Matlab R2016a als 64-Bit Version (für Mac)

In Abbildung 87 sind die Software Anforderungen für *Matlab R2016a* als 64-Bit Version für Mac dargestellt. In Abbildung 88 sind die Software Anforderungen für *Matlab R2016a* als 64-Bit Version für Linux dargestellt.

64-Bit MATLAB, Simulink, and Polyspace Product Families				
Operating Systems	Processors	Disk Space	RAM	Graphics
Qualified distributions*: Ubuntu 14.04 LTS and 16.04 LTS Red Hat Enterprise Linux 6 and 7** SUSE Linux Enterprise Desktop 12*** Debian 7.x, 8.x	Any Intel or AMD x86-64 processor AVX2 instruction set support is recommended With Polyspace, 4 cores is recommended	2 GB for MATLAB only, 4-6 GB for a typical installation	2 GB With Simulink, 4 GB is required With Polyspace, 4 GB per core is recommended	No specific graphics card is required. Hardware accelerated graphics card supporting OpenGL 3.3 with 1GB GPU memory is recommended. Use of vendor-supplied proprietary drivers is strongly recommended.

Abbildung 88: Software Anforderungen für Matlab R2016a als 64-Bit Version (für Linux)

9.2 Benutzereingabe

Die Benutzereingabe kann über eine Texteingabe oder über ein graphisches User Interface (GUI) in Matlab erfolgen. Es kann der Start- und Zielpunkt eingegeben werden und es können die Kriterien bestimmt werden. Es müssen mindestens zwei und können maximal vier Kriterien ausgewählt werden. Als Hauptkriterium können nur die Gehzeit, Weglänge (schräg), Weglänge (horizontal) und der totale Höhenunterschied angenommen werden. Als weitere Kriterien können alle definierten Kriterien ausgewählt werden. Zusätzlich muss die Optimierungsmethode ausgewählt werden. Es können eine Einzellösung, eine Teilmenge mit veränderter lexikographischer Optimalität oder alle Pareto-optimalen Lösungen bestimmt werden.



Abbildung 89: GUI - Benutzereingabe in Matlab

In den Abbildungen 89 bis 91 ist die graphische Benutzeroberfläche und eine beispielhafte Eingabe dargestellt. Um die Nummern der Start- und Zielknoten dem realen Wandernetz

zuordnen zu können, werden in der folgenden Tabelle (51) die Knotennummern mit der zugehörigen Knotenbezeichnung aufgelistet:

Tabelle 51: Knotennummern mit Bezeichnungen

Knotennr.	Bezeichnung	Knotennr.	Bezeichnung
1	<i>Start Zwenewaldweg</i>	88	<i>Knoten Erlsbachersteig</i>
2	<i>Knoten Bloßhütte</i>	89	<i>Alte Abfahrt</i>
3	<i>Geigensee</i>	90	<i>Abzweigung Feistritz</i>
4	<i>Pumpersee</i>	91	<i>Leonhardlift</i>
5	<i>Regenstein</i>	92	<i>Feistritzlift</i>
6	<i>Kugelspitze</i>	93	<i>Brunnalm Talstation</i>
7	<i>Villgrater Joch</i>	94	<i>St. Jakob Wassererlebnisweg</i>
8	<i>Knoten Villgrater Joch</i>	95	<i>Knoten Wassererlebnisweg</i>
9	<i>Knoten Gagenalmbach</i>	96	<i>Lacken</i>
10	<i>Zeigerle</i>	97	<i>St. Leonhard</i>
11	<i>Gagenhöhe</i>	98	<i>Feistritz</i>
12	<i>Knoten Gagenhöhe</i>	99	<i>Chr.-Erlsbacher-Steig</i>
13	<i>Kleinitzer Törl</i>	100	<i>Knoten Schwefelquelle</i>
14	<i>Leisacher Alm</i>	101	<i>Schwefelquelle</i>
15	<i>Knoten Volkzeiner Hütte</i>	102	<i>Defereggenghotel</i>
16	<i>Knoten Herz-Ass-Tour</i>	103	<i>Leppetalm</i>
17	<i>Walderalm</i>	104	<i>Knoten Leppetalm</i>
18	<i>Knoten Walderalm</i>	105	<i>Abzweigung Ochsenlacke-Leppetalm</i>
19	<i>Gsaritzer Törl</i>	106	<i>Knoten Ochsenlacke</i>
20	<i>Knoten Ochsenleger</i>	107	<i>Knoten Kl. Leppleskofl</i>
21	<i>Knoten Beilspitze</i>	108	<i>Kl. Leppleskofl</i>
22	<i>Ochsenleger</i>	109	<i>Tönig</i>
23	<i>Gsaritzer Alm</i>	110	<i>Alzenbrunn</i>
24	<i>Beilspitze</i>	111	<i>Zotten Tischlerei</i>
25	<i>Knoten Heinkaralm</i>	112	<i>Zotten</i>
26	<i>Knoten Degenhornsee</i>	113	<i>St. Veit</i>
27	<i>Degenhornsee</i>	114	<i>Speikbodenhütte</i>
28	<i>Großes Degenhorn</i>	115	<i>Abzweigung Froezalm-Speikboden</i>
29	<i>Ochsenlenke</i>	116	<i>Knoten Speikboden</i>
30	<i>Knoten Ochsenlenke</i>	117	<i>Zischkealm</i>
31	<i>Großer Leppleskofl</i>	118	<i>Speikboden</i>
32	<i>Knoten Großer Leppleskofl</i>	119	<i>Proh</i>
33	<i>Knoten Mooseralm</i>	120	<i>Knoten Zischkealm</i>
34	<i>Mooseralm</i>	121	<i>Knoten Speikbodenhütte</i>
35	<i>Weißspitzhütte</i>	122	<i>Gritzer Alm</i>
36	<i>Knoten Weißspitzhütte</i>	123	<i>Knoten Gritzer Alm</i>
37	<i>Knoten Bruggeralmbach</i>	124	<i>Abzweigung Gritzer Alm-Steitenegge</i>
38	<i>Brugger Alm</i>	125	<i>Steitenegge</i>
39	<i>Knoten Brugger Alm</i>	126	<i>Knoten Froditzalm</i>
40	<i>Röte</i>	127	<i>Froditzen</i>
41	<i>Hochalmspitze</i>	128	<i>Knoten Gritzer Hörndle</i>

42	<i>Volkzeiner Hütte</i>	129	<i>Gritzer Hörndle</i>
43	<i>Heinkaralm</i>	130	<i>Gritzer Seen</i>
44	<i>Knoten Ragötzlalm</i>	131	<i>Knoten Froditzten</i>
45	<i>Ragötzlalm</i>	132	<i>Knoten Gasser Hörndle</i>
46	<i>Wetterkreuz</i>	133	<i>Gasser Hörndle</i>
47	<i>St. Jakob Schule</i>	134	<i>Tögisch</i>
48	<i>St. Jakob Ort</i>	135	<i>St. Jakob Abzweigung Tögisch</i>
49	<i>Merschenalm</i>	136	<i>Start Trojer Almtal</i>
50	<i>Mariahilf</i>	137	<i>Trogach Jagerstube</i>
51	<i>Hochkreuzspitze</i>	138	<i>Seespitzhütte</i>
52	<i>Hirschbichl</i>	139	<i>Oberseitsee</i>
53	<i>Erlsbach Parkplatz</i>	140	<i>Seespitze</i>
54	<i>Patscher Hütte</i>	141	<i>Gassen</i>
55	<i>Knoten Ragötzlenke</i>	142	<i>Bruggen</i>
56	<i>Ragötzlenke</i>	143	<i>Görtschach</i>
57	<i>Knoten Weiße Spitze</i>	144	<i>Mellitz</i>
58	<i>Knoten Oberstalleralm</i>	145	<i>St. Veit Außeregg</i>
59	<i>Weiße Spitze</i>	146	<i>Maschlerhof</i>
60	<i>Knoten Villgrater Törl</i>	147	<i>Hof</i>
61	<i>Unterstalleralm</i>	148	<i>Stanzling</i>
62	<i>Knoten Schwarzsee</i>	149	<i>Knoten Maschlerhof</i>
63	<i>Schwarzsee</i>	150	<i>Knoten Hof</i>
64	<i>Knoten Rote Spitze</i>	151	<i>Hopfgarten Oberes Dorf</i>
65	<i>Rote Spitze</i>	152	<i>Innerhopfgarten</i>
66	<i>Beim Garten</i>	153	<i>Hopfgarten Dorfplatz</i>
67	<i>Arntaler Lenke</i>	154	<i>Abzweigung Plon-Innerhopfgarten</i>
68	<i>Villgrater Törl</i>	155	<i>Plon Sportplatz</i>
69	<i>Knoten Hochkreuzspitze</i>	156	<i>Plon Oben</i>
70	<i>Hintere Stalle</i>	157	<i>Plon Abzw. Sportplatz-Schmittenkirchl</i>
71	<i>Deferegger Pfannhorn</i>	158	<i>Bloßhütte</i>
72	<i>Knoten Staller Almbach</i>	159	<i>Knoten Moosalm</i>
73	<i>Obersee</i>	160	<i>Gagenalm</i>
74	<i>Oberseehütte</i>	161	<i>Abzweigung Gagenalm-Bichleram</i>
75	<i>Obersee Parkplatz</i>	162	<i>Knoten Zwenewaldweg</i>
76	<i>Staller Sattel</i>	163	<i>Eggalm</i>
77	<i>Erlsbach Alpenrose</i>	164	<i>Penzenkopf</i>
78	<i>Jagdhausalmen</i>	165	<i>Birinig</i>
79	<i>Brunnalm Bergstation</i>	166	<i>Bernardahütte</i>
80	<i>Langschneid</i>	167	<i>Knoten Kleinitzen</i>
81	<i>Brunnalm Schlepplift</i>	168	<i>Knoten Kleinitzer Almbach</i>
82	<i>Knoten Brunnalm</i>	169	<i>Knoten Plöss</i>
83	<i>Abzweigung Eggenalm</i>	170	<i>Plöss</i>
84	<i>Eggenalm</i>	171	<i>Kleinitzalm</i>
85	<i>Knoten Eggenalm</i>	172	<i>Knoten Kleinitzalm</i>
86	<i>Abkürzung Eggenalm</i>	173	<i>Hoher Gang</i>
87	<i>Familienabfahrt</i>	174	<i>Merschenhöhe</i>

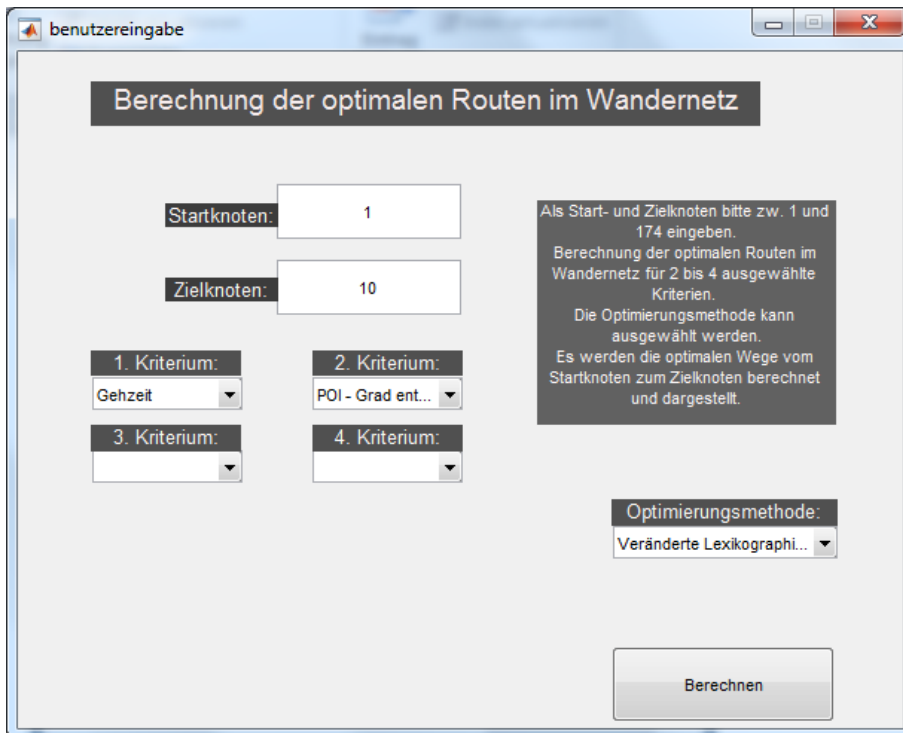


Abbildung 90: GUI - beispielhafte Benutzereingabe (1)

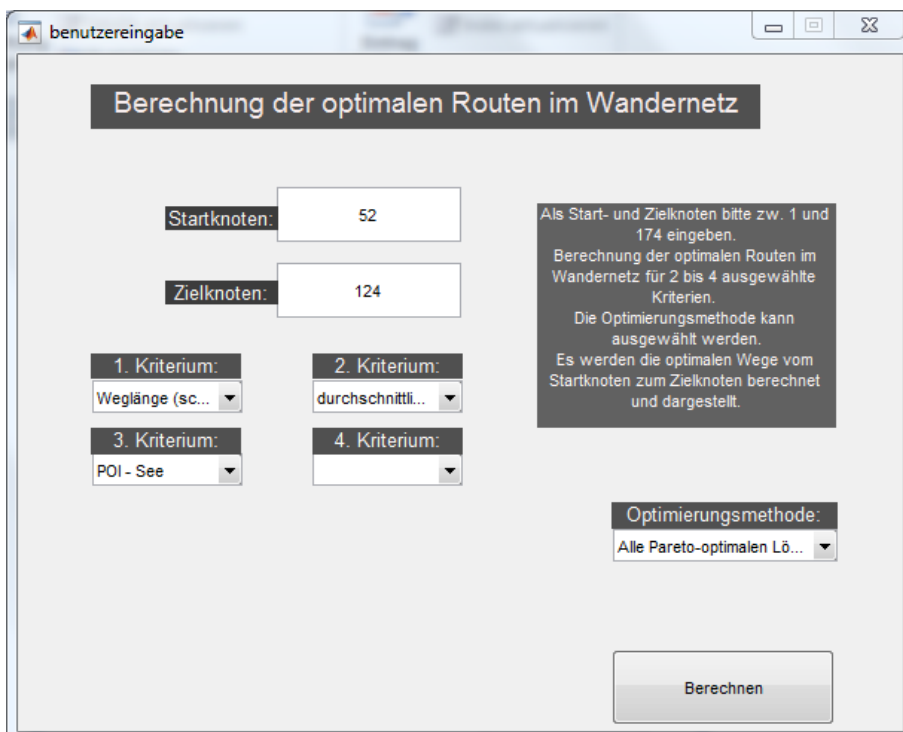
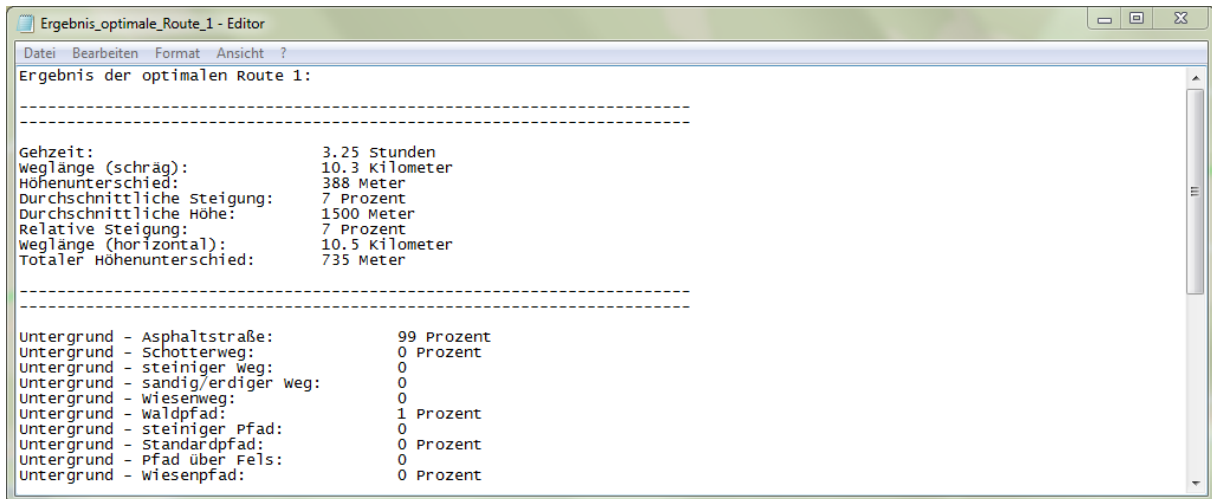


Abbildung 91: GUI - beispielhafte Benutzereingabe (2)

9.3 Benutzerausgabe

Die Ausgabe der berechneten Ergebnisse erfolgt als Textdatei, Matlab-File und in Form von Visualisierungen.

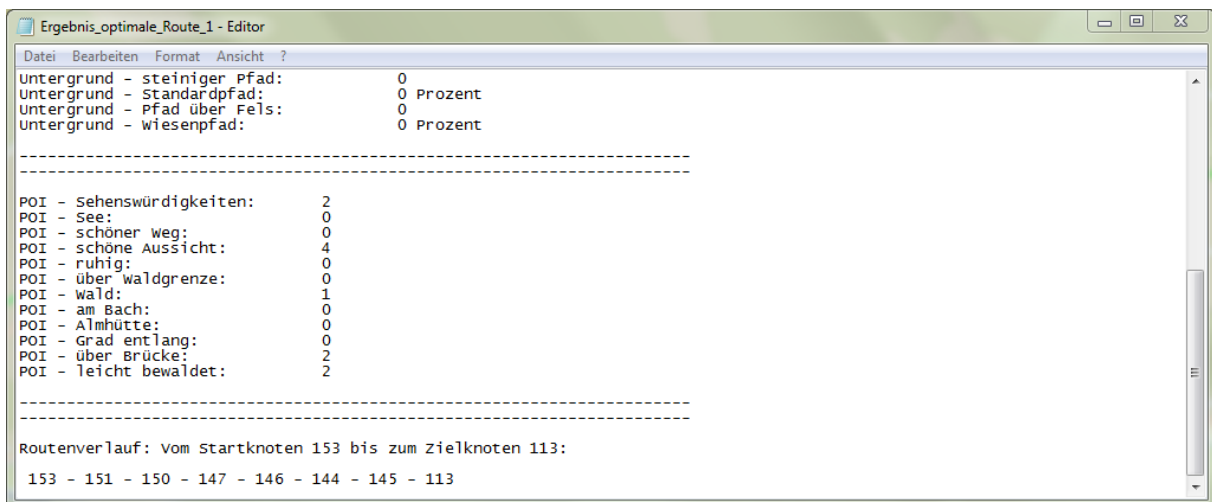
In den Abbildungen 92 bis 94 ist die Textausgabe der berechneten Ergebnisse einer Beispielanwendung dargestellt.



```
Ergebnis der optimalen Route 1:
-----
Gehzeit:                3.25 Stunden
Weglänge (schräg):      10.3 Kilometer
Höhenunterschied:       388 Meter
Durchschnittliche Steigung: 7 Prozent
Durchschnittliche Höhe: 1500 Meter
Relative Steigung:      7 Prozent
Weglänge (horizontal):  10.5 Kilometer
Totaler Höhenunterschied: 735 Meter
-----

Untergrund - Asphaltstraße:    99 Prozent
Untergrund - Schotterweg:       0 Prozent
Untergrund - steiniger weg:    0
Untergrund - sandig/erdiger weg: 0
Untergrund - wiesenweg:        0
Untergrund - waldfpfad:        1 Prozent
Untergrund - steiniger pfad:   0 Prozent
Untergrund - standardpfad:     0 Prozent
Untergrund - pfad über fels:   0
Untergrund - wiesenspfad:      0 Prozent
```

Abbildung 92: Benutzerausgabe - Textdatei (1)



```
Untergrund - steiniger Pfad:    0
Untergrund - Standardpfad:     0 Prozent
Untergrund - Pfad über Fels:    0
Untergrund - wiesenspfad:      0 Prozent
-----

POI - Sehenswürdigkeiten:      2
POI - See:                     0
POI - schöner weg:             0
POI - schöne Aussicht:        4
POI - ruhig:                   0
POI - über waldgrenze:        0
POI - wald:                    1
POI - am Bach:                 0
POI - Almhütte:                0
POI - Grad entlang:            0
POI - über Brücke:             2
POI - leicht bewaldet:         2
-----

Routenverlauf: vom Startknoten 153 bis zum Zielknoten 113:
153 - 151 - 150 - 147 - 146 - 144 - 145 - 113
```

Abbildung 93: Benutzerausgabe - Textdatei (2)

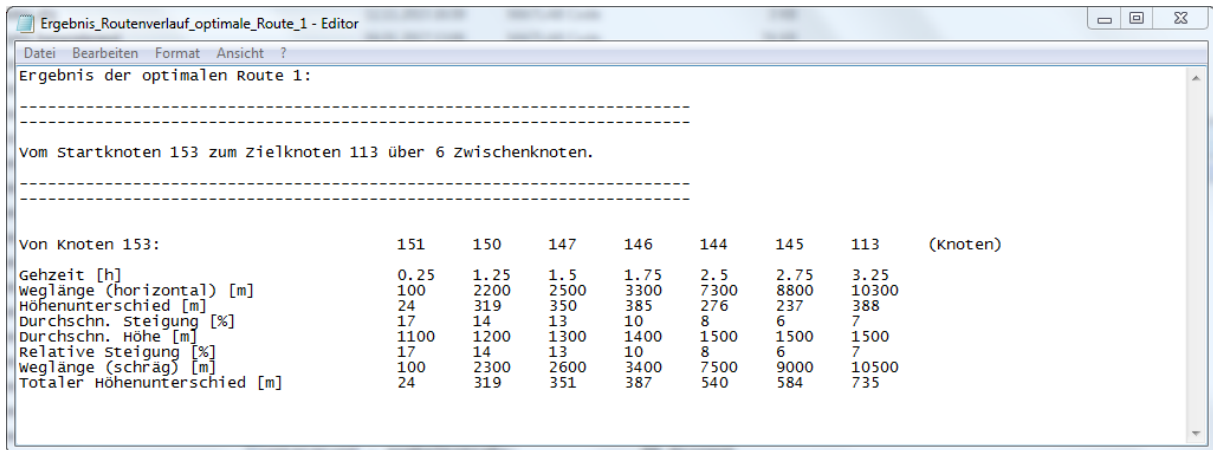


Abbildung 94: Benutzerausgabe - Textdatei (3)

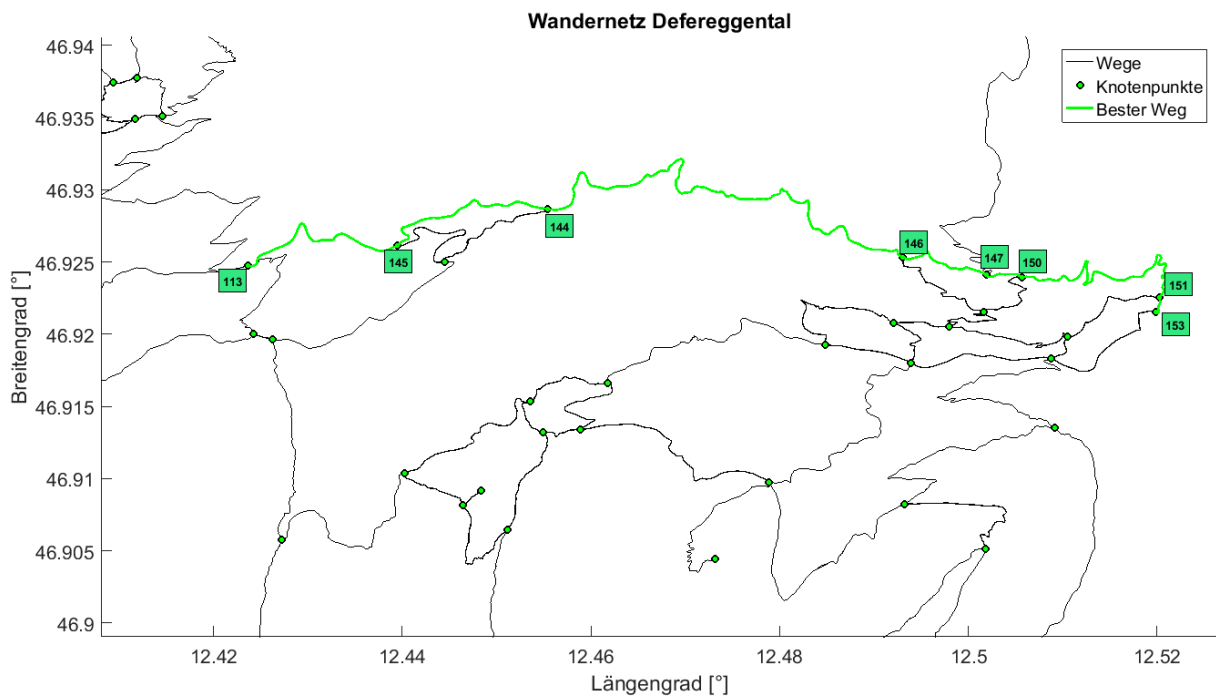


Abbildung 95: Benutzerausgabe - Übersichtskarte (Ausschnitt)

In den Abbildungen 95 bis 100 sind die visualisierten Ergebnisse dieser Beispielanwendung dargestellt.

Wandernetz Defereggental

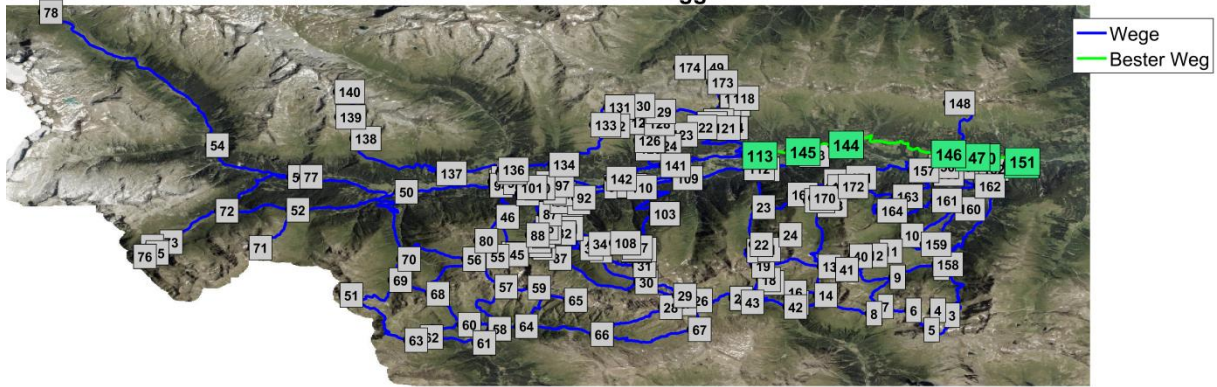


Abbildung 96: Benutzerausgabe - Routenverlauf im Wandernetz (grün)

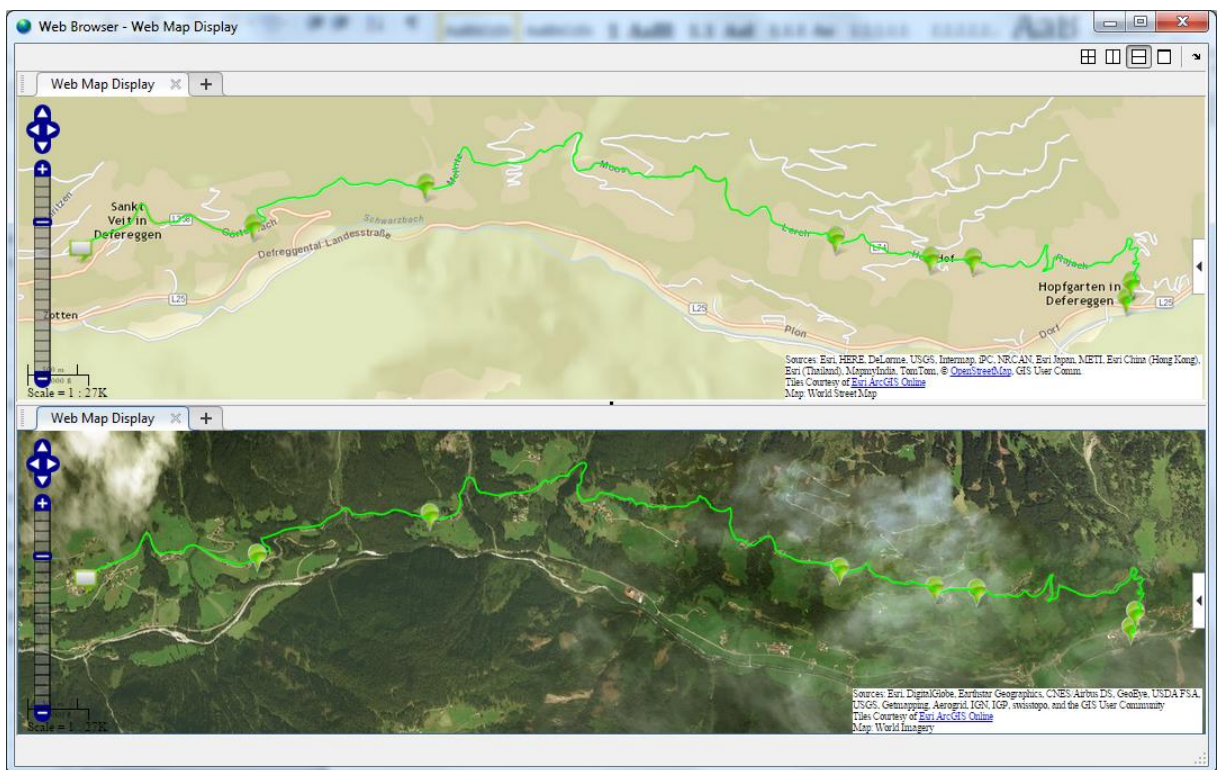


Abbildung 97: Benutzerausgabe - Routenverlauf im WMS (Vergleich zweier unterschiedlicher Baselayer (1))

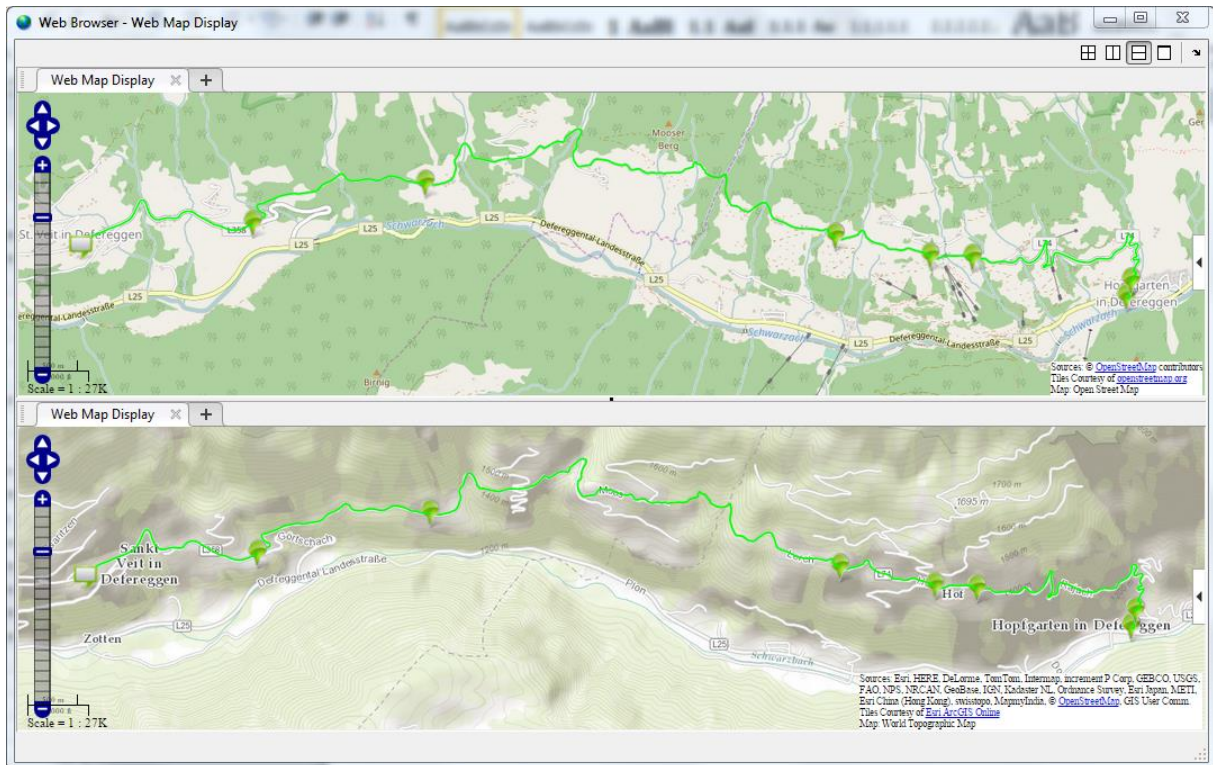


Abbildung 98: Benutzerausgabe - Routenverlauf im WMS (Vergleich zweier unterschiedlicher Baselayer (2))

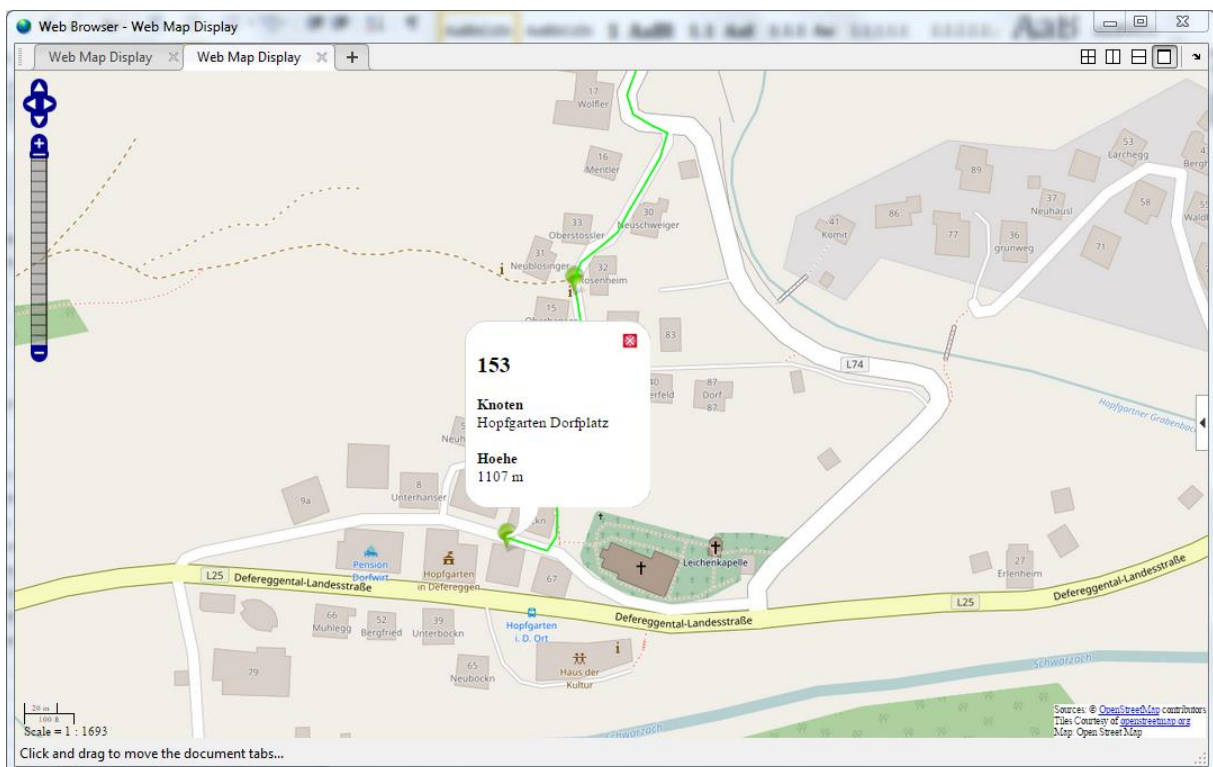


Abbildung 99: Benutzerausgabe - Ausschnitt aus dem WMS (Startknoten)

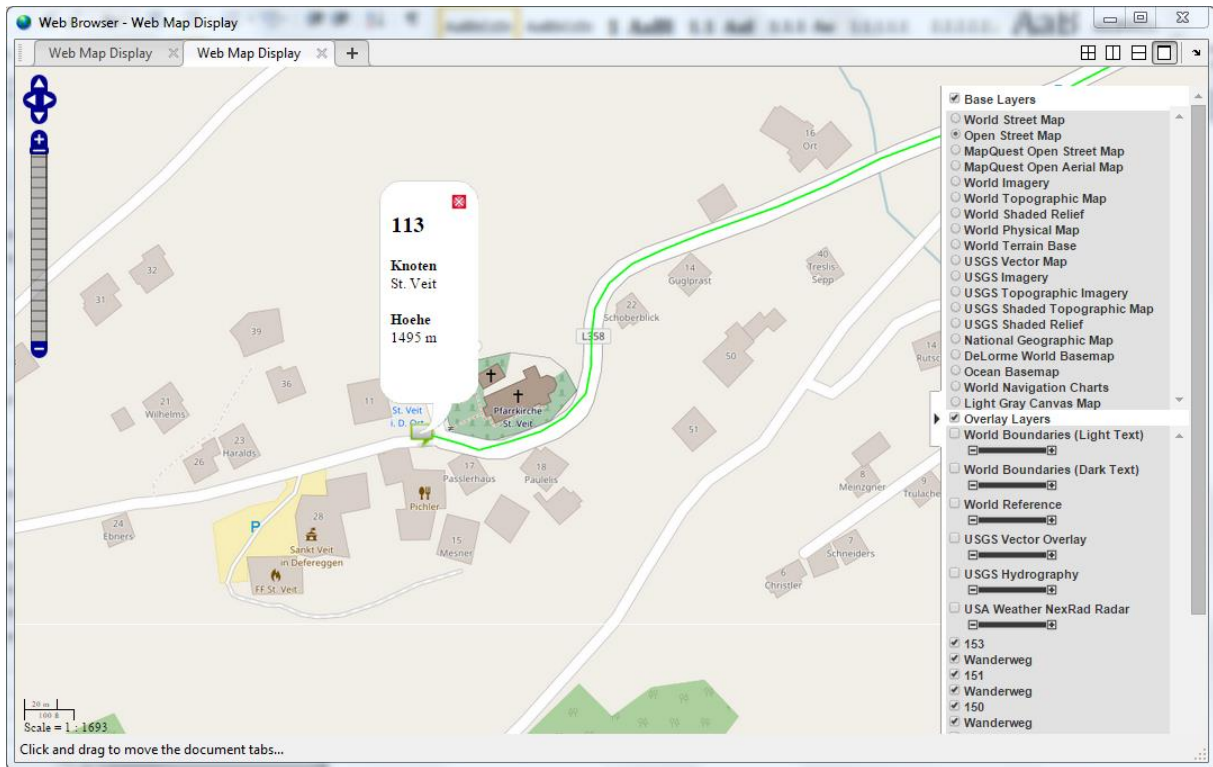


Abbildung 100: Benutzerausgabe - Ausschnitt aus dem WMS (Zielknoten)

10 Zusammenfassung und Ausblick

Ziel dieser Arbeit war es, eine multikriterielle Routenplanung für Wanderer/Wanderinnen in einem Wandergebiet zu entwickeln, die erhaltenen Ergebnisse darzustellen und zu analysieren. Dieser Vorgang wurde in den vorangegangenen Kapiteln beschrieben. Es wurde versucht, verschiedenste Forschungsfragen zu diskutieren und zu beantworten. Zusammenfassend kann man sagen, dass die Erweiterung des Dijkstra Algorithmus, um multikriterielle Problemstellungen zu lösen, interessante Möglichkeiten mit sich bringt. Die Möglichkeit mehrere Kriterien in einem Optimierungsverlauf zu berücksichtigen ist entscheidend für die Anwendungsmöglichkeiten und für das Finden optimaler Lösungen. Durch die Verwendung von verschiedenen Optimierungsmethoden, seien es „Nicht-Skalarisierungsmethoden“, um approximierte Lösungen zu erhalten oder alle Pareto-optimalen Wege zu finden, ergeben sich verschiedenste Möglichkeiten, um unterschiedliche Varianten zu testen und die Ergebnisse zu vergleichen. Im Vergleich zur „single-criterion“ Optimierung können nun mehrere Kriterien simultan berücksichtigt werden. Die Möglichkeit verschiedene Kriterien auszuwählen und somit deren Wichtigkeit festzulegen ermöglicht es benutzerspezifisch, optimale Wege zu finden. Allerdings muss erwähnt werden, dass nicht alle Kriterien als Hauptkriterien geeignet sind. Diese Untauglichkeit betrifft Kriterien wie die Steigung, die Untergrundbeschaffenheit und die POI's, wobei das Hauptproblem dieser Kriterien die Art und Weise, wie die Kriterien jeweils iterativ berechnet werden, ist. Allerdings eignen sich die Kriterien sehr wohl als nebensächliche Kriterien. Die Sinnhaftigkeit der Verwendung eines multikriteriellen Routenplanungsalgorithmus in einem Wandernetz muss ebenfalls erwähnt werden. Da sich das Wandergebiet über eine ziemlich große Fläche verteilt (340 km^2) gibt es teilweise Wege, bei denen die nächstgelegenen Knotenpunkte weit entfernt sind. Wären mehr Knotenpunkte innerhalb des Wegenetzes, so würde es auch mehr Möglichkeiten geben, wie die optimale Route verläuft. Allerdings gibt es genug Bereiche im Wandernetz, bei denen die Dichte an Knotenpunkten gut genug ist, um die Auswirkungen unterschiedlicher Einstellungen zu sehen. Je nachdem welche Kriterien verwendet werden, in welcher Reihenfolge (z.B. Hauptkriterien und Nebenkriterien) diese angeordnet werden und welche Optimierungsmethode gewählt wird, entstehen unterschiedliche Lösungen. Eine entscheidende Rolle spielt auch, ob die Kriterien ihren tatsächlichen, berechneten Wert besitzen oder ob die Werte gerundet werden. Es können starke und schwache Pareto-optimale Lösungen produziert werden. Die Vielfältigkeit der Nutzung dieses Routenplanungsalgorithmus ist dadurch auf jeden Fall gegeben. Allerdings ist, wie schon erwähnt, das Wandergebiet über eine sehr große Fläche verteilt und somit ergeben sich teilweise sehr große Distanzen zum Zielknoten, die ein Wanderer/eine Wanderin normalerweise nicht in einem Tag zurücklegen kann. Hier könnte man noch Hütten, bei denen es Übernachtungsmöglichkeiten gibt, hinzufügen, um so eine mehrtägige Tour zu planen. Für die praktische Anwendung ist diese Routenplanung daher nur bedingt geeignet, allerdings ist die Möglichkeit, sich verschiedene optimale Wege im Vorhinein anschauen zu können, eine interessante. Um die Routenplanung für

Wanderer/Wanderinnen noch interessanter zu gestalten, könnte man, wenn möglich, weitere Knoten und Kanten dem Netzwerk hinzufügen, um so eine größere Dichte an Abbiegepunkten zu erhalten. Dadurch würde die Sinnhaftigkeit der Anwendung eines multikriteriellen Routenplanungsalgorithmus deutlich gesteigert werden. Darüber hinaus könnte man das in *Matlab R2016a* erstellte Programm zu einer App weiterentwickeln, um so mehr Benutzern/Benutzerinnen den Zugang zu dieser Anwendungsmöglichkeit zu verschaffen. Schlussendlich bleibt als Fazit zu sagen, dass die Anwendung einer multikriteriellen Routenplanung für Wanderwege durchaus sinnvoll ist, allerdings muss auf die verwendeten Kriterien und auf das vorhandene Wandernetz geachtet werden, um aussagekräftige und interessante Ergebnisse zu erhalten.

Literaturverzeichnis

- Balakrishnan, R., & Ranganathan, K. (1999). *A textbook of graph theory*. New York: Springer.
- Bartelme, N. (2005). *Geoinformatik: Modelle, Strukturen, Funktionen*. Graz: Springer.
- Bartelme, N. (2014). *GIS Analysetechniken 1*. LV-Skriptum. Graz.
- Bartelme, N., & Rautz, K. (2014). *Visualisierung von Geoinformation*. LV-Skriptum. Graz.
- Collette, Y., & Siarry, P. (2003). *Multiobjective Optimization: Principles and Case Studies*. Springer.
- Ehrgott, M. (2005). *Multicriteria Optimization*. Auckland: Springer.
- Hallam, C., Harrison, K. J., & Ward, J. A. (2001). A multiobjective optimal path algorithm. *Digital Signal Processing*, S. 133-143.
- Hansen, P. (1979). Multiple Criteria Decision Making Theory and Application. *Lecture Notes in Economics and Mathematical Systems*, S. 109-127.
- Hoffmann-Wellenhof, B., Legat, K., & Wieser, M. (2003). *Navigation: principles of positioning and guidance*. Wien: Springer.
- Hofmann-Wellenhof, B. (2010). *Satellitengestützte Positionierung*. LV-Skriptum. Graz.
- Hofmann-Wellenhof, B., & Kienast, G. (2009). *Bezugssysteme*. LV-Skriptum. Graz.
- Jungnickel, D. (2002). *Graphs, Networks and Algorithms*. Springer.
- Korte, B., & Vygen, J. (2006). *Combinatorial Optimization: Theory and algorithms*. Berlin: Springer.
- Martins, E. (1984). On a multicriteria shortest path problem. *European Journal of Operational Research*, S. 236-245.
- Pressl, B. (2012). *Knowledge-based route planning*. Dissertation TU Graz.
- Schnee, M. (2009). *Fully Realistic Multi-Criteria Timetable Information Systems*. Dissertation TU Darmstadt.
- Steuer, R. E. (1986). *MULTIPLE CRITERIA OPTIMIZATION: THEORY, COMPUTATION, AND APPLICATION*. John Wiley&Sons.
- Tarapata, Z. (2007). *Selected Multicriteria Shortest Path Problems: An Analysis of Complexity, Models and Adaptation of Standard Algorithms*. International Journal of Applied Mathematics and Computer Science, S. 269-287.
- T'kindt, V., & Billaut, J.-C. (2002). *Multicriteria Scheduling*. Springer.
- Wieser, M. (2013/14). *Skriptum Navigation Systems, Routing and guidance*.