



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Masterthesis

Vor- und Zuname:

██████████
███
██████████████████

Stefan Fortmann

██████████
██████
██████

Titel:

Entwicklung und Evaluation von Lösungsansätzen für ein Rundreiseproblem mit Zeitlimits

Abgabedatum:

26.05.2018

Betreuender Professor:

Herr Prof. Dr. Bradtke

Zweiter Prüfender:

Herr Prof. Dr. Röhrs

Fakultät Wirtschaft und Soziales

Department Wirtschaft

Studiengang:

International Logistics and Management

I. Inhaltsverzeichnis

II.	Abbildungsverzeichnis.....	III
III.	Abkürzungs-/ Variablenverzeichnis.....	IV
IV.	Zusammenfassung.....	VI
1.	Einleitung.....	1
2.	Grundlagen.....	2
2.1	Das Rundreiseproblem.....	2
2.2	Lösungsverfahren.....	5
2.2.1	Vollständige Enumeration.....	6
2.2.2	Heuristische Verfahren für symmetrische TSP.....	6
2.2.3	Konstruktionsverfahren.....	6
2.2.4	Verfahren des besten Nachfolgers.....	7
2.2.5	Verfahren der Sukzessiven Einbeziehung von Stationen.....	9
2.2.6	Verbesserungsverfahren.....	10
3.	Vorgehensweise.....	11
3.1	Abgrenzung/ Definition des Problems.....	11
3.2	Definition des Problems.....	11
3.3	Ansatz des Lösungsverfahrens.....	13
3.4	Evaluation der Methoden.....	14
4.	Entwicklung der Lösungsmethode.....	15
4.1	Schrittweise Erklärung des Lösungsverfahrens.....	15
	Kostenkriterium.....	21
	Pufferkriterium.....	21
	Gewichtetes Kriterium.....	22
4.2	Bewertung des Ergebnisses.....	25
4.3	Einfügereihenfolgen.....	25
5.	Vorgehensweise der Evaluation.....	29
5.1	Erstellung der Datengrundlage.....	29
5.1.1	Methodik der Problemgenerierung.....	29
5.1.2	Limits:.....	30
5.2	Untersuchte Einflussfaktoren.....	38
5.2.1	Zusammenfassung der zu untersuchenden Faktoren.....	42

5.3	Fokus der Analyse.....	43
5.4	Bewertungsfaktoren der Analyse.....	43
5.4.1	Erfolgsquote:	44
5.4.2	Gesamtkosten:.....	44
6.	Erster Teil der Analyse.....	45
6.1	Messreihen im ersten Teil.....	45
6.2	Auswertung erster Teil.....	46
6.2.1	Auswertung der Erfolgsquoten	46
6.2.2	Auswertung der Gesamtkosten.....	48
6.2.3	Betrachtung beider Bewertungsfaktoren in Relation:	49
7.	Zweiter Teil der Analyse.....	51
7.1	Messreihen im zweiten Analyseteils.....	51
7.2	Auswertung zweiter Teil.....	52
7.2.1	Anmerkung zum Einfluss der Standortverteilung	53
7.2.2	Gegenüberstellung der Lösungswege allgemein.....	53
7.2.3	Analyse des Einflusses der Limits	55
7.2.4	Untersuchung der Ergebnisse bei zentraler Ballung der Standorte.....	60
7.2.5	Lage des Startpunktes	63
7.2.6	Exkurs zum Einfluss der Puffer im Entscheidungskriterium	65
7.3	Fazit der Auswertung	65
7.	Implementierung des Verfahrens	67
7.1	Erklärung der verwendeten Funktionen:	68
7.2	Variablen in Visual Basic:.....	69
7.3	Generierung von Problemen:	71
7.3.1	Generierung der Koordinaten	71
7.3.2	Generierung der Kostenmatrix.....	74
7.3.3	Bestimmung der Limits.....	75
7.4	Durchführung der Methode	76
7.5	Bestimmung der Einfügereihenfolgen	82
8.	Schlussbetrachtung	85
V.	Quellenverzeichnis	VII
VI.	Eidesstattliche Erklärung	VIII
VII.	Einverständniserklärung.....	VIII

II. Abbildungsverzeichnis

Abbildung 1: Aufteilung eines Problems in kleinere Rundreiseprobleme durch Clustering	3
Abbildung 2: Beispielhafte Struktur einer Kostenmatrix mit $n=4$ Standorten	3
Abbildung 3: Kategorisierung von Lösungsverfahren	5
Abbildung 4; Abbildung 5: Beispielhafte Problemlösungen mit dem Verfahren des besten Nachfolgers.....	8
Abbildung 6: Extremes Beispiel einer Lösung mit dem Verfahren des besten Nachfolgers.....	8
Abbildung 7: Beispielhafte Kostenmatrix mit $n=4$ Orten und deren Zeitlimits	16
Abbildung 8: Beispielhafte Ausgangssituation für die Wahl einer Einfügeposition	23
Abbildung 9: Auswirkung verschiedener Gewichtungen auf die Wahl der Einfügeposition ...	23
Abbildung 10: Beispielhafte Verteilung der Limits bei Grenzen von 5 und 10 Einheiten.....	32
Abbildung 11: Ergebnisse der Testreihe zur Ermittlung der \emptyset Länge von Einzelstrecken in der finalen Tour	35
Abbildung 12: Gesetzte Zielwerte für die \emptyset Länge von Einzelstrecken in der finalen Tour.....	35
Abbildung 13: Mittlere Höhe der Obergrenzen in Abhängigkeit der Standortanzahl	38
Abbildung 14: Beispielhafte zufällige Verteilung der Orte.....	40
Abbildung 15: Beispielhafte Verteilung mit zentraler Ballung.....	40
Abbildung 16: Beispielhafte Zufällige Verteilung der Orte im gestreckten Raum (50x200 Einheiten)	40
Abbildung 17: Beispielhafte Verteilung der Orte bei Generierung in „Städteclustern“	41
Abbildung 18: Beispielhafte absolute Gesamtkosten, welche durch verschiedene Lösungswege ermittelt wurden	44
Abbildung 19: Darstellung der Gesamtkosten in Relation zum Optimum des jeweiligen Datensatzes	45
Abbildung 20: Erfolgsquoten der Lösungswege im ersten Analyseteil.....	46
Abbildung 21: \emptyset Gesamtkosten der Lösungswege im ersten Analyseteil	49
Abbildung 22: Grafische Darstellung der Erfolgsquoten und \emptyset Gesamtkosten als Mittelwert aller Messreihen.....	50
Abbildung 23: Anzahl zulässig gelöster Probleme aller im zweiten Analyseteil vorgenommenen Messreihen	52
Abbildung 24: Erfolgsquoten und \emptyset Gesamtkosten bei zufälliger Generierung der Standorte	54
Abbildung 25: Erfolgsquoten und \emptyset Gesamtkosten ausgewählter Lösungswege bei 20 zu besuchenden Orten.....	56
Abbildung 26: Erfolgsquoten und \emptyset Gesamtkosten ausgewählter Lösungswege bei 50 zu besuchenden Orten.....	57
Abbildung 27: Erfolgsquoten und \emptyset Gesamtkosten ausgewählter Lösungswege bei 100 zu besuchenden Orten.....	58
Abbildung 28: Erfolgsquoten bei zentral geballter Generierung von 20/ 50 Orten	61
Abbildung 29: Erfolgsquoten bei zentral geballter Generierung von 100 Orten.....	62
Abbildung 30: Erfolgsquoten in Abhängigkeit der Lage des Startpunkts	64

III. Abkürzungs-/ Variablenverzeichnis

In diesem Verzeichnis werden nur die Variablen des Hauptteils aufgeführt. Die für die Implementierung in Visual Basic relevanten Variablen erhalten ein eigenes Verzeichnis zu Beginn von Teil 8.

- $D_{dif}(z)$:..... Differenz zwischen den min. Restpuffer an Einfügeposition z und dem Optimum aller Positionen
- $D_{min}(z)$:..... Minimaler Restpuffer an einem der hinter der Einfügeposition gelegenen Punkte nach dem Einfügevorgang
- D_{opt} : Maximum der minimalen Restpuffer aller Einfügepositionen z
- $D_p(z)$:Verbleibender Puffer am einzufügenden Punkt nach Einfügung an Position z
- $D_{zu}(z)$: Geringster Puffer, welcher aus dem Einfügevorgang an Position z resultieren würde
- E_c : Einfügekosten eines Punktes c
- $E_{dif}(z)$: Differenz zwischen den Einfügekosten der Einfügeposition z und den optimalen Einfügekosten
- E_{opt} : Geringste Einfügekosten (im Vergleich aller Einfügepositionen)
- $E_{p_s}(z)$: Einfügekosten des in Iterationsschritt s einzufügenden Punktes an Einfügeposition z
- F : Faktor zur Errechnung der unflexiblen Limit-Obergrenze
- $F(20)$: Angepasster Faktor, welcher bei 20 Orten zum Zielwert s_z führt
- $F(100)$: Angepasster Faktor, welcher bei 100 Orten zum Zielwert s_z führt
- F_{flex} : Flexibler Faktor zur Errechnung der Limit-Grenzen, welcher von der Standortanzahl m abhängig ist
- $f = \{i_1, i_2, \dots, i_n\}$:..... Reihenfolge, in der die Punkte der finalen Tour zugeordnet sind
- G_m : Matrixgröße; ein Faktor zur Errechnung der Limits
- $G(z)$: Entscheidungskriterium beim gewichteten Kriterium, welches für jede Einfügeposition z bestimmt wird
- G_{opt} : Optimaler (kleinster) Wert von $G(z)$
- g_{kost} : Gewichtung der Kosten zur Errechnung von $G(z)$
- g_{puff} : Gewichtung der Puffer zur Errechnung von $G(z)$
- $i_{s;j}$: Beschreibt den Punkt, welcher nach Iterationsschritt s an j -ter Stelle der Tour steht
- $k(a, b)$: Kosten für die Fahrt von Punkt a zu Punkt b

- k_{\emptyset} : Mittelwert aller Werte der Kostenmatrix ausgenommen der Strecken zum Startpunkt
- $k_{\emptyset}(a)$: Mittlere Entfernung des Punktes a zu allen anderen Punkten
- K_{ges} : Gesamtkosten
- $K_s(a)$: Kumulierte Kosten zur bei Erreichung von Punkt a nach Iterationsschritt s
- $L(a)$: Limit von Punkt a
- L_{max} : Limit-Obergrenze
- L_{min} : Limit-Untergrenze
- $P = (p_1, p_2, \dots, p_m)$: Einfügereihenfolge
- p_s : Der Punkt, welcher an s-ter Stelle der Einfügereihenfolge steht und somit in Iterationsschritt s eingefügt wird.
- n : Anzahl Standorte einschließlich Start
- m : Anzahl zu besuchender Orte (exklusive des Startpunktes)
- $R_s(a)$: Verbleibender Restpuffer von Punkt a nach Iterationsschritt s
- s : Zähler für den Iterationsschritt
- S_{kalk} : Mittlere Länge eines Streckenabschnitts der finalen Tour, welche die Obergrenze maximal erlaubt
- S_z : Gesetzter Zielwert für die mittlere Länge eines Streckenabschnitts der finalen Tour
- $T_s = (i_{s;0}, i_{s;1}, \dots, i_{s;s}, i_{s;s+1})$: Neue (Teil-)Tour, welche in Iterationsschritt s gebildet wird
- T_m : Finale Tour, welche in Iterationsschritt s=m gebildet wird
- U : Faktor zur Errechnung der unflexiblen Untergrenze
- V_{ges} : Gesamte Verspätung/ Überschreitung der Limits in der finalen Tour
- x_a : x-Koordinate von Punkt a
- y_a : y-Koordinate von Punkt a
- z : Zähler für die Einfügeposition
- z_{opt} : Durch das Entscheidungskriterium gewählte Einfügeposition

IV. Zusammenfassung

In dieser Masterarbeit werden Lösungsansätze für ein Rundreiseproblem mit Zeitlimits entwickelt und erläutert. Diese basieren auf dem Verfahren der sukzessiven Einbeziehung von Stationen, verwenden jedoch unter anderem unterschiedliche Entscheidungskriterien zur Auswahl der Einfügeposition. Der Hauptteil dieser Arbeit beschäftigt sich mit der Evaluation der entwickelten Methode und dessen Varianten. Dazu wird zunächst die Vorgehensweise zur Evaluation vorgestellt und anschließend eine Analyse der erhobenen Daten durchgeführt.

Neben den Entscheidungskriterien zur Wahl der Einfügeposition setzt sich die Analyse primär mit dem Einfluss der Einfügereihenfolgen auseinander. Untersucht werden dabei in mehreren Messreihen sowohl die Erfolgsaussichten als auch die resultierenden Gesamtkosten unter verschiedenen Rahmenbedingungen.

Unter den gewählten Einflussfaktoren ging als Ergebnis der Analyse primär ein Lösungsweg hervor, welcher sich klar von den anderen abheben konnten. Hinsichtlich der Rahmenbedingungen der zu lösenden Probleme erwiesen sich besonders die Anzahl der Standorte sowie die Höhe der Limits als ausschlaggebend für die Ergebnisse. Aus den gezogenen Schlüssen wird zum Ende der Analyse versucht, Handlungsempfehlungen für die Lösung in Abhängigkeit der Rahmenbedingungen zu formulieren.

1. Einleitung

Innerhalb der letzten Jahrzehnte haben sich auf dem Gebiet von Logistik und Produktion viele Änderungen vollzogen. Nicht nur globalere Märkte und moderne Informationsflüsse, sondern auch moderne Organisations- und Steuerkonzepte wie Just-in-time oder Lean-Production erlauben effizienteres Wirtschaften und finden daher zunehmend Anwendung. In der Folge haben mehr Faktoren und Interessensgruppen einen Einfluss auf die logistische Umgebung und Zielsetzung. Vor diesem Hintergrund werden logistische Problemstellungen zunehmend komplexer und spezifischer.

Fast schon im Gegensatz dazu stehen klassische logistische Problemstellungen wie das Transportproblem oder das Rundreiseproblem, für die in der Literatur eine Vielzahl an Lösungsverfahren zu finden sind, welche jedoch nur selten die komplexen Fragestellungen der Realität abbilden können. Gleichmaßen stoßen auch diesen Problemen gewidmete Lösungsverfahren schnell an ihre Grenzen, sofern die eng gefassten Rahmenbedingungen nicht eingehalten werden. Hintergrund dieser Arbeit ist der Versuch, solche Rahmenbedingungen zu überschreiten und somit ein klassisches, logistisches Problem durch die Erweiterung um eine neue Dimension für die Realität relevanter zu machen.

Nun muss eine logistische Problemstellung natürlich nicht auf einer bereits bekannten basieren, jedoch können auf diese Weise zumindest bedingt die Vorteile der klassischen Problemstellung übernommen werden. So sind diese einerseits meist durch ihre klare Abgrenzung sehr einfach zu verstehen und andererseits können durch simple, heuristische Lösungsverfahren gute Ergebnisse erzielt werden. Diese Arbeit beschäftigt sich konkret mit einem modifizierten Rundreise- oder Traveling Salesman Problem, welches um die Komponente von Zeitlimits erweitert wird.

Als Vorbild für dieses konkretisierte Problem dienen unter anderem moderne Supply Chains und Logistikkonzepte wie Just-in-time oder Just-in-Sequence. Der allgemeine Trend dabei geht in Richtung eines Abbaus von Lagerbeständen und hin zu kurzfristigen, zeitlich genau abgestimmten Lieferungen. Jedoch auch abseits dieser spezifischen Konzepte werden Zuliefernetzwerke vor dem Hintergrund der Effizienzsteigerung komplexer und schnellläufiger.

Die gewünschte Effizienzsteigerung kann sowohl durch Reduktion des Inputs als auch durch Maximierung des Outputs erreicht werden. Während diese Begriffe klassisch eher mit der Produktion in Zusammenhang gebracht werden, zählen in der Logistik zum Beispiel finanzielle Mittel und Fahrzeuge zum Input. Im konkreten Kontext des Rundreiseproblems bedeutet die Wahl einer kostengünstigeren Tour eine Reduktion der Finanzmittel, welche als Input aufgebracht werden müssen. Die Maximierung des Outputs äußert sich im Kontext der Logistik zum Beispiel darin, dass vorhandene Infrastruktur oder Transportmittel besser ausgelastet werden oder das vorhandene Raumangebot effizienter genutzt wird. Konkret kann das bedeuten, dass für Anlieferung und Abholung von Waren konkrete Zeitfenster vorgegeben werden, damit weniger Laderampen in besserer Auslastung benötigt werden.

Hinsichtlich des Platzangebots werden nicht wertschöpfende Lagerflächen reduziert und die Bestände durch regelmäßige Anlieferungen minimiert. In der Konsequenz kommt es zu extrem kostspieligen Stillständen, sofern Ware nicht bis zu einem gewissen Zeitpunkt geliefert wurde. All dies setzt ein hohes Maß an zeitlicher Präzision für die Transportlogistik voraus, insbesondere weil sich durch den Abbau von Sicherheiten die Kosten von Störungen erhöht haben.

Als Zielsetzung dieser Arbeit ergibt sich das Finden einer Lösungsmethode für ein Rundreiseproblem, welches diese zeitlichen Begrenzungen berücksichtigen kann. Dafür können die bestehenden Verfahren als Grundlage dienen, müssen jedoch an die neuen Rahmenbedingungen angepasst werden. Während es die neue Methode ermöglichen muss, das komplexere Problem zu lösen, wäre es gleichsam zu erhoffen, dass die Übersichtlichkeit und vergleichsweise einfache Lösungssystematik des Vorbilds erhalten werden kann.

2. Grundlagen

In diesem Teil werden das klassische Rundreiseproblem sowie einige der bekanntesten Lösungsverfahren vorgestellt. Bei den Lösungsmodellen wird primär auf die allgemeine Funktionsweise eingegangen und weniger intensiv auf die konkreten Rechenschritte. Diese sind zum Großteil auch im modifizierten Verfahren enthalten und werden ausführlich bei dessen Erklärung in Teil 4.1 genauer erläutert.

2.1 Das Rundreiseproblem

Das Rundreiseproblem, auch bekannt unter dem Problem des Handlungsreisenden oder der englischen Bezeichnung *traveling salesman problem* (TSP) ist ein kombinatorisches Optimierungsproblem, welches das Ziel verfolgt, eine Vielzahl von Knoten mit einer möglichst günstigen Tour zu verbinden. Die Rundreise muss anschließend wieder am Startpunkt enden.¹ Das TSP ist ein Sonderfall eines allgemeinen Tourenplanungsproblems und grenzt sich von diesem dadurch ab, dass alle Knoten auf einer einzelnen Tour untergebracht werden.² Kapazitätsrestriktionen, wie sie bei Tourenplanungsproblemen relevant sind, werden im klassischen Rundreiseproblem nicht beachtet.³

Das Problem kann für eine Vielzahl von Anwendungen verwendet werden, so können die Knoten verschiedene Orte darstellen, jedoch auch beispielsweise Produkte, welche in einer gewissen Reihenfolge mit variierenden Rüstkosten gefertigt werden sollen.⁴ Auch kann ein Rundreiseproblem ein Teilproblem eines größeren Tourenplanungsproblems sein, zum Beispiel nachdem ein Clustering der Knoten vorgenommen wurde.⁵ (siehe dazu Abbildung 1)

¹ Vgl. Neumann, Morlock(2002) S. 438

² Vgl. Thonemann (2005) S. 407 ff.

³ Vgl. Thonemann (2005) S. 417 ff.

⁴ Vgl. Zimmermann, Stache(2001) S. 148

⁵ Vgl. Wenger(2009) S. 40f.

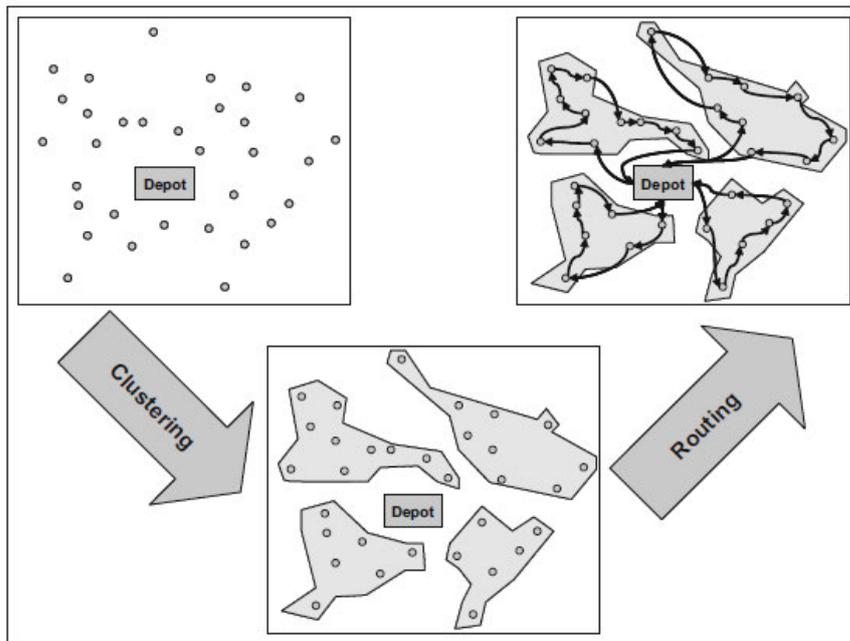


Abbildung 1: Aufteilung eines Problems in kleinere Rundreiseprobleme durch Clustering⁶

Das in Abbildung 1 links oben zusehende Problem kann also auch als Rundreiseproblem bezeichnet werden, sofern die Prämisse gilt, dass nur eine Tour gebildet werden kann. Zur einfachen Verbildlichung eines Rundreiseproblems eignet sich besonders die namensgebende Vorstellung eines Handlungsreisenden, welcher eine bestimmte Anzahl von Kunden in verschiedenen Städten besuchen muss und anschließend zu seinem Ausgangsort zurückkehren will. Für diese Kundenbesuche gilt es, die optimale Reihenfolge zu bestimmen, um gefahrene Strecke, die Fahrtkosten oder den Zeitaufwand zu minimieren.⁷

Zur Bestimmung einer Lösung wird zunächst eine Kostenmatrix benötigt, welche alle Kosten $k(a, b)$ ⁸ für die Fahrten von einem Punkt a zu einem Punkt b enthält.⁹ Bei n Standorten einschließlich des Start- und Endpunktes enthält die Matrix daher n^2 Werte. Analog kann bei anderen Zielkriterien auch z.B. eine Strecken- oder Zeitmatrix verwendet werden.¹⁰

		Nach			
		Punkt 0	Punkt 1	Punkt 2	Punkt 3
Von	Punkt 0	k (0,0)	k (0,1)	k (0,2)	k (0,3)
	Punkt 1	k (1,0)	k (1,1)	k (1,2)	k (1,3)
	Punkt 2	k (2,0)	k (2,1)	k (2,2)	k (2,3)
	Punkt 3	k (3,0)	k (3,1)	k (3,2)	k (3,3)

Abbildung 2: Beispielhafte Struktur einer Kostenmatrix mit $n=4$ Standorten

⁶ Entspricht aus: Wenger (2009) S. 41

⁷ Vgl. Zimmermann, Stache (2001)S. 147

⁸ Die Schreibweise wurde an die im Hauptteil verwendete angepasst

⁹ Vgl. Zimmermann, Stache (2001) S. 148

¹⁰ Vgl. Hinrichsen (1975) S. 9

Wenn die Kostenmatrix zu Beginn der Berechnung unbekannt oder unvollständig ist, so müssen die Werte zunächst erhoben werden.¹¹ Es kann in der Praxis jedoch auch sinnvoll sein, keine Vervollständigung vorzunehmen und somit die Werte in der Matrix auf wenige, sinnvolle Verbindungen zu beschränken.¹² In diesem Fall können diese Felder mit hohen Werten blockiert werden.

Allgemein unterscheidet man zwischen dem symmetrischen und asymmetrischen Rundreiseproblem. Im Rahmen dieser Arbeit wird mit symmetrischen Problemen gearbeitet, welchen graphentheoretisch ein ungerichteter Graph zugrunde liegt.¹³ Das bedeutet, dass die Kosten einer Strecke zwischen zwei Punkten in beiden Richtungen gleich sind.¹⁴ Also gilt für alle Strecken zwischen zwei Punkten a und b:

$$k(a, b) = k(b, a)$$

Eine zulässige Lösung besteht in einer geschlossenen Tour (Startpunkt=Endpunkt), welcher jeder zu besuchende Ort genau einmal zugeordnet ist. Dieses Ergebnis lässt sich beispielsweise beschreiben in Form der Reihenfolge f, in der die Punkte der Tour zugeordnet sind:¹⁵

$$f = \{i_1, i_2, \dots, i_n\}$$

Für diese Reihenfolge lassen sich die Gesamtkosten der Tour $K(f)$ anhand der Kostenmatrix wie folgt bestimmen:¹⁶

$$K(f) = k(i_1, i_2) + k(i_2, i_2) + \dots + k(i_{n-1}, i_n) + k(i_n, i_1)$$

Zielsetzung der Problemlösung ist es, die Reihenfolge f_{opt} zu finden, deren Kosten $K(f_{\text{opt}})$ minimal sind.¹⁷ Die Komplexität des Rundreiseproblems besteht darin, dass es, zumindest bei einer Vielzahl von Standorten, eine sehr große Anzahl an zulässigen Lösungen gibt. Diese Anzahl entspricht bei n Standorten $(n-1)!$ ¹⁸, wie kurz anhand eines Beispielproblems mit n=4 Standorten gezeigt werden soll:

Beispiel: Ein Handlungsreisender möchte 3 Kunden in den Städten A, B und C besuchen. Er selbst lebt in Stadt D, von wo aus er seine Reise startet und wieder beendet. Für seinen ersten Kundenbesuch hat er also 3 Optionen: Stadt A, B oder C. Nachdem er einen der drei Kunden besucht hat, muss er sich für die nächste Station zwischen den beiden verbliebenden entscheiden. Da er jeden Kunden nur einmal besuchen will, bleibt zuletzt nur

¹¹ Vgl. Domschke, Drexl, Klein, Scholl (2015) S. 152

¹² Ebd.

¹³ Vgl. Domschke, Drexl, Klein, Scholl (2015) S. 151

¹⁴ Vgl. Neumann, Morlock (2002) S. 439

¹⁵ Vgl. Hinrichsen (1975) S. 9

¹⁶ Vgl. Hinrichsen (1975) S. 9 ; Schreibweise wurde an die im Hauptteil verwendete angepasst

¹⁷ In Anlehnung an Hinrichsen (1975) S.9

¹⁸ Vgl. Zimmermann, Stache (2001) S.148

noch der eine unbesuchte Kunde übrig. Danach muss er in jedem Fall wieder nach Stadt D zurück. Es gibt für ihn hier also $3 \times 2 \times 1 = 6$ verschiedene Möglichkeiten:

- D-A-B-C-D**
- D-A-C-B-D**
- D-B-A-C-D**
- D-B-C-A-D**
- D-C-A-B-D**
- D-C-B-A-D**

2.2 Lösungsverfahren

Wie Grafik 3 zeigt, können Lösungsverfahren nach verschiedenen Faktoren kategorisiert werden. Im Folgenden werden das exakte Verfahren der vollständigen Enumeration sowie zwei heuristische Verfahren vorgestellt.

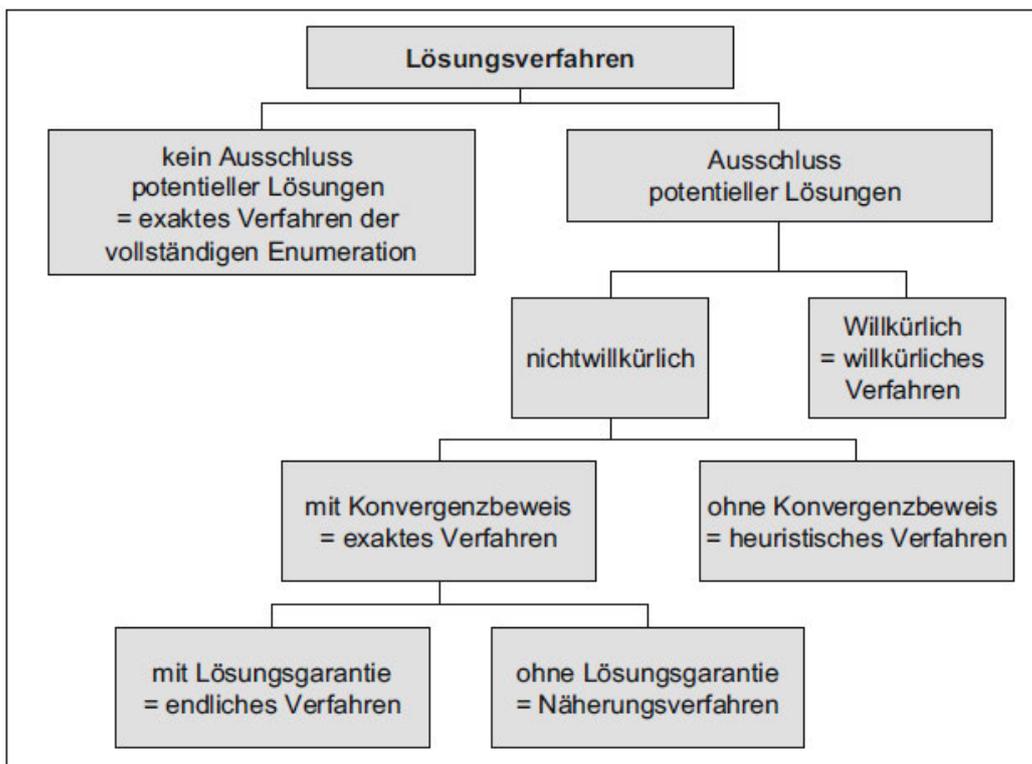


Abbildung 3: Kategorisierung von Lösungsverfahren¹⁹

¹⁹ Entspringt aus: Wenger (2009) S. 68

2.2.1 Vollständige Enumeration

Bei der vollständigen Enumeration werden alle der $(n-1)!$ Lösungen ermittelt und aus diesen das beste Ergebnis ausgewählt. Somit kann garantiert die ideale Lösung gefunden werden, jedoch ist der damit verbundene Rechenaufwand bei großen n extrem hoch.²⁰

Das vorangegangene Beispiel mit $n=4$ Standorten ist mit 6 möglichen Lösungen natürlich sehr übersichtlich und leicht zu berechnen. Erhöht sich jedoch die Standortanzahl, wird die Anzahl Lösungen allerdings zunehmend unübersichtlich und ist nur noch unter großem Aufwand komplett zu berechnen. So hat ein Problem mit $n=15$ Standorten bereits $14! = 87178291200$ mögliche Lösungen.²¹

2.2.2 Heuristische Verfahren für symmetrische TSP

Da der Rechenaufwand bei der vollständigen Enumeration zur Findung der idealen Lösung unverhältnismäßig ist, muss in der Praxis bei großen n auf heuristische Verfahren, das heißt Näherungsverfahren zurückgegriffen werden.²² Diese Verfahren sind systematische Suchverfahren, welche versuchen, mit relativ geringem Rechenaufwand eine sich an das Optimum annähernde Lösung zu finden.²³ In der Praxis sind diese im Vergleich sehr schnell gewonnenen Näherungslösungen meistens praktikabler als die erheblich aufwendigere exakte Bestimmung.²⁴

Man unterscheidet bei heuristischen Verfahren zwischen **Tourenkonstruktionsverfahren**, welche eine möglichst günstige Tour konstruieren, und **Tourenverbesserungsverfahren**, welche versuchen eine bestehende Tour zu verbessern.²⁵

2.2.3 Konstruktionsverfahren

Heuristische Konstruktionsverfahren haben im Allgemeinen das Problem, dass es schwierig ist, die Güte der resultierenden Lösung einzuschätzen. Das heißt, es ist nicht ersichtlich, wie weit die gewonnene Lösung vom Optimum abweicht.²⁶ In diesem Teil werden zwei der bekanntesten Verfahren zur Lösung von Rundreiseproblemen behandelt: Das Verfahren des besten Nachfolgers und das Verfahren der sukzessiven Einbeziehung (von Stationen) oder Einfügeverfahren.²⁷

²⁰ Zimmermann, Stache (2001) S. 148

²¹ Ebd.

²² Zimmermann, Stache (2001) S.150

²³ Ebd.

²⁴ Vahrenkamp (2003) S. 204

²⁵ Ebd.

²⁶ Zimmermann, Stache S.150

²⁷ Vahrenkamp (2003) S. 204 ff ; Zimmermann, Stache (2001) S. 151

2.2.4 Verfahren des besten Nachfolgers

Das Verfahren des besten Nachfolgers verwendet eine schrittweise Vorgehensweise, bei der an die konstruierte Tour in jedem Schritt ein weiterer Punkt angefügt wird. Dabei wird jener Punkt gewählt, welcher den geringsten Kostenzuwachs mit sich bringt. Die konkrete Vorgehensweise lässt sich in drei Teilen beschreiben.²⁸

Initialisierung: Es wird ein beliebiger Knoten als Startpunkt gewählt und einer neuen, leeren Tour hinzugefügt.²⁹ In einem praxisbezogenen Beispiel mit einem fixierten Startpunkt muss dieser nicht zwangsläufig dem tatsächlichen Startpunkt entsprechen. Da das Ergebnis der Methode eine geschlossene Rundreise ist, kann der Startpunkt anschließend beliebig verändert werden.

Iteration: In jedem Iterationsschritt wird geprüft, welcher noch nicht in die neue Tour integrierte Knoten a dem bisherigen Endpunkt der Tour b am nächsten ist, das heißt, die niedrigsten Kosten $k(a, b)$ aufweist. Dieser Knoten wird dann der Tour als neuer Endpunkt hinzugefügt.³⁰

Abbruch: Die Iteration bricht ab, wenn alle Knoten in die neue Tour integriert sind. Das ist der Fall nach $(n-1)$ Iterationsschritten. Da die Tour einen geschlossenen Kreis bilden soll, wird zuletzt der gewählte Startpunkt als neuer Endpunkt der Tour hinzugefügt, wodurch das Verfahren abgeschlossen ist.³¹

Kritische Betrachtung: Das Verfahren des besten Nachfolgers liefert in den meisten Fällen Ergebnisse, welche vergleichsweise weit vom Optimum entfernt liegen. Das liegt daran, dass zum Ende des Verfahrens die Anzahl möglicher Verbindungen stark abnimmt, wodurch zum Teil sehr schlechte Verbindungen in Kauf genommen werden müssen.³² Außerdem kann schon in den ersten Schritten eine Reihenfolge entstehen, welche sich negativ auf die Möglichkeiten in späteren Schritten auswirkt. Da die Struktur der generierten Teiltour nicht mehr geändert werden kann, ist dies problematisch. Die nachfolgenden Beispiele (Grafiken 4 – 6) sollen dieses Problem veranschaulichen.

²⁸ Vahrenkamp (2003) S. 204 f.

²⁹ Vgl. Domschke, Drexl, Klein, Scholl (2015); S. 152; Vahrenkamp (2003) S. 204

³⁰ Ebd.

³¹ Zimmermann, Stache (2001) S. 151

³² Zimmermann, Stache (2001) S. 151

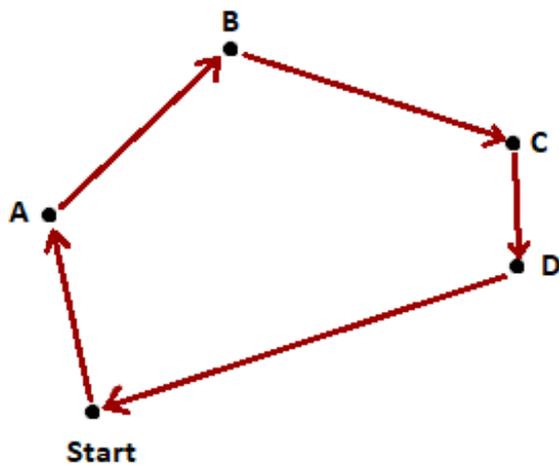


Abbildung 4; Beispielhafte Problemlösungen mit dem Verfahren des besten Nachfolgers

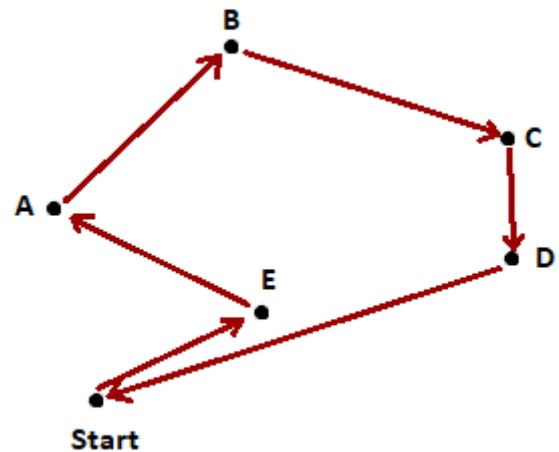


Abbildung 5:

Grafik 4 zeigt ein Problem, welches mit dem Verfahren des besten Nachfolgers sehr gut gelöst werden kann, da die Entfernungen der Punkte zueinander gut zur Systematik des Verfahrens passen. Grafik 5 zeigt jedoch, dass durch die Einbeziehung des Punktes E ein eindeutig schlechteres Ergebnis erzielt wird. Obwohl E mit nur sehr geringem Umweg zwischen D und dem Start eingefügt werden könnte, wird hier ein deutlich größerer Umweg gewählt. Diese Problematik entsteht, weil der Abstand zwischen A und E geringer ist als der zwischen E und D. Würde in diesem Fall Punkt C als Start gewählt werden, so ergäbe sich eine bessere Tour: C-D-E-Start-A-B-C.

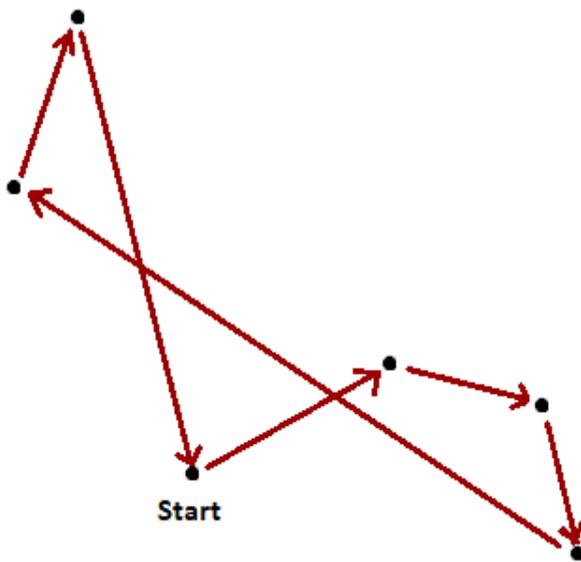


Abbildung 6: Extremes Beispiel einer Lösung mit dem Verfahren des besten Nachfolgers

In Abbildung 6 ist ein noch extremeres Beispiel zu sehen, bei dem in den ersten Schritten zwar kurze Wege gewählt werden, jedoch anschließend durch Mangel an Alternativen zwei sehr schlechte Verbindungen hingenommen werden müssen.

Insgesamt kann die Wahl des Startpunktes einen erheblichen Einfluss auf den Ausgang des Verfahrens haben. Es ist jedoch besonders bei größeren Problemen extrem schwierig einzuschätzen, welche Wahl zum besten Ergebnis führt.

2.2.5 Verfahren der Sukzessiven Einbeziehung von Stationen

Bei der sukzessiven Einbeziehung von Stationen wird eine bereits geschlossene Tour als Ausgangspunkt gewählt, welche in jedem Schritt um einen Punkt erweitert wird. Für den Zweck dieser Erklärung wird die in Iterationsschritt s resultierende Tour wie folgt bezeichnet:

$$T_s = (i_0, i_1, \dots, i_0)^{33}$$

Wie auch das Verfahren des besten Nachfolgers lässt sich auch das Einfügeverfahren in drei Teile gliedern:³⁴

Initialisierung: In diesem Verfahren werden anfangs zwei beliebige Knoten ausgewählt. Einer davon wird als Start- und Endknoten i_0 festgelegt. Der zweite Knoten i_1 bildet mit diesen die Anfangstour $T_0 = (i_0, i_1, i_0)^{35}$. Wie auch beim Nachfolgerverfahren ist das Ergebnis eine Tour, in der der Startpunkt frei verschoben werden kann. Daher sind die hier gewählten Anfangspunkte beliebig. Hinsichtlich derer konkreter Wahl gibt es Ansätze zur Optimierung des Verfahrens, auf die bei den Erläuterungen zur Einfügereihenfolge eingegangen wird.

Iteration: In jedem Iterationsschritt s wird ein weiterer Knoten c gewählt, welcher noch nicht zur Tour T_{s-1} gehört.³⁶ Für diesen wird geprüft, an welcher Position dessen Einfügen in die Tour die geringsten Einfügungskosten verursacht. Die Einfügungskosten von Knoten c zwischen zwei benachbarten Knoten der bestehenden Tour a und b bestimmen sich aus der Summe der Kosten der hinzukommenden Strecken abzüglich denen der Strecke, welche ersetzt wird. Sie können demnach wie folgt errechnet werden:³⁷

$$E_c = k(a, c) + k(c, b) - k(a, b)$$

Somit wird die Tour in jedem Iterationsschritt um einen Punkt erweitert, sowie auch beim Verfahren des besten Nachfolgers. Gegensätzlich zu diesem stehen allerdings mit jedem Schritt mehr Möglichkeiten der Einfügung zur Auswahl.

Abbruch: Das Verfahren bricht ab, sobald alle Knoten in die neue Tour integriert sind. Das ist hier schon nach $(n-2)$ Iterationsschritten der Fall, weil bei der Initialisierung bereits zwei Punkte in die Tour integriert werden.

³³ Die Schreibweise ist auch hier an die im Hauptteil verwendete angeglichen

³⁴ Vgl. Vahrenkamp (2003) S. 209 ; Domschke, Drexl, Klein, Scholl (2015) S. 153 ; Neumann Morlock (2002) S. 445 f.

³⁵ Vgl. Neumann Morlock 445, Schreibweise angepasst an die im Hauptteil verwendete

³⁶ Vgl. Domschke, Drexl, Klein, Scholl (2015) S. 153

³⁷ Vgl. Arnold, Isermann, Kuhn, Tempelmeier, Furmanns (2008) S. 148; Schreibweise wurde auch hier an die im Hauptteil verwendete angepasst.

Einfluss der Einfügereihenfolge: Während beim Verfahren des besten Nachfolgers die Wahl des Startpunktes einen Einfluss auf das Ergebnis hat³⁸, so haben hier sowohl die Wahl des Startpunktes, insbesondere aber auch die Reihenfolge, nach der die Punkte eingefügt werden, einen zum Teil erheblichen Einfluss auf den Ausgang des Verfahrens. Für die Bestimmung der Einfügereihenfolge existieren verschiedene Ansätze, welche neben den Auswirkungen auf die Gesamtkosten jedoch auch zum Teil einen großen Rechenaufwand zu deren Bestimmung mit sich bringen.³⁹ Vergleichsweise gute Ergebnisse werden erzielt, wenn die grobe Form der Tour in den ersten Schritten der Iteration bereits etabliert wird. Das heißt, dass aus den am weitesten voneinander entfernten Punkten zunächst ein Grundgerüst konstruiert wird, welches durch weitere Punkte mit kürzeren Entfernungen weiter detailliert wird.⁴⁰ Daraus folgt die Einfügesystematik, dass in jedem Schritt jener Punkt eingefügt wird, welcher von der bestehenden Tour am weitesten entfernt ist.⁴¹ Hinsichtlich der Wahl der beiden Knoten für die Initialisierung des Verfahrens wird daher in verschiedenen Quellen empfohlen, zwei möglichst weit voneinander entfernte Punkte zu wählen.⁴² Eine zufällige Wahl der Einfügereihenfolge hat gegenüber anderen Systematiken den Vorteil eines geringen Rechenaufwands und kann bei einer großen Anzahl Standorte sogar nahezu vergleichbar gute Ergebnisse hervorbringen.⁴³

Kritische Betrachtung: In der Fachliteratur findet sich oftmals der Schluss, dass das Einfügeverfahren, insbesondere bei früher Konstruktion der groben Tour, deutlich bessere Ergebnisse liefert als das Verfahren des besten Nachfolgers.⁴⁴ Dadurch, dass einmal gewählte Verbindungen wieder ersetzt werden können, werden Ineffizienzen unwahrscheinlicher und geringfügiger. Gegenüber dem Nachfolgerverfahren ist jedoch ein höherer Rechenaufwand für das eigentliche Verfahren sowie für die Bestimmung der Einfügereihenfolge anzumerken.

2.2.6 Verbesserungsverfahren

Tourenverbesserungsverfahren gehen von einer bereits konstruierten, zulässigen Tour aus und versuchen diese so zu verändern, dass reduzierte Gesamtkosten resultieren. Beispiele für Verbesserungsverfahren sind die Austauschverfahren 2-opt, 3-opt oder Or-opt.⁴⁵ Aufgrund der begrenzten Relevanz für diese Arbeit wird auf diese hier jedoch nicht weiter eingegangen.

³⁸ Vgl. Vahrenkamp (2003) S. 205 f.

³⁹ Vgl. Vahrenkamp (2003) S. 210

⁴⁰ Vgl. Vahrenkamp (2003) S. 210

⁴¹ Vgl. Vahrenkamp (2003) S. 209 f.

⁴² Vgl. Neumann, Morlock (2002) S.445; Vahrenkamp(2003) S. 210; Arnold, Isermann, Kuhn, Tempelmeier, Furmans (2008) S. 148; Domschke, Drexl, Klein, Scholl (2015) S. 153

⁴³ Vgl. Vahrenkamp (2003) S. 210

⁴⁴ Vgl. Zimmermann, Stache (2001) S. 151; Vahrenkamp (2003) S. 205 ff.

⁴⁵ Siehe hierzu Vahrenkamp (2003) S. 216 ff. oder Neumann, Morlock (2002) S. 451 ff. für eine ausführliche Beschreibung sowie Arnold, Isermann, Kuhn, Tempelmeier, Furmans (2008) S. 149 für eine komprimierte Erläuterung.

3. Vorgehensweise

3.1 Abgrenzung/ Definition des Problems

Die Dynamik eines logistischen Systems setzt sich in der Realität aus einer Vielzahl von Faktoren zusammen, welche im Fall eines konkreten Problems unterschiedlich starken Einfluss haben. Für den Warentransport sind Zeitfenster, Fahrzeugauslastung, Pausenzeiten, Verkehrsaufkommen, Beladezeiten, Kraftstoffeffizienz nur einige dieser Faktoren.

Die Funktion eines Modells im Allgemeinen ist es, ein möglichst genaues, jedoch vereinfachtes Abbild der Realität herzustellen. Im Normalfall bedeutet dabei eine Zunahme der Genauigkeit auch die Zunahme an Komplexität. Es ist daher nicht leicht abzuwägen, wie viel Komplexität sinnvoll hingenommen werden kann und ab wann der Mehrwert an Genauigkeit nicht mehr im Verhältnis zum Mehraufwand steht.

Um eine angemessene Komplexität zu bewahren, beschränkt sich das in dieser Arbeit untersuchte Problem nur auf Zeitlimits bei den zu besuchenden Standorten. Dabei wird kein Zeitfenster angenommen, sondern nur eine Obergrenze, also ein Zeitpunkt zu dem der Punkt spätestens erreicht werden muss.

Zusätzlich wird das Problem so begrenzt, dass ein fixierter Startpunkt angenommen wird. Durch Einbezug des Zeitlimits kann der Startpunkt nicht mehr frei innerhalb einer gebildeten Tour verschoben werden. Folglich würde eine Verschiebung ein komplett anderes Ergebnis bedeuten, was die Anzahl möglicher Lösungen von $(n-1)!$ zu $n!$ anheben würde. Da in der Realität allerdings der Startpunkt in den meisten Fällen bekannt und fixiert ist – zum Beispiel im Falle eines Zentrallagers – macht es Sinn, die Komplexität durch einen fixierten Startpunkt zu begrenzen.

Ein beispielhaftes Abbild dieses Problems in der Realität wäre ein Haustransport oder Transportservice, welcher von einem Zentrallager aus Kleinteile an eine Vielzahl von Produktionsstandorten liefert. Die zeitlichen Limits ergeben sich dabei durch den Zeitpunkt, zu welchem die Teile in der Produktion benötigt werden.

3.2 Definition des Problems

Konkret beschreiben lässt sich das Problem durch die Dimensionen, welche für die Lösung relevant sind, sowie durch die Zielsetzung und die Definition der Bedingungen, unter denen eine Lösung zulässig ist. Im Folgenden wird dargestellt, inwieweit diese drei Faktoren im Vergleich zum klassischen Rundreiseproblem erweitert werden müssen:

Dimensionen (zur Lösung benötigte Größen):

- Klassisches Rundreiseproblem
 - Die vollständige Kostenmatrix mit den Kosten der Fahrt von jedem Punkt a zu einem anderen Punkt b
- Rundreiseproblem mit Zeitlimits:
 - Die vollständige Kostenmatrix
 - Die jeweiligen Limits, welche jedem Punkt zugeordnet sind
 - Verbindliche Vorgabe eines Startpunktes

Definition einer zulässigen Lösung:

- Klassisches Rundreiseproblem
 - Eine einzige, geschlossene Tour, welcher jeder Ort genau einmal zugeordnet ist
- Rundreiseproblem mit Zeitlimits:
 - Eine einzige, geschlossene Tour, welcher jeder Ort genau einmal zugeordnet ist
 - Die kumulierten Kosten bzw. die vergangene Zeit beim Erreichen jedes einzelnen Punktes muss unterhalb des entsprechenden Limits liegen.

Zielsetzung:

- Klassisches Rundreiseproblem
 - Minimierung der Gesamtkosten
- Rundreiseproblem mit Zeitlimits:
 - Minimierung der Gesamtkosten

Während die Zielsetzung hier gleich bleibt, begrenzt die neue Dimension der zeitlichen Limits die Anzahl an zulässigen Lösungen und birgt somit einen fundamentalen Unterschied zum klassischen Problem. So hat das klassische Rundreiseproblem so viele zulässige Lösungen wie es Reihenfolgen gibt, in denen alle Punkte einer Tour zugeordnet werden können. Sofern also mithilfe eines Lösungsverfahrens jeder Punkt der Tour nur genau einmal zugeordnet wird, ergibt sich zwangsläufig ein zulässiges Ergebnis.

Durch die zusätzliche Zulässigkeitsbedingung kann die Anzahl der zulässigen Lösungen mehr oder weniger stark begrenzt werden. So können durch die Vorgabe der neuen Bedingung zwischen 0% und 100% aller sonst zulässigen Touren unzulässig werden. Die Erwartungshaltung an ein heuristisches Lösungsverfahren darf daher nicht nur die Hoffnung auf ein möglichst kostengünstiges Ergebnis einschließen. Im Gegenteil muss der primäre Fokus darauf liegen, dass das Lösungsverfahren überhaupt eine unter den neuen Gesichtspunkten zulässige Lösung hervorbringen kann.

Für die Erklärung und die Bearbeitung der Methode werden hier Kosten, Zeit und Strecken gleichgesetzt. Also braucht es für das Zurücklegen einer Streckeneinheit immer genau eine Zeiteinheit und es fallen dafür immer genau Kosten in Höhe einer Kosteneinheit an. Das heißt, die in der Kostenmatrix hinterlegten Werte spiegeln analog sowohl monetäre und zeitliche Kosten sowie Streckeneinheiten wieder. Diese Annahme erlaubt es, einen Sachverhalt mit dem jeweils sinnvollsten Begriff zu beschreiben. So machen Limits kaum Sinn in Zusammenhang mit Kosten oder zurückgelegten Strecken. Gleichzeitig sind die Begriffe „Kostenmatrix“ und „Gesamtkosten“ gegenüber den zeitlichen Äquivalenten geläufiger.

Da der Startpunkt fixiert ist, wird des Weiteren unterschieden zwischen der Gesamtzahl an Orten n , in welcher der Start eingeschlossen ist, und der Anzahl zu besuchender Orte m , welche demnach $m = n-1$ beträgt.

3.3 Ansatz des Lösungsverfahrens

Als Grundlage für die Lösung des modifizierten Problems mit Zeitlimits dient das Einfügeverfahren oder Verfahren der sukzessiven Einbeziehung. Das Verfahren des besten Nachfolgers erscheint bereits in der klassischen Form zu fehleranfällig. Des Weiteren ist durch die Variable der Zeitlimits ein Faktor gegeben, welcher besonders in den späteren Iterationsschritten viel Flexibilität erfordert. Da die Auswahlmöglichkeiten des Verfahrens des besten Nachfolgers zum Ende hin abnehmen, ist diese Flexibilität nicht gegeben. Folglich wird das Einfügeverfahren verwendet, welches mit jedem Iterationsschritt an möglichen Einfügepositionen und somit Flexibilität gewinnt.

Um das modifizierte Problem lösen zu können, wird speziell die Systematik verändert, nach der die Einfügeposition ermittelt wird. Während beim klassischen Einfügeverfahren die Höhe der Einfügekosten als Entscheidungskriterium dient, werden für das modifizierte Verfahren drei verschiedene Kriterien vorgestellt:

Kostenkriterium: Wie die Bezeichnung suggeriert, dienen bei dieser Variante die Einfügekosten als Entscheidungskriterium für die Wahl der Einfügeposition, wodurch sich diese Variante nur kaum vom klassischen Verfahren unterscheidet. Die Einbeziehung der Limits geschieht in Form einer davor vorgenommenen Zulässigkeitskontrolle, durch die jene Positionen gesperrt werden können, welche zu unzulässigen Ergebnissen führen. Die zugrundeliegende Idee ist simpel: In jedem Schritt soll die kostengünstigste und somit effizienteste Einfügeposition gewählt werden, welche zu einem zulässigen⁴⁶ Teilergebnis führt. Gelingt es somit, in den ersten Schritten eine effiziente Tour zu generieren, sollte sich dies durch mehr Flexibilität auch positiv auf den Rest des Verfahrens auswirken können.

⁴⁶ Als zulässige Teiltour sei an dieser Stelle eine Tour definiert, welche sämtliche Limits der beteiligten Punkte einhält. Die Bedingung, dass alle Punkte genau einmal enthalten sein müssen, kann die Teiltour nicht erfüllen und ist somit nur für das finale Ergebnis relevant. Wenn jedoch eines der Limits innerhalb einer Teiltour überschritten wird, so führt dies zwangsläufig zur Nichteinhaltung der Bedingung in der finalen Tour. Demnach kann diese Zulässigkeitsbedingung auch auf Teiltouren übertragen werden.

Pufferkriterium: Das Entscheidungskriterium dieser Variante ermittelt sich auf Basis der Puffer, welche zwischen kumulierten Kosten und Limits an jedem Punkt der neuen Teiltour verbleiben. Die Entscheidung wird so vorgenommen, dass der minimale Puffer maximiert wird. Die Idee dahinter ist es, dass die Position gewählt wird, welche den großzügigsten Engpass hinterlässt. Als Engpass ist dabei der Punkt zu sehen, welcher innerhalb der gebildeten Teiltour den geringsten verbleibenden Puffer aufweist. Da durch jeden nachfolgenden Iterationsschritt die verbleibenden Puffer geringer werden, soll durch Maximierung des Engpasses die Wahrscheinlichkeit gesenkt werden, dass das Verfahren scheitert. Somit werden die Kosten als Zielgröße zugunsten einer Verbesserung der Erfolgsaussicht vernachlässigt.

Gewichtetes Kriterium: Dieses Kriterium errechnet sich sowohl aus den Kosten als auch aus dem verbleibenden Puffer und bietet somit den Mittelweg zwischen Kosten- und Erfolgsfokussierung. Auf die konkrete Funktionsweise sowie Hintergedanken dazu wird in Teil 4.1 eingegangen, nachdem auch die anderen beiden Kriterien näher erklärt worden sind.

Neben den Kriterien wird ebenfalls auf die Reihenfolge, in der die Punkte in die Tour einbezogen werden (**Einfügereihenfolge**) eingegangen. Da diese schon beim klassischen Einfügeverfahren einen nicht unerheblichen Einfluss auf das Ergebnis hat, ist es sinnvoll, auch hinsichtlich der Einfügereihenfolgen verschiedene Ansätze zu erarbeiten, welche einen positiven Einfluss auf den Ausgang haben könnten. Die Einfügereihenfolgen bilden zusammen mit den drei Varianten des Entscheidungskriteriums eine Vielzahl an unterschiedlichen **Lösungswegen**.

3.4 Evaluation der Methoden

Durch die Dynamik eines heuristischen Verfahrens - in diesem Fall speziell des zugrunde liegenden Einfügeverfahrens – ist es äußerst unwahrscheinlich, dass eine Systematik entwickelt werden kann, mit welcher jedes theoretisch lösbare Problem tatsächlich gelöst werden kann. Stellt man sich hier ein Problem mit 30 zu besuchenden Orten vor, welches nur eine einzige zulässige Lösung besitzt, so wird klar, wie unwahrscheinlich es ist, dass diese eine Lösung aus $30! = 2,65 * 10^{32}$ möglichen Lösungen erzielt wird.

Folglich soll diese Arbeit nicht nur Lösungsmethoden entwickeln, sondern vor allem auch untersuchen, wie gut diese tatsächlich auch zur Lösung des definierten Problems geeignet sind. Die Evaluation der aus Entscheidungskriterium und Einfügereihenfolge hervorgehenden Lösungswege bildet den Kern dieser Arbeit und wird in Teil 5 näher erläutert.

4. Entwicklung der Lösungsmethode

In diesem Teil wird das modifizierte Lösungsverfahren des Problems und dessen drei Varianten in Form der vorgestellten Entscheidungskriterien erklärt. Die Vorgehensweise zur Lösung wird schrittweise erklärt und soweit möglich in Form von Formeln dargestellt. Da sich die drei Varianten nur bei Wahl der Einfügeposition unterscheiden, wird auf die Unterschiede erst im entsprechenden Schritt eingegangen.

Es ist dazu anzumerken, dass die manuelle Lösung des Verfahrens aufgrund vieler rechenintensiver Zwischenschritte äußerst aufwendig ist. Die Arbeitsschritte sind daher auf eine durch Software unterstützte Lösung ausgelegt. Um die Formeln innerhalb einer Software nutzen zu können, müssen sie hinreichend präzise formuliert sein, was wiederum zu einer Vielzahl an benötigten Variablen führt. Eine Übersicht über diese findet sich im Variablenverzeichnis.

4.1 Schrittweise Erklärung des Lösungsverfahrens

Schritt 0: Vorbereitung: Um die Methode durchführen zu können wird eine vollständige **Kostenmatrix** benötigt, welche sämtliche **Fahrtzeiten** bzw. **Kosten $k(a, b)$** ⁴⁷ für die Wege von einem Ort a zu einem Ort b enthält. Dies schließt auch den als Startpunkt definierten Punkt ein. Alle Einträge mit $a=b$ müssen mit einem sehr hohen Wert (Für die Implementation in Teil 5. wurde 9999 gewählt) belegt werden, wodurch sie für die Verwendung in der Methode gesperrt sind.⁴⁸ Optional können auch sonstige ungewollte Strecken ebenfalls mit hohen Werten blockiert werden. Neben den Fahrtzeiten müssen von jedem Ort a auch die **Limits $L(a)$** bekannt sein, um die Methode durchführen zu können. Hat eine Station keine Begrenzung, kann dem entsprechenden Limit einfach ein sehr hoher Wert zugewiesen werden, welcher dann effektiv keinen Einfluss auf das Verfahren hat. Weist man dem Start- und Zielpunkt ein Limit zu, so wirkt sich dies als eine Obergrenze der Gesamtkosten aus. Sofern keine solche Obergrenze vorgesehen ist, wird **$L(0)$** ebenfalls ein sehr hoher Wert zugewiesen. Wie in Teil 3.2 beschrieben, werden Kosten, Zeiten und Strecken im Sinne dieser Erklärung gleichgesetzt. Ansonsten müssen Limits sowie die Elemente der Kostenmatrix auf dieselbe Einheit angeglichen werden um die Methode anwenden zu können.

Schritt 1: Sortierung der Matrix: Die **Kostenmatrix** wird so sortiert und benannt, dass der Startpunkt in der ersten Zeile und Spalte steht. Dieser wird von nun an als Punkt 0 bezeichnet. Die verbleibenden Punkte werden nach ihrem Zeitlimit sortiert in die Matrix eingefügt und aufsteigend nummeriert, mit dem kleinsten Limit beginnend. Somit wird also der Punkt mit dem knappsten (kleinsten) Limit in die zweite Zeile/ Spalte als Punkt 1 eingefügt, während in der letzten, n-ten Zeile und Spalte dann der Punkt m mit dem großzügigsten (größten) Limit steht. Dieser Schritt ist nicht zwangsläufig erforderlich, macht

⁴⁷ Um bei den komplexeren Formeln Übersichtlichkeit zu wahren, wurde diese Schreibweise der in der Literatur weiter verbreiteten Schreibweise k_{ab} vorgezogen.

⁴⁸ Dies sind also die Strecken von einem Punkt zu sich selbst. Da eine solche Verbindung nicht zulässig ist, wird sie gesperrt.

jedoch die Matrix deutlich übersichtlicher und verringert die Komplexität einiger Rechenschritte, gerade bei einer automatisierten Lösung durch Software. Abbildung 7 zeigt eine beispielhafte, sortierte Kostenmatrix mit $n=4$ bzw. $m=3$ Orten sowie die zugewiesenen Limits.

		Nach			
		Punkt 0	Punkt 1	Punkt 2	Punkt 3
Von	Punkt 0	$k(0,0)=999$	$k(0,1)=3$	$k(0,2)=5$	$k(0,3)=2$
	Punkt 1	$k(1,0)=3$	$k(1,1)=999$	$k(1,2)=4$	$k(1,3)=1$
	Punkt 2	$k(2,0)=5$	$k(2,1)=4$	$k(2,2)=999$	$k(2,3)=4$
	Punkt 3	$k(3,0)=2$	$k(3,1)=1$	$k(3,2)=4$	$k(3,3)=999$
Limits		$L(0)=999$	$L(1)=20$	$L(2)=30$	$L(3)=30$

Abbildung 7: Beispielhafte Kostenmatrix mit $n=4$ Orten und deren Zeitlimits

Schritt 2: Ermittlung der Einfügereihenfolge: Für die Lösung der Methode muss eine konkrete Methodik zur Bestimmung der Einfügereihenfolge ausgewählt und anhand dieser die resultierende Reihenfolge errechnet werden. Als Ergebnis erhält man die Einfügereihenfolge $P = (p_1, p_2, \dots, p_m)$.

Analog zur Vorabbestimmung in diesem Schritt kann auch während jedes Iterationsschritts jeweils der nächste einzufügende Punkt bestimmt werden, jedoch wird dieser Schritt hier der Übersicht halber vor Beginn der Iteration vorgenommen. Auf verschiedene Ansätze zur Bestimmung der Einfügereihenfolge wird in Teil 4.3 ausführlich eingegangen.

Schritt 3: Vorbereitung der Iteration: Wie in Teil 3.2 erläutert, ist für das Problem ein fixierter Startpunkt definiert, welcher gleichzeitig auch der letzte Punkt der Tour sein muss. Wie im Schritt 1 vorgegeben ist dies immer Punkt 0. Als Vorbereitung des ersten Iterationsschritts bildet dieser mit sich selbst die Starttour $T_0=(0, 0)$.

Schritt 4: Durchführung der Iteration in m Iterationsschritten: Die Iterationsschritte verlaufen grundsätzlich ähnlich zu denen eines klassischen Einfügeverfahrens. Im Iterationsschritt s wird der Punkt an s -ter Stelle der Einfügereihenfolge p_s der bestehenden Tour T_{s-1} hinzugefügt. Dies ist nur zwischen zwei benachbarten Punkten von T_{s-1} möglich. Daher gibt es im Iterationsschritt s auch immer genau s verschiedene Möglichkeiten p_s einzufügen. Das liegt daran, dass die Tour T_s immer $s+2$ Punkte enthält: Start- und Endpunkt sowie s weitere. Daraus folgt, dass im ersten Iterationsschritt nur eine einzige Position - zwischen Start- und Endpunkt - möglich ist. Die Einfügemöglichkeiten werden im Folgenden **Einfügepositionen** genannt und mit z bezeichnet.

Die Schritte 4.1 bis 4.3 erläutern den genauen Ablauf eines Iterationsschritts. Da im Iterationsschritt 1 nur eine Möglichkeit der Einfügung existiert, können die Schritte 4.1 bis 4.3 übersprungen werden und die neue Tour $T_1=(0, p_1, 0)$ kann direkt gebildet werden.

Schritt 4.1: Bestimmung der Einfügekosten: Wie beim klassischen Einfügeverfahren werden zunächst für jede mögliche Einfügeposition die Einfügekosten berechnet. Diese stellen die Mehrkosten bzw. den zeitlichen Zusatzbedarf dar, um welche sich die bisherige Tour T_{s-1} vergrößern würde. Wie in Teil 2.2.5 erläutert lassen sich die Einfügekosten E_c von einem Punkt c zwischen zwei benachbarte Punkte a und b wie folgt berechnen:

$$E_c = k(a, c) + k(c, b) - k(a, b)$$

Da im Iterationsschritt s immer der Punkt p_s der Tour hinzugefügt wird, muss also $c = p_s$ gelten. Somit lässt sich die Formel wie folgt konkretisieren:

$$E_{p_s} = k(a, p_s) + k(p_s, b) - k(a, b)$$

Um diese Formel noch weiter konkretisieren zu können, sind zunächst einige Definitionen notwendig:

Die Teiltour, welche im Iterationsschritt s gebildet wird sei dafür wie folgt beschrieben:

$$T_s = (i_{s;0}, i_{s;1}, \dots, i_{s;s}, i_{s;s+1})$$

Demzufolge beschreibt $i_{s;j}$ den Punkt, welcher im Iterationsschritt s an j -ter Stelle der Tour steht. An 0-ter und $(s+1)$ -ter Stelle steht dabei nach jedem Iterationsschritt der Start- und Endpunkt, welcher als Punkt 0 bezeichnet wurde.

Des Weiteren sei die **Einfügeposition** z definiert als Position zwischen den Punkten, welche in der bisherigen Tour T_{s-1} an $(z-1)$ -ter und z -ter Position standen. Demzufolge beschreibt beispielsweise die Einfügeposition $z=1$ im Iterationsschritt $s=2$ die Position zwischen den Punkten $i_{1;0}$ und $i_{1;1}$, also dem Startpunkt 0 und dem in T_1 darauf folgenden Punkt. Dies bedeutet auch, dass der an Position z eingefügte Punkt in der neuen Tour immer an z -ter Stelle steht.

Unter Beachtung dieser Definitionen, kann die Formel für die Einfügekosten des Punktes p_s in Abhängigkeit der **Einfügeposition** z konkret beschrieben werden:

$$E_{p_s}(z) = k(i_{s-1; z-1}, p_s) + k(p_s, i_{s-1; z}) - k(i_{s-1; z-1}, i_{s-1; z})$$

Für $z = (1, \dots, s)^{49}$

Mit:

s = Iterationsschritt

z = Einfügeposition

p_s = In Schritt s einzufügender Punkt

$i_{s-1; z-1}$ = Punkt der bestehenden Tour direkt vor der Einfügeposition

$i_{s-1; z}$ = Punkt der bestehenden Tour direkt hinter der Einfügeposition

$k(i_{s-1; z-1}, p_s)$ und $k(p_s, i_{s-1; z})$ = Kosten für die Streckenabschnitte, welche durchs Einfügen hinzukommen

$k(i_{s-1; z-1}, i_{s-1; z})$ = Kosten des Streckenabschnitts, welcher durchs Einfügen entfällt

Schritt 4.2 Zulässigkeitskontrolle: Die Zulässigkeitskontrolle wird als zusätzlicher Schritt im Vergleich zum klassischen Einfügeverfahren hinzugefügt, um die Bildung von unzulässigen Touren zu verhindern. Dazu muss für jede der Einfügepositionen geprüft werden, ob durch das Einfügen eine zulässige Teiltour entstehen würde. Das heißt, dass das Zeitlimit von keinem der Punkte der neuen Teiltour überschritten werden darf. Da sich durch das Einfügen weiterer Punkte zur Tour die kumulierte Zeit, welche zu einem Punkt benötigt wird, nur erhöhen und nicht verringern kann, wäre das Endergebnis auf jeden Fall unzulässig, sobald in einem der Iterationsschritte ein Limit überschritten wird.

Die Zulässigkeitskontrolle, welche für jede der s Einfügepositionen durchgeführt werden muss, kann wiederum aufgeteilt werden in 3 Schritte:

Schritt 4.2a: Prüfung des einzufügenden Punktes: Zunächst wird geprüft, ob das Zeitlimit des einzufügenden Punktes an dieser Einfügeposition eingehalten werden kann. Geprüft wird dies, indem der verbleibende Zeitpuffer bestimmt wird, welcher, nachdem dieser Punkt erreicht wird, verbleibt. Dieser berechnet sich aus dem Zeitlimit des einzufügenden Punktes abzüglich der kumulierten Kosten⁵⁰, welche bis zum Erreichen des Punktes anfallen. Diese kumulierten Kosten lassen sich am einfachsten bestimmen, indem auf die kumulierten Kosten beim Vorgänger-Punkt die Strecke von diesem zum einzufügenden Punkt hinzugerechnet wird. Um dies durch Formeln auszudrücken, sind zunächst wieder einige Definitionen erforderlich. Die kumulierten Kosten zur Erreichung eines Punktes a innerhalb

⁴⁹ Die Berechnung muss für jede Einfügeposition z vorgenommen werden. Da es im Iterationsschritt s immer auch genau s Einfügepositionen gibt, entspricht s also auch der Anzahl der zu bestimmenden Werte.

⁵⁰ In diesen ist auch hier wieder die Zeit zu sehen, welche zum Erreichen des Punktes benötigt wird.

der Tour T_s seien bezeichnet als $K_s(a)$.⁵¹ Wie schon in Schritt 4.1 hergeleitet, ist der Punkt vor dem Einzufügenden immer der $(z-1)$ -te Punkt der im letzten Iterationsschritt $(s-1)$ gebildeten Tour: $i_{s-1; z-1}$

Dessen kumulierte Kosten am Ende des letzten Iterationsschritts können also als $K_{s-1}(i_{s-1; z-1})$ bezeichnet werden. Der verbleibende Zeitpuffer bzw. das Defizit des einzufügenden Punktes p_s , welches in Abhängigkeit von der Einfügeposition z entsteht, sei als $D_p(z)$ definiert und lässt sich wie folgt berechnen:

$$D_p(z) = L(p_s) - [K_{s-1}(i_{s-1; z-1}) + k(i_{s-1; z-1}, p_s)]$$

Für $z = (1, \dots, s)$

Mit:

s = Iterationsschritt

z = Einfügeposition

p_s = In Schritt s einzufügender Punkt

$L(p_s)$ = Limit von p_s

$i_{s-1; z-1}$ = Punkt der bestehenden Tour vor der Einfügeposition

$K_{s-1}(i_{s-1; z-1})$ = kumulierte Kosten zur Erreichung dieses Punktes

$k(i_{s-1; z-1}, p_s)$ = (zeitliche) Kosten für die Strecke von $i_{s-1; z-1}$ zu p_s

Schritt 4.2b: Prüfung der nachfolgenden Punkte: Für alle nachfolgenden Punkte - also Punkte, die in der bisherigen Tour hinter der Einfügeposition liegen – muss ebenfalls geprüft werden, ob ihre jeweiligen Limits noch eingehalten werden. Dazu werden die Einfügekosten $E_{p_s}(z)$ von den bisherigen Zeitpuffern der nachfolgenden Punkte abgezogen. Die Zeitreserve eines Punktes a , welcher Teil der Tour T_s ist, sei als $R_s(a)$ bezeichnet.⁵² Relevant für die Bewertung einer Einfügeposition ist jeweils nur der kritischste, also kleinste der nachfolgenden Zeitpuffer. Von diesem wird der Einfügearbeit abgezogen, um zu prüfen, ob nach Einfügung noch ein zulässiges Ergebnis erzielt werden würde. Diese Differenz – also der minimale Zeitpuffer, welcher nach dem Einfügevorgang an einem der hinter der Einfügeposition gelegenen Punkte verbleiben würde – sei als $D_{\min}(z)$ bezeichnet und ist wie folgt zu ermitteln:

⁵¹ Anmerkung: Die kumulierten Kosten werden erst in Schritt 4.5 ermittelt, um sie in Schritt 4.2 der nachfolgenden Iteration zu verwenden. Im ersten Iterationsschritt liegen demnach noch keine Werte vor. Da die Unterpunkte 4.1-4.3 dort allerdings übersprungen werden, werden sie auch nicht benötigt.

⁵² Anmerkung: Wie die kumulierten Kosten werden diese im Schritt 4.5 vorbereitend ermittelt.

$$D_{\min}(z) = \min[R_{s-1}(i_{s-1; z}), \dots, R_{s-1}(i_{s-1; s})] - E_{p_s}(z)$$

Für $z = (1, \dots, s)$

Mit:

$R_{s-1}(i_{s-1; z})$ = (zeitlicher) Puffer, welcher nach dem letzten Iterationsschritt $s-1$ am Punkt $i_{s-1; z}$ verbleibt.

$i_{s-1; z}$ = Der Punkt, welcher direkt hinter der Einfügeposition z liegt, also der erste Punkt, dessen Zeitpuffer durch Einfügen des neuen beeinflusst wird.

$R_{s-1}(i_{s-1; s})$ = Puffer des letzten Punktes der bisherigen Tour (dies ist immer Punkt 0)

$E_{p_s}(z)$ = Einfügekosten von Punkt p_s an Einfügeposition z .

p_s = Punkt, welcher der Einfügereihenfolge nach in Iterationsschritt s eingefügt wird.

Die Größe der Gruppe aus Zeitpuffern, aus denen das Minimum bestimmt wird, ist abhängig von der Einfügeposition. Für alle Punkte, welche vor der Einfügeposition liegen, ändert sich nichts an der Zeit, zu der sie erreicht werden. Demnach muss nur für alle Punkte hinter der Einfügeposition – angefangen mit dem Punkt, welcher bisher diese Position belegte $i_{s-1; z}$ – geprüft werden ob deren Limits eingehalten werden. Für die letzte Einfügeposition $z=s$ bedeutet das beispielsweise, dass nur der Zielpunkt $i_{s-1; s} = 0$ relevant ist.

Schritt 4.2c: Zusammenführung der Kriterien: Um zu bewerten, ob das Einfügen an einer Position z ein zulässiges Zwischenergebnis produziert, sind beide in den Teilschritten a und b errechneten Zeitpuffer $D_p(z)$ und $D_{\min}(z)$ relevant. Nur wenn beide einen positiven Betrag haben, ist die resultierende Tour von den Limits her zulässig. Demnach wird als finales Zulässigkeitskriterium $D_{zu}(z)$ definiert und als kleinerer der beiden Werte bestimmt:

$$D_{zu}(z) = \min[D_p(z), D_{\min}(z)]$$

Dieses Zulässigkeitskriterium sagt aus, welches der geringste Zeitpuffer innerhalb der neuen Tour wäre, wenn der Punkt an Position z eingefügt wird. Ausgenommen davon sind die Zeitpuffer der Punkte vor der Einfügeposition, da sich diese nicht verändern und daher für die Entscheidung nicht relevant sind. Ist der errechnete Wert von $D_{zu}(z)$ negativ, so werden durch den Einfügevorgang an Position z ein oder mehrere Limits überschritten, wodurch ein unzulässiges Ergebnis zustande käme.

Für die Verwendung im nachfolgenden Schritt werden zuletzt noch der beste, also höchste dieser Werte D_{opt} und die zugehörige Einfügeposition $z(D_{opt})$ bestimmt.

$$D_{opt} = \max[D_{zu}(1), \dots, D_{zu}(s)]$$

Schritt 4.3: Bestimmung des Entscheidungskriteriums: In diesem Schritt wird bestimmt, an welcher Position p_s eingefügt wird. Dies ist der einzige Schritt in dem sich die drei Varianten **Kostenkriterium**, **Pufferkriterium** und **Gewichtetes Kriterium** unterscheiden.

Kostenkriterium

Schritt 4.3 - Kostenkriterium: Wie in Teil 3.3 erläutert, beruht die Wahl der Einfügeposition hier auf der Höhe der Einfügekosten. Vor der Auswahl der Position wird jedoch auf Basis der Zulässigkeitskontrolle eine Sperrung unzulässiger Positionen durchgeführt.

Schritt 4.3a - Kostenkriterium: Sperren unzulässiger Positionen: Zunächst wird geprüft, ob überhaupt ein zulässiger Einfügevorgang möglich ist. Dazu wird der in Schritt 4.2c errechnete Maximalwert D_{opt} herangezogen.

Option A: $D_{opt} \geq 0$: Es gibt noch mindestens eine Möglichkeit, eine zulässige Zwischenlösung zu erzeugen. Daher müssen alle nicht zulässigen Einfügepositionen gesperrt werden. Dies ist am einfachsten möglich, indem die Einfügekosten $E_{p_s}(z)$ von allen Positionen z , welche ein negatives Zulässigkeitskriterium $D_{zu}(z) < 0$ aufweisen, mit einem sehr hohen Wert⁵³ überschrieben werden.

Option B: $D_{opt} < 0$: In diesem Fall gibt es keine zulässige Einfügeposition, was bedeutet, dass das Verfahren gescheitert ist. Wenn ausschließlich eine zulässige Lösung gesucht wird, so kann das Verfahren abgebrochen werden. Alternativ kann das Verfahren fortgeführt werden, um idealerweise eine möglichst gute unzulässige Lösung zu erzielen. Würden wie bei Option A alle unzulässigen Positionen gesperrt werden, bliebe keine Position übrig. Da das keinen Sinn macht, wird zur Fortführung des Verfahrens keine der Positionen blockiert.

Schritt 4.3b - Kostenkriterium: Auswahl der Einfügeposition: Aus allen nicht blockierten Einfügepositionen wird die mit den geringsten Einfüge- bzw. Gesamtkosten ausgewählt. Das heißt aus den Einfügekosten aller Positionen $E_{p_s}(z)$ wird der geringste Wert ermittelt und als E_{opt} bezeichnet. Zu diesem wird dann die zugehörige Einfügeposition $z(E_{opt})$ bestimmt.

Die im nachfolgenden Schritt verwendete Einfügeposition z_{opt} entspricht beim Kostenkriterium also $z(E_{opt})$.

Pufferkriterium

Schritt 4.3 - Pufferkriterium: Beim Pufferkriterium wird die Einfügeposition basierend auf dem in Schritt 4.2c errechneten geringsten, entstehenden Zeitpuffer $D_{zu}(z)$ gewählt. Dieser repräsentiert den Engpass, welcher sich für die neue Tour ergeben würde. Idee hinter dem Pufferkriterium ist es, einen möglichst großzügigen Engpass zu erzielen, um somit auch möglichst viel Spielraum für die nachfolgenden Iterationsschritte zu haben. Die Einfügekosten spielen hier keine Rolle, müssen daher auch nicht blockiert werden.

⁵³ Auch hier wird in der Implementation wieder 9999 gewählt.

Da der Engpass maximiert werden soll, wird der bereits ermittelte Wert D_{opt} als Optimum angenommen. Demnach wird auch z_{opt} in Abhängigkeit von D_{opt} bestimmt: $z_{opt} = z(D_{opt})$. Ist D_{opt} kleiner als 0, so ist das Verfahren auch hier gescheitert und kann entweder weitergeführt oder abgebrochen werden. Wird das Verfahren trotz Scheiterns fortgesetzt, ändert dies nichts an der Auswahl von z_{opt} .

Gewichtetes Kriterium

Schritt 4.3 - Gewichtetes Kriterium: Die Variante des gewichteten Kriteriums ist eine Kombination aus den beiden anderen Kriterien. Für die Entscheidung der Einfügeposition sind sowohl die Kosten, als auch der minimale verbleibende Puffer relevant.

Schritt 4.3a - Gewichtetes Kriterium: Sperren unzulässiger Positionen: Dieser Schritt verläuft identisch zum Kostenkriterium. Auch hier werden, sofern eine zulässige Tour gebildet werden kann, die Einfügekosten der unzulässigen Positionen blockiert.

Schritt 4.3b - Gewichtetes Kriterium: Auswahl der Einfügeposition: Die Wahl der Einfügeposition z_{opt} geschieht auf Basis des Kriteriums $G(z)$, welches sich aus den Einfügekosten $E_{p_s}(z)$ und den minimalen verbleibenden Zeitpuffern $D_{zu}(z)$ berechnet. Diese beiden Kriterien können unterschiedlich gegeneinander gewichtet werden, wodurch sich das gewichtete Kriterium mehr in die eine oder andere Richtung ausrichten lässt. Die beiden Gewichtungen werden als g_{kost} und g_{puff} bezeichnet. Diesen können beliebige Werte zugewiesen werden, relevant für den Ausgang der Methode ist ausschließlich deren Verhältnis zueinander. Ein hoher Wert (im Vergleich zum anderen) bedeutet eine hohe Gewichtung, also einen stärkeren Einfluss auf die Entscheidung.

Das Entscheidungskriterium $G(z)$ wird konkret ermittelt, indem für jede Einfügeposition z geprüft wird, wie weit die beiden Kriterien $E_{p_s}(z)$ und $D_{zu}(z)$ vom den Optima E_{opt} und D_{opt} entfernt sind. Diese Differenzen werden dann erst gewichtet und anschließend addiert um das gewichtete Entscheidungskriterium $G(z)$ zu bilden. Das heißt:

Kostendifferenz zwischen Einfügeposition z und dem Optimum: $E_{dif}(z) = E_{p_s}(z) - E_{opt}$

Differenz des minimalen Puffers bei Position z zum Optimum: $D_{dif}(z) = D_{opt} - D_{zu}(z)$

Daraus ergibt sich die folgende Formel für die Ermittlung von $G(z)$:

$$G(z) = E_{dif}(z) * g_{kost} + D_{dif}(z) * g_{puff}$$

Für alle Formeln gilt: $z = (1, \dots, s)$

Der beste Wert, den $G(z)$ annehmen kann, ist 0. Dies geschieht, wenn sowohl $E_{p_s}(z) = E_{opt}$ und $D_{zu}(z) = D_{opt}$ sind. Der niedrigste Wert aller $G(z)$ wird als G_{opt} bezeichnet und zur Bestimmung der Einfügeposition genutzt. Also gilt hier: $z_{opt} = z(G_{opt})$.

$$G_{opt} = \min[G(1), \dots, G(s)]$$

Welche Auswirkung die Gewichtung haben kann, soll am nachfolgenden Beispiel für einen Einfügevorgang im dritten Iterationsschritt, also mit drei Einfügepositionen gezeigt werden:

Position:	z=1	z=2	z=3
$E_{ps}(z)$	2	3	5
$E_{dif}(z)$	0	1	3
$D_{zu}(z)$	17	20	21
$D_{dif}(z)$	4	1	0

Tabelle 8 zeigt die Ausgangssituation, aus der hier bereits die für die Entscheidung relevanten Differenzen $E_{dif}(z)$ und $D_{dif}(z)$ errechnet worden sind. Aus diesen wird im nächsten Schritt unter Verwendung verschiedener Gewichtungen das Entscheidungskriterium $G(z)$ errechnet.

Abbildung 8: Beispielhafte Ausgangssituation für die Wahl einer Einfügeposition

		Ergebnisse für $G(z)$			
g_{kost}	g_{puff}	z=1	z=2	z=3	z_{opt}
1	1	4	2	3	z=2
5	1	4	6	15	z=1
1	3	12	4	3	z=3

Abbildung 9: Auswirkung verschiedener Gewichtungen auf die Wahl der Einfügeposition

Wie in Tabelle 9 ersichtlich, werden hier durch drei unterschiedliche Gewichtungen drei unterschiedliche Einfügepositionen ausgewählt. Kostenkriterium und Pufferkriterium sind eigentlich die beiden entgegengesetzten Extremfälle des gewichteten Kriteriums. Eine Gewichtung von $g_{kost}=1$ zu $g_{puff}=0$ entspricht dem Kostenkriterium, die umgekehrte Gewichtung dem Pufferkriterium. Da allerdings jeweils nur einer der Faktoren lösungsrelevant ist, können diese Verfahren abgekürzt werden.

Schritt 4.4: Bildung der neuen Tour: In diesem Schritt wird basierend auf der gewählten Einfügeposition z_{opt} die neue Tour T_s bestimmt. Dazu wird der einzufügende Punkt p_s vor dem Punkt $i_{s-1; z_{opt}}$ der vorherigen Tour T_{s-1} eingefügt. Alle Punkte vor der Einfügeposition bleiben von T_{s-1} zu T_s unverändert, alle darauf folgenden Punkte werden in der gleichen Reihenfolge hinter p_s angeordnet. Dies lässt sich wie folgt durch Formeln ausdrücken:

$$i_{s; z_{opt}} = p_s$$

$$i_{s; j} = i_{s-1; j} \quad \text{Für alle } j = \{0, 1, \dots, (z_{opt} - 1)\}$$

$$i_{s; j} = i_{s-1; j-1} \quad \text{Für alle } j = \{(z_{opt} + 1), \dots, s, (s + 1)\}$$

Schritt 4.5: Vorbereitung des nächsten Iterationsschritts: Dieser Schritt dient als Vorbereitung für den nächsten Iterationsschritt sowie in der letzten Iteration $s=m$ zur Ermittlung des Endergebnisses. Für die Verwendung in Schritt 4.2 des nachfolgenden Iterationsschritts müssen die kumulierten Kosten und die verbleibenden Puffer für die Punkte der neuen Tour T_s bestimmt werden. Im ersten Iterationsschritt werden diese nicht benötigt, da Schritt 4.2 nicht durchgeführt wird. Es werden allerdings vorbereitend für $s=2$ die entsprechenden Werte errechnet.

Schritt 4.5a: Bestimmung der kumulierten Kosten: Für jeden Punkt in der neuen Tour T_s werden die Zeiten bzw. Kosten errechnet, welche zur Erreichung dieses Punktes benötigt werden. Diese kumulierten Kosten bei Erreichung eines Punktes a nach Iterationsschritt s werden als $K_s(a)$ bezeichnet. Für die kumulierten Kosten am Startpunkt gilt unabhängig vom Iterationsschritt logischerweise immer $K_s(i_s; 0)=0$. Für die darauf folgenden Punkte $i_s; j$, welche an j -ter Stelle von T_s stehen, gilt:

$$K_s(i_s; j) = K_s(i_s; j-1) + k(i_s; j-1, i_s; j)$$

Für $j=\{1, \dots, s+1\}$

Mit:

$K_s(i_s; j-1)$ = Kumulierte Kosten am Punkt, welcher direkt vor $i_s; j$ in der Tour liegt.

$k(i_s; j-1, i_s; j)$ = Kosten für die Fahrt vom vorherigen Punkt zu $i_s; j$

Da die Formel die jeweils vorherigen kumulierten Kosten benötigt, muss mit $j=1$ begonnen werden und nacheinander mit aufsteigenden j durchgeführt werden bis $j=s+1$ erreicht wird.

Schritt 4.5b: Bestimmung der verbleibenden Puffer: Basierend auf den kumulierten Kosten kann abschließend der verbleibende Puffer, also der Abstand der aktuellen kumulierten Kosten zum Limit, bestimmt werden. Diese Zeitreserve eines Punktes a nach Iterationsschritt s wird als $R_s(a)$ bezeichnet und berechnet sich wie folgt:

$$R_s(a) = L(a) - K_s(a)$$

Der verbleibende Puffer muss für alle Punkte, die Teil von T_s sind, ermittelt werden. Das heißt die Formel lässt sich konkretisieren:

$$R_s(i_s; j) = L(i_s; j) - K_s(i_s; j)$$

Für $j=\{1, \dots, s+1\}$

Sofern alle dieser Zeitpuffer einen positiven Betrag haben, verstößt die Tour gegen keine Limitrestriktion. Ein negativer Puffer wird als Defizit oder Verspätung bezeichnet und sorgt für ein unwiderruflich unzulässiges Ergebnis.

Abbruchbedingung: Wenn alle Punkte der Tour hinzugefügt worden sind, also nachdem der letzte Punkt der Einfügereihenfolge hinzugefügt wurde, ist die Tour komplett. Dies ist nach m Durchläufen der Fall, was bedeutet, dass die Iteration nach Schritt 4.5 von Iterationsschritt $s=m$ endet.

Für ein aussagekräftiges Ergebnis des Verfahrens werden drei Faktoren benötigt:

1. Die finale Tour, welche im letzten Iterationsschritt gebildet wurde:

$$T_m = (i_{m;0}, i_{m;1}, \dots, i_{m;m}, i_{m;n})$$

2. Die Gesamtkosten K_{ges} der Tour, welche den kumulierten Kosten des letzten Punktes der finalen Tour entsprechen: $K_{ges} = K(i_{m;n})$
3. Die Summe aller Verspätungen, also der Gesamtbetrag um den die Limits überschritten worden sind. Dieser wird bezeichnet als V_{ges} und berechnet sich als Betrag der Summe aller $R_m(a)$, welche einen negativen Betrag haben. Dies gilt für alle Punkte $a=\{1, \dots, m, n\}$.⁵⁴

Wenn das Verfahren erfolgreich ein zulässiges Ergebnis produziert hat, so ist $V_{ges} = 0$, da alle Limits unterschritten worden sind. Ist das Verfahren gescheitert, so gibt V_{ges} an, um wie weit eine zulässige Lösung verfehlt worden ist, bzw. eine wie große Gesamtverspätung in Kauf genommen werden muss, wenn T_m so akzeptiert wird.

4.2 Bewertung des Ergebnisses

Wie schon in Teil 3 angedeutet, ist allein die Betrachtung der Gesamtkosten für die Bewertung nicht ausreichend. Da aus der Durchführung der Methode nicht zwangsläufig ein zulässiges Ergebnis resultiert, bildet die **Erfolgsquote** den wichtigsten Bewertungsfaktor. Diese soll aussagen, mit welcher Wahrscheinlichkeit unter gewissen Rahmenbedingungen eine zulässige Lösung erzielt werden kann. Die Höhe der **Gesamtkosten** bleibt jedoch der zweite wichtige Bewertungsfaktor. Beide Faktoren werden in Teil 5.4 genauer beschrieben.

4.3 Einfügereihenfolgen

Denkbar sind verschiedene Kriterien, nach denen die Einfügereihenfolge bestimmt werden kann. Im Folgenden werden daher einige Einfügesystematiken vorgestellt und die Vorgehensweise ihrer Ermittlung erklärt.

1. **Aufsteigend nach Höhe des Limits sortiert:** Bei dieser Systematik werden die Punkte sortiert nach der Höhe ihrer Limits – beginnend mit dem niedrigsten – der Einfügereihenfolge zugeordnet. Daher ist diese Systematik speziell für das modifizierte Rundreiseproblem mit Limits anwendbar.

Da die Kostenmatrix in der Vorbereitung des Verfahrens nach Höhe des Limits sortiert wird, ergibt sich die sehr einfach zu bestimmende Einfügereihenfolge:

$$P = (1, 2, \dots, m)$$

⁵⁴ Punkt n , welcher der Start- und Zielpunkt ist, muss nur einbezogen werden, sofern diesem ein Limit zugewiesen wurde.

Die Idee dieser Einfügesystematik ist es, dass in jedem Iterationsschritt der verbleibende Punkt eingefügt wird, dessen Limit am knappsten ist. Das heißt, dass zuerst der am schwierigsten einzufügende Punkt hinzugefügt wird. Darauf folgt dann in jedem Schritt ein großzügiger limitierter Punkt, welcher somit einfacher einzufügen ist.

Aufgrund der ansteigenden Limits wird tendenziell eine Einfügeposition im hinteren Teil der bestehenden Tour gewählt werden. Theoretisch sollte auch dies eine positive Auswirkung auf die Verlässlichkeit der Methode haben, da ein Einfügevorgang in den hinteren Einfügepositionen die Puffer von weniger Punkten reduziert. Somit sollte für die nachfolgenden Schritte mehr Flexibilität bewahrt werden.

Aufgrund dieser Faktoren ist durch Verwendung dieser Einfügereihenfolge auf eine vergleichsweise hohe Erfolgsquote zu hoffen. Hinsichtlich der resultierenden Gesamtkosten ist nicht mit den besten Ergebnissen zu rechnen, da die Systematik keine Schritte vornimmt, welche diese explizit beeinflussen.

2. **Absteigend nach Höhe des Limits sortiert:** Diese Systematik ist das exakte Gegenteil der Ersten, wodurch auch die resultierende Einfügereihenfolge exakt invertiert ist:

$P = (m, \dots, 2, 1)$

Durch die in jedem Schritt verringerten Limits, sollten durch die Lösungsmethodik tendenziell Einfügepositionen am Anfang der Tour ausgewählt werden. Aus den gleichen Gründen, aus denen die Systematik der aufsteigenden Limits eine gute Erfolgsquote hervorbringen sollte, ist hier - insbesondere in Kombination mit dem Kostenkriterium - mit einer vergleichsweise schlechten Quote zu rechnen. Das begründet sich darin, dass die letzten einzufügenden Punkte aufgrund ihres niedrigen Limits nur in den ersten Einfügepositionen eingefügt werden können. Wenn sich allerdings hinter diesen Positionen ein Punkt mit sehr geringem verbleibendem Puffer befindet, so können die Einfügekosten sehr leicht zu einer Limitüberschreitung führen. Insbesondere das Kostenkriterium verwendet keine Systematik, welche diese Konstellation und somit das wahrscheinliche Scheitern des Verfahrens verhindert.

3. **Aufsteigend nach „Entfernung vom Startpunkt“ sortiert:** Beginnend mit dem Punkt welcher vom Startpunkt aus am günstigsten zu erreichen ist, wird durch diese Systematik die Einfügereihenfolge in Abhängigkeit der Entfernung vom Start bestimmt. Die Entfernung ist gleichbedeutend mit dem Kosten- und Zeitbedarf für die Überwindung der Strecke. Die Punkte $b = (1, 2, \dots, m)$ werden also sortiert nach ihrer jeweiligen Entfernung zum Start $k(0; b)$ der Einfügereihenfolge P zugeordnet.

4. **Absteigend nach „Entfernung vom Startpunkt“ sortiert:** Wie bei den limitorientierten Einfügereihenfolgen ist diese Systematik das Gegenstück zu Systematik 3. Die Punkte werden auf dieselbe Weise sortiert, jedoch in umgekehrter Reihenfolge zu P hinzugefügt.

Diese Einfügereihenfolge orientiert sich an den unter anderem von Neuman, Morlock und Vahrenkamp⁵⁵ beschriebenen Ansätzen, bei denen als Starttour die zwei am

⁵⁵ Vgl. Neumann, Morlock (2002) S.445; Vahrenkamp (2003) S. 210

weitesten voneinander entfernten Punkte gewählt werden, um früh die generelle Struktur der Tour festzulegen. Ist diese Grundstruktur früh bekannt, lassen sich tendenziell effizientere Einfügeentscheidungen treffen. Da der Startpunkt jedoch nicht frei gewählt werden kann, wird hier nur der erste einzufügende Punkt möglichst weit entfernt gelegt. Auch aus Sicht der Limiteinhaltung ist es in der Theorie sinnvoll, zuerst die am weitesten vom Start entfernten Punkte anzusteuern. Der Puffer $R_s(a)$ eines Ortes a wird nämlich mindestens um den Zeitbedarf vom Start zu a reduziert. Das heißt also, dass der Puffer und somit die Flexibilität eines Punktes sich stärker reduzieren, je weiter dieser vom Start entfernt ist. Wie schon bei Reihenfolge 1 erläutert, ist es sinnvoll, die unflexibleren Punkte zuerst abzuarbeiten. Die Flexibilität hängt allerdings nicht nur von der Entfernung vom Start, sondern primär auch von der eigentlichen Höhe der Limits ab.

Es ist zu erhoffen, dass auf diese Weise generierte Einfügereihenfolgen sowohl zu einer erhöhten Erfolgsquote, als auch überdurchschnittlich guten Werten bei den Gesamtkosten führen.

5. **Aufsteigend nach „mittlerer Entfernung zu allen anderen Punkten“ sortiert:** Diese Systematik ist ähnlich zu den beiden vorherigen, da auch hier die Abstände zu anderen Punkten ausschlaggebend für die Sortierung sind. Hier werden allerdings nicht nur der Startpunkt, sondern alle Punkte betrachtet. Dazu wird ein Mittelwert aus den Entfernungen zu allen anderen Punkten gebildet. Dieser wird als $k_\phi(a)$ eines Punktes a bezeichnet und berechnet sich als Mittelwert aller $k(a, b)$ mit: $b=\{0, 1, \dots, a-1, a+1, \dots, m\}$. Der Mittelwert muss bestimmt werden für alle Punkte außer dem Startpunkt, da dieser nicht Teil der Einfügereihenfolge wird. Die errechneten Mittelwerte werden dann in aufsteigender Reihenfolge sortiert und die zugehörigen Punkte in dieser Abfolge zur Einfügereihenfolge P hinzugefügt. Die Einfügereihenfolge wirkt sich so aus, dass zuerst die zentralsten Punkte der Tour hinzugefügt werden und zum Schluss die am weitesten abgelegenen, da diese die weiteste mittlere Entfernung zu anderen Punkten haben. Aufgrund der zusätzlichen Rechenschritte zur Bestimmung des Mittelwertes $k_\phi(a)$ für jeden Punkt a , ist die Ermittlung dieser Einfügereihenfolge rechenaufwendiger als die bisherigen, auch wenn dieser Aufwand durch Einsatz von Software sehr gering ist.
6. **Absteigend nach „mittlerer Entfernung zu allen anderen Punkten“ sortiert:** Erneut ist diese Systematik wieder das Gegenstück zur vorherigen und die resultierende Einfügereihenfolge exakt umgekehrt. Da durch die absteigende Reihenfolge im Verfahren zunächst die dezentralsten Punkte einbezogen werden, verfolgt auch diese Reihenfolge den Ansatz, in den ersten Schritten die grobe Form der Tour festzulegen. Das Argument der stärker reduzierten Puffer bei weiter entfernten Punkten - welches bei Einfügereihenfolge 4 diskutiert wurde - lässt sich hier erneut, wenn auch in geringerem Maße anwenden. Es ist wahrscheinlich, dass die am weitesten vom Start entfernten Punkte gleichermaßen zu den am weitesten von anderen Punkten entfernten gehören.

Es ist hierzu noch anzumerken, dass bei der anfänglichen Wahl von langen Strecken die daraus resultierenden Einfügepositionen klarer voneinander abgegrenzt sind. Somit ist es wahrscheinlich, dass schon in frühen Schritten nicht alle Einfügepositionen gewählt werden können. Generell sollte jedoch eine effiziente Tour auch mehr Flexibilität für spätere Schritte liefern. Daraus folgt, dass zunächst weniger, zum Ende hin jedoch mehr Einfügepositionen zur Wahl stehen sollten.

Generell ist von dieser Einfügereihenfolge zu erhoffen, dass überdurchschnittlich gute Gesamtkosten bei ebenfalls überdurchschnittlicher Erfolgsquote verbucht werden können.

- 7. Größte Entfernung zur bestehenden Tour:** Nach dieser Systematik wird in jedem Schritt jener Punkt der Tour hinzugefügt, welcher am weitesten von der bestehenden Tour entfernt ist. Das heißt der Punkt, dessen minimaler Abstand von einem der Punkte der Tour maximal ist. Dieses Verfahren ist von allen beschriebenen mit Abstand das komplexeste, da für jede Position der Einfügereihenfolge erneut der maximale Minimalabstand zur Tour bestimmt werden muss. Hinzu kommt, dass der Vergleich zu dessen Ermittlung in zwei Schritten durchgeführt werden muss und daher noch aufwendiger ist.

Zunächst wird für jeden noch nicht zur Tour gehörigen Punkt dessen minimaler Abstand zur Tour bestimmt, indem dessen Abstand zu jedem zur Tour gehörigen Punkt verglichen wird. Im zweiten Schritt werden diese Minimalabstände miteinander verglichen und der Punkt ausgewählt, der den maximalen Wert aufweist.

Vom Grundgedanken her wird hier wieder zuerst die grobe Tour festgelegt, indem die äußersten Punkte in den ersten Schritten abgehandelt werden. Auf Kosten des höheren Aufwands wird dies im Vergleich zu Systematik 4 und 6 noch gezielter verfolgt. Insgesamt ist mit sehr ähnlichen Resultaten zu diesen beiden Systematiken zu rechnen.

- 8. Zufällige Reihenfolge:** Wie der Name schon sagt, werden die Punkte hier ungeachtet jeglicher Faktoren in einer zufälligen Reihenfolge zur Tour hinzugefügt.

Die Untersuchung in den nachfolgenden Teilen 5 bis 7 dieser Arbeit soll unter anderem zeigen, wie stark sich die Einfügereihenfolge auf das Ergebnis auswirkt. Die Einfügereihenfolgen 1, 4, 6 und 7 folgen unterschiedlichen Ansätzen, welche in der Theorie eine positive Auswirkung auf das Ergebnis haben sollten. Die Systematiken 2, 3 und 5 dienen hingegen primär als deren Gegenbeispiel, und sollten der gleichen Logik zufolge eher unterdurchschnittliche Ergebnisse erzielen. Die zufällige Reihenfolge stellt für die Auswertung einen Referenzwert dar, an dem gemessen werden kann, in wie weit die Ergebnisse der anderen Ansätze in die eine oder andere Richtung von diesem abweichen.

5. Vorgehensweise der Evaluation

Um, wie in Teil 3 erläutert, eine Bewertung des entwickelten Lösungsverfahrens vornehmen zu können, ist eine Evaluation notwendig. Diese erfordert ihrerseits eine aussagekräftige Datengrundlage sowie Bewertungsfaktoren, nach denen die Daten untersucht werden können. Letztlich müssen jene Einflussfaktoren definiert und ausgewählt werden, welche in der Analyse betrachtet werden sollen.

Hinsichtlich der Bewertungsfaktoren wurden in Teil 4.2 bereits die Erfolgsquote und die Gesamtkosten vorgestellt. Wenn mithilfe einer Methode kein zulässiges Ergebnis erzielt wird, kann es jedoch zusätzlich relevant sein, wie hoch die gesamte Verspätung ist, also wie knapp eine zulässige Lösung verfehlt wurde. So könnte ggf. eine geringe Verspätung an einigen Standorten hingenommen werden, wenn die Gesamtkosten im Gegenzug gering sind. Im Rahmen der Analyse wurde auch versucht solche Werte in Form einer **erweiterten Erfolgsquote** auszudrücken, jedoch waren die Ergebnisse bezüglich dieser äußerst aussageschwach und generell nicht weit von den Ergebnissen der normalen Erfolgsquote entfernt. Daher wird auf die erzielten Ergebnisse in dieser Arbeit nicht weiter eingegangen. Auf die konkrete Bestimmung der Bewertungsfaktoren wird erst in Teil 5.4 eingegangen, da für deren Bestimmung die Struktur der Datengrundlage bekannt sein muss.

5.1 Erstellung der Datengrundlage

Die zentrale Fragestellung der Evaluation ist es, wie oft und mit welchen Gesamtkosten die einzelnen Lösungswege⁵⁶ dazu in der Lage sind, ein gegebenes Problem zu lösen. Daraus folgt, dass als Datengrundlage eine große Anzahl an Problemen und deren Lösungen dienen muss. Um dies herzustellen wurde in *Visual Basic* ein Programm geschrieben, welches zufällige Probleme generiert und diese anschließend mit der beschriebenen Methodik löst. In diesem Teil wird zunächst erklärt, nach welcher Methodik die Probleme generiert werden und welche Rahmenbedingungen für die Analyse einbezogen werden. Die konkrete Implementation von Problemgenerierung und – lösung innerhalb des Programms wird in Teil 8 erklärt.

5.1.1 Methodik der Problemgenerierung

Um die erklärten Lösungsansätze anwenden zu können, werden zunächst lösbare, modifizierte Rundreiseprobleme benötigt. Diese bestehen aus einem Startpunkt und m Standorten, von denen mindestens einer über ein Zeitlimit verfügen muss. Es muss des Weiteren eine vollständige bzw. vervollständigte Kostenmatrix vorliegen, welche die Kosten für den Weg von allen Punkten zueinander enthält.

⁵⁶ Kombination aus Entscheidungskriterium und Einfügereihenfolge

Kostenmatrix:

Theoretisch könnten die einzelnen Werte der Kostenmatrix zwar zufällig generiert werden, jedoch würde das zu sehr realitätsfernen Werten führen und somit wäre auch die Aussagefähigkeit der Testreihe äußerst eingeschränkt. Um die Methoden sinnvoll testen zu können, ist vor allem wichtig, dass die direkte Verbindung zwischen zwei Punkten kürzer ist, als wenn ein Umweg über einen oder mehrere weitere Punkte getätigt wird. Außerdem sollte mit Werten gearbeitet werden, welche in einem sinnvollen Verhältnis zueinander stehen, damit die Ergebnisse eine Aussagefähigkeit für die Realität erhalten.

Als Grundlage der Kostenmatrix dienen daher Standorte, welche in Form von Koordinaten in einer euklidischen Ebene ausgedrückt werden. Die Entfernungen zwischen den Punkten können dann exakt mit dem *Satz des Pythagoras* bestimmt werden. Die Ebene, in der die Punkte liegen, wird mit einer Größe von 100x100 Einheiten vorgegeben. Folglich erhält jede x- und y-Koordinate einen Zufallswert zwischen 0 und 100. Die geometrische Bestimmung der Kostenmatrix hat auch zur Folge, dass ein symmetrisches Problem generiert wird, bei dem eine Verbindung in beiden Richtungen gleich ist. Die Strecke von einem Punkt $a := (x_a | y_a)$ zu einem Punkt $b := (x_b | y_b)$ wird in der Kostenmatrix als $k(a, b)$ angelegt und kann wie folgt errechnet werden:

$$k(a, b) = \sqrt{(x_a - x_b)^2 + (y_a - y_b)^2}$$

Bei den Sonderfällen mit $a=b$, also der Strecke eines Punktes zu sich selbst, werden die Einträge der Kostenmatrix mit dem Wert 9999 blockiert.

5.1.2 Limits:

Aufgrund der Definition des modifizierten Problems muss mindestens einem der Punkte ein Limit zugewiesen werden. Für diese Untersuchung sollen alle Punkte außer dem Startpunkt ein Limit erhalten. Eine angemessene Höhe und Verteilung dieser Limits ist dabei ein bedeutender Faktor, denn die Limits sind maßgebend dafür verantwortlich, wie leicht ein Problem zu lösen ist, bzw. ob es überhaupt eine zulässige Lösung gibt.

Problematisch ist, dass auch nicht ohne weiteres festgestellt werden kann, ob ein Problem überhaupt eine zulässige Lösung besitzt. Das heißt, wenn für ein spezifisches Problem durch keine der getesteten Lösungswege ein zulässiges Ergebnis erzielt werden kann, ist unklar, ob es unmöglich war ein Ergebnis zu erzielen oder ob es nur mit den durchgeführten Lösungswegen unmöglich war.

Wenn ein Problem zulässige Lösungen besitzt, ist es natürlich auch entscheidend, wie viele der $(n-1)!$ Lösungen zulässig sind. Wenn von den $14! = 87178291200$ Lösungen eines Problems mit 15 Standorten 100.000 Lösungen zulässig sind, ist die Erfolgswahrscheinlichkeit natürlich höher, als wenn nur 1.000, 100 oder 10 zulässige Lösungen existieren. Ob zulässige Lösungen existieren, kann ggf. nur durch ein extrem aufwendiges exaktes Lösungsverfahren, wie die vollständige Enumeration bewiesen werden.

Aufgrund der großen Anzahl zu generierender Probleme, ist diese Prüfung nicht ohne weiteres möglich und vom Aufwand her nicht angemessen.

Die Systematik zur automatischen Bestimmung der Limits ist daher äußerst bedeutend und muss so gewählt werden, dass mit einer möglichst hohen Wahrscheinlichkeit ein lösbares, jedoch auch nicht zu leicht lösbares Problem generiert wird. Damit die Limits in der richtigen Höhe generiert werden können, muss deren Generierung in Abhängigkeit des mittleren Abstands der Punkte zueinander geschehen. Dieser ist zwar kein perfekter Indikator, kann jedoch eine grobe Tendenz liefern, welche Limits einhaltbar sind. Dies ist besonders wichtig bei Problemen mit einer geringen Anzahl von Standorten, bei denen der mittlere Abstand stark schwanken kann. Die Aufgabe, eine angemessene Systematik zu finden, wird jedoch noch erheblich dadurch erschwert, dass neben Lage der Orte zueinander, auch deren Anzahl einen erheblichen Einfluss auf die angemessene Höhe von Limits hat.

So nimmt die durchschnittliche Länge einer Einzelstrecke innerhalb der finalen Tour mit zunehmender Anzahl Orte⁵⁷ ab. Das liegt daran, dass bei Erhöhung der Standortanzahl zwar die mittlere Entfernung zwischen zwei Punkten gleich bleibt, sich diese jedoch aus einer größeren Anzahl über- und unterdurchschnittlich langen Strecken zusammensetzt. Somit stehen bei der Lösung von Problemen mit vielen Standorten mehr kurze und effiziente Wege zur Auswahl, was die Wahrscheinlichkeit senkt, dass ineffiziente Wege gewählt werden müssen. Wenn Ineffizienzen dennoch hingenommen werden müssen, fallen diese außerdem im Verhältnis zum längeren Gesamtweg weniger ins Gewicht.

Ignoriert man diesen Effizienzzuwachs bei der Generierung der Limits, so erhält man „große“ Rundreiseprobleme mit zu großzügigen Limits, während „kleine“ Probleme mit erhöhter Wahrscheinlichkeit keine zulässige Lösung haben. In der Konsequenz muss die Standortanzahl also einen Einfluss auf die Limitgenerierung haben.

Als Grundlage für die Generierung der Limits wird der Einfachheit halber von einer bereits sortierten Kostenmatrix ausgegangen. Den Punkten müssen also – beginnend mit dem niedrigsten Limit für Punkt 1 - aufsteigende Limits zugewiesen werden. Die Vorgehensweise ist hier so gewählt, dass zunächst dieses niedrigste sowie das höchste Limit (für Punkt n) bestimmt werden.

Alle weiteren Limits werden dann in gleichmäßigen Abständen zwischen diesen beiden Werten festgelegt. Es wird hier also kein Limit zur Erreichung des Zielpunktes gesetzt. Abbildung 10 zeigt eine beispielhafte Verteilung der Limits zwischen unterstem und oberstem Limit.

⁵⁷ Dabei wird angenommen, dass mehr Orte im gleichen Bereich liegen, also dass die Dichte der Punkte höher ist.

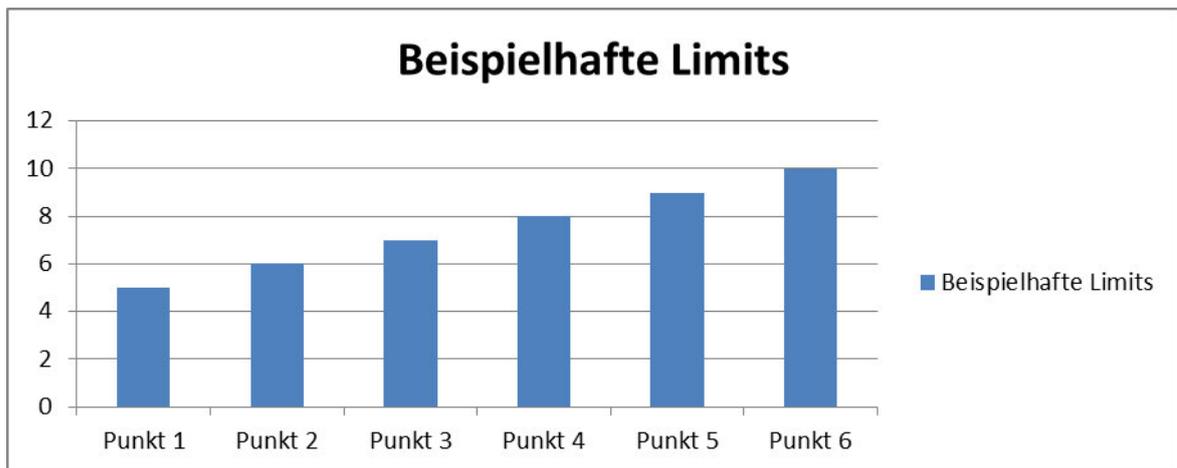


Abbildung 10: Beispielhafte Verteilung der Limits bei Grenzen von 5 und 10 Einheiten

Die Schwierigkeit der Generierung von geeigneten Limits beschränkt sich somit auf die Findung von geeigneten Ober- und Untergrenzen. Um die mittleren Entfernungen der Punkte zueinander einzubeziehen, wird als Berechnungsgrundlage die **Matrixgröße G_m** definiert, welche als Summe der mittleren Entfernungen zu jedem der zu besuchenden Punkt definiert wird. Die Strecken von einem der Punkte zum Startpunkt werden dabei nicht einbezogen, da die Rückfahrt zum Start nicht relevant für die Limits ist.⁵⁸ Die Strecken vom Startpunkt zu einem anderen Punkt hingegen sind relevant und werden einbezogen. Die Matrixgröße lässt sich demnach wie folgt berechnen:

$$G_m = \sum_{a=1}^m \frac{\sum_{b=0}^m k(b, a)}{m}$$

Mit:

G_m =Matrixgröße

$k(b, a)$ = Kosten/ Strecke der Fahrt von Punkt b zu Punkt a

m = Anzahl der Orte ohne Startpunkt

Es ist dabei zu beachten, dass die Strecke eines Punktes zu sich selbst 0 ist. Diese Strecken werden von der Berechnung des Mittelwertes ausgeschlossen. Daher sollte diese Berechnung durchgeführt werden, bevor diese Positionen mit dem Wert 9999 blockiert werden. Ansonsten muss die Formel wie folgt angepasst werden:

$$G_m = \sum_{a=1}^m \frac{[\sum_{b=0}^m k(b, a)] - 9999}{m}$$

⁵⁸ Das liegt daran, dass dem Startpunkt in diesem Fall kein Limit zugeordnet wird.

G_m kann alternativ auch direkt als Mittelwert aller relevanten Kosten multipliziert mit der Anzahl der Orte bestimmt werden:

$$G_m = k_{\emptyset} * m$$

Mit:

k_{\emptyset} = Mittelwert aller Strecken der Kostenmatrix ausgenommen jener zum Startpunkt⁵⁹

Anhand von G_m werden mit verschiedenen Systematiken die Ober- und Untergrenzen der Limits ermittelt:

Limitsystematik 1: Unflexible Limits Basis: Bei unflexibler Limitgenerierung wird der Effizienzzuwachs der Lösung bei zunehmender Standortanzahl nicht berücksichtigt. Ober- und Untergrenze werden anhand der Matrixgröße G_m sowie zweier festgesetzter Faktoren $F=1,45$ und $U=3,3$ wie folgt bestimmt:

Obergrenze:

$$L_{max} = \frac{G_m}{F} = \frac{G_m}{1,45}$$

Untergrenze:

$$L_{min} = G_m * \frac{U}{m} = G_m * \frac{3,3}{m}$$

Die Höhe der Obergrenze gibt vor, wie lang die mittlere Strecke zwischen zwei Punkten der finalen Tour (ausgenommen der Rückfahrt zum Start) maximal sein darf. Diese kalkulierte mittlere Streckenlänge wird als S_{kalk} bezeichnet und lässt sich wie folgt bestimmen:⁶⁰

$$S_{kalk} = \frac{L_{max}}{m}$$

Daraus folgt:

$$S_{kalk} = \frac{G_m}{1,45 * m}$$

Und:

$$S_{kalk} = \frac{k_{\emptyset} * m}{1,45 * m} = \frac{k_{\emptyset}}{1,45}$$

⁵⁹ Für die konkrete Berechnung der Matrixgröße werden die Strecken zum Startpunkt ausgenommen, da die Rückfahrt zum Start nicht für die Limits relevant ist. Für die nachfolgenden Herleitungen wird diese Differenzierung jedoch nicht weiter beachtet, da Einbezug oder Ausschluss der Strecken zum Start einerseits kaum einen Unterschied macht und andererseits im Mittel der gleiche Wert resultiert.

⁶⁰ Die Formel rechnet hier mit m und nicht mit n , da dem Zielort kein Zeitlimit zugewiesen wird.

Demnach sind die kalkulierten Kosten einer Strecke in der finalen Tour nur abhängig von der mittleren Entfernung der Punkte zueinander und nicht von der Anzahl Standorte. Es wird hier also kalkuliert, dass eine Strecke der finalen Tour immer ca. zwei Dritteln von k_{\emptyset} entsprechen darf. Aufgrund der erwähnten Effizienzsteigerung, erhöht sich die Limit-Obergrenze bei zunehmender Standortanzahl schneller als die Gesamtkosten der gebildeten Touren. In der Theorie sollte sich dieser Effekt selbst noch weiter verstärken, da höhere Limits mehr Optionen der Tourengenerierung ermöglichen und folglich auch noch effizientere Touren mit geringeren Gesamtkosten zulassen. Auf diese Problematik wird im Zuge der Limitsystematiken 4, 5 und 6 weiter eingegangen.

(Limitsystematik 2: Unflexible Limits mit vergrößertem Spektrum): Diese Systematik ist eine Variante von Limitsystematik 1 mit modifizierten Faktoren $F=1,25$ und $U=2,75$. Dadurch entsteht ein größerer Abstand zwischen Unter- und Obergrenze. Diese Systematik wird allerdings zugunsten der flexiblen Limitsystematiken nicht weiter verwendet.

(Limitsystematik 3: Unflexible Limits mit reduziertem Spektrum): Wie die vorherige, ist auch diese Systematik eine Variation von der ersten mit den Faktoren $F=1,7$ und $U=4$. Daraus ergibt sich eine verkleinerte Spanne zwischen den Grenzen. Auch diese Systematik wird nicht weiter verwendet.

Limitsystematik 4: Flexible Limits Basis: Aufgrund der Effizienzsteigerung mit zunehmender Anzahl Orte, wird mit diesem Ansatz versucht, die kalkulierten Kosten einer Strecke der finalen Tour S_{kalk} und in der Konsequenz auch die Obergrenze so zu ermitteln, dass diese ebenfalls von der Anzahl der Standorte abhängig ist. Als Ausgangspunkt dient dabei die Formel für S_{kalk} :

$$S_{kalk} = \frac{k_{\emptyset}}{1,45}$$

Da die Abhängigkeit von der mittleren Entfernung k_{\emptyset} erhalten bleiben soll, wird der Faktor $F=1,45$ angepasst, um eine Abhängigkeit von der Standortanzahl herzustellen. Dazu muss zunächst eingeschätzt werden, wie eine solche Abhängigkeit aussehen muss. Dafür wurde unter Verwendung der ersten Limitsystematik und zufällig generierten Orten ein Test mit verschiedenen Standortanzahlen durchgeführt. Untersucht wurden die durchschnittlichen Gesamtkosten von je 100 Messwerten je Standortanzahl unter Verwendung verschiedener Einfügereihenfolgen und Entscheidungskriterien (Kosten, Puffer, Gewichtet). Die resultierenden mittleren Gesamtkosten wurden anschließend durch die Standortanzahl dividiert, wodurch man die mittleren Entfernungen zwischen zwei Punkten der finalen Tour erhält. Diese ergaben in Abhängigkeit vom Lösungsweg sehr variable Ergebnisse.

Grafik 11 zeigt die resultierenden Mittelwerte aller Messwerte in Rot sowie die Ergebnisse des Lösungsweges mit den geringsten Gesamtkosten in Blau.

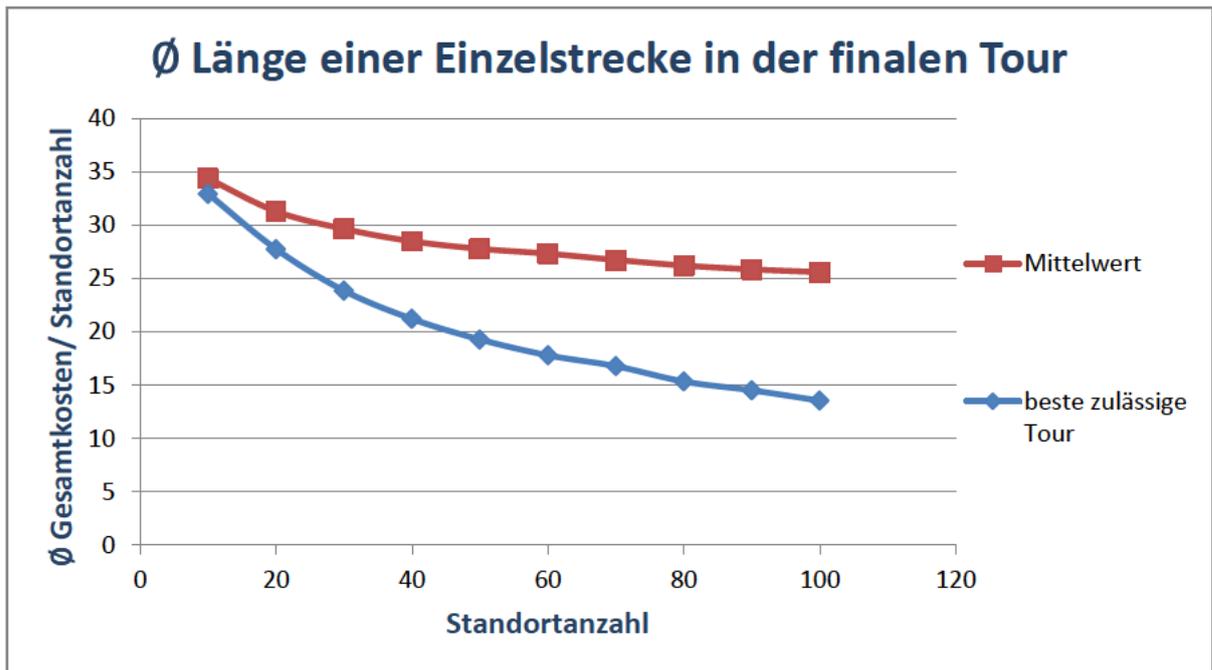


Abbildung 11: Ergebnisse der Testreihe zur Ermittlung der Ø Länge von Einzelstrecken in der finalen Tour

Da sich die Spanne zwischen Mittelwert und Optimum mit zunehmender Standortanzahl immer weiter vergrößert, wurden anhand der Daten Zielwerte für 20 und 100 Orte ermittelt, welche sich am Optimum orientieren, jedoch mehr Spielraum bieten. Tabelle 12 die Zielwerte, welche als s_z bezeichnet werden.

Durchschnittliche Entfernung zweier Punkte in der finalen Tour			
Anzahl Orte	Optimum	Mittelwert	gesetzter Zielwert s_z
20	27,78	31,32	32
100	13,55	25,61	17

Abbildung 12: Gesetzte Zielwerte für die Ø Länge von Einzelstrecken in der finalen Tour

Neben den Gesamtkosten wurde in den Testläufen anhand der Kostenmatrizen auch die mittlere Entfernung zwischen zwei Orten k_\emptyset ermittelt. Da der Startpunkt ebenfalls zufällig ermittelt wurde, ergibt sich der gleiche Wert unabhängig davon, ob der Start einbezogen wird oder nicht. Für diese Entfernung ergab sich gerundet der Wert $k_\emptyset=52$.

Unter Verwendung dieser Werte lassen sich für 20 und 100 Orte neue Faktoren $F(20)$ und $F(100)$ ermitteln, unter deren Verwendung die neuen Zielwerte s_z erreicht werden können:

$$s_z = \frac{k_\emptyset}{F}$$

Daraus folgt:

$$F(20) = \frac{k_\emptyset}{s_{z\ 20}} = \frac{52}{32} = 1,625$$

$$F(100) = \frac{k_\emptyset}{s_{z\ 100}} = \frac{52}{17} = 3,0588$$

Der Zusammenhang zwischen Standortanzahl und Faktor wird hier als linear angenommen. Demnach lässt sich anhand der beiden Werte eine Funktion zur Errechnung von F ermitteln, welche für die Ermittlung flexibler Limits verwendet werden kann:

$$F(m) = a * m + b$$

Steigung und y-Achsenabschnitt lassen sich dann wie folgt errechnen: ⁶¹

$$a = \frac{F(100)-F(20)}{100-20} = \frac{3,0588-1,625}{80} = 0,0179 \approx 0,018$$

$$b = F(20) - a * 20 = 1,625 - 0,018 * 20 = 1,265 \approx 1,3$$

Somit ergibt ein von der Standortanzahl m abhängiger Faktor F_{flex} :

$$F_{flex} = 0,018 * m + 1,3$$

Als neuer Ansatz für die Berechnung der Obergrenze folgt daraus:

$$L_{max} = \frac{G_m}{F_{flex}} = \frac{G_m}{0,018 * m + 1,3}$$

Bei einem ersten Test unter Verwendung der neuen Obergrenze sowie der Untergrenze von Systematik 1 ergaben sich jedoch sehr viele nicht gelöste Ergebnisse. Daher werden die Formeln im nächsten Schritt wie folgt ergänzt:

$$\text{Obergrenze: } L_{max} = G_m * \frac{1,25 + \frac{2}{m}}{1,3 + m * 0,018}$$

Die Matrixgröße wird nun noch zusätzlich mit 1,25 multipliziert, was primär bei Problemen mit großer Standortanzahl (und folglich großer Matrixgröße) zu mehr Flexibilität führen soll. Zusätzlich wird ebenfalls der Faktor 2/m hinzu addiert, welcher besonders bei Problemen mit geringer Standortanzahl einen Unterschied machen sollte.

$$\text{Untergrenze: } L_{min} = G_m * \frac{0,15 + \frac{4}{m}}{1,3 + m * 0,018}$$

Auch die Formel der Untergrenze wird hier flexibel angepasst. Als Ausgangspunkt dient hier jedoch nicht das 1,25-fache der Matrixgröße, sondern das 0,15-fache. Hinzu kommt der flexible Faktor 4/m, welcher erneut besonders den Problemen mit geringer Standortanzahl zu mehr Flexibilität bei der Lösung verhelfen soll. Während die Obergrenze vorgibt, wie lang eine Strecke der finalen Tour maximal sein darf, bestimmt die Höhe der Untergrenze vor allem, wie flexibel besonders die kritischen Punkte innerhalb der finalen Tour untergebracht werden können. Vor allem beeinflusst die Untergrenze auch massiv die Wahrscheinlichkeit mit der ein generiertes Problem lösbar ist. Wenn der Punkt mit dem niedrigsten Limit eine

⁶¹ Da auch die für die Ermittlung verwendeten Zielwerte frei gesetzt worden sind, werden die Ergebnisse hier großzügig gerundet.

größere Entfernung vom Start aufweist, als die Untergrenze erlaubt, so scheidet das Verfahren zwangsläufig an der Erreichung dieses Punktes. Um dies zu vermeiden sind die Untergrenzen hier deutlich großzügiger dimensioniert.

Limitsystematik 5: Flexible Limits mit vergrößertem Spektrum: Diese Systematik ist das flexible Äquivalent zur nicht mehr verwendeten Systematik 2. Während es selbstverständlich ist, dass eine Reduzierung einer oder beider Grenzen zu verringerter Flexibilität und folglich schlechteren Ergebnissen führt, soll mit den Systematiken 5 und 6 untersucht werden, wie sich eine gegensätzliche Veränderung der beiden Grenzen auswirkt.

$$\text{Obergrenze: } L_{max} = G_m * \frac{1,5 + \frac{4}{m}}{1,3 + m * 0,018}$$

$$\text{Untergrenze: } L_{min} = G_m * \frac{0,125 + \frac{3}{m}}{1,3 + m * 0,018}$$

Die Obergrenze wird durch die Anpassung des Faktors zu $(1,5 + 4/m)$ im Vergleich zur Systematik 4 deutlich angehoben, während die Untergrenze abgesenkt wird. Es folgt ein größerer Abstand zwischen den Grenzen und somit auch ein größerer Abstand zwischen den einzelnen Limits. Durch Anhebung der Obergrenze sollten besonders die Punkte mit hohen Limits deutlich flexibler verteilt werden können. Die auch sonst unflexibleren Punkte mit niedrigen Limits sind hier noch deutlich mehr an eine frühe Position in der finalen Tour gebunden.

Limitsystematik 6: Flexible Limits mit reduziertem Spektrum: Limitsystematik 6 bildet mit einer Veränderung beider Grenzen in die entgegengesetzte Richtung das Gegenstück zu Systematik 5. Folglich sind auch die umgekehrten Schlüsse für Flexibilität zu ziehen. Durch die hohe Untergrenze können auch Punkte mit niedrigem Limit deutlich flexibler in der Tour verteilt werden. Die Erzielung einer effizienten Lösung ist hierbei besonders wichtig, da die zulässige Höhe der Gesamtkosten durch die niedrige Obergrenze entschieden knapper ist als bei den andern Systematiken.

$$\text{Obergrenze: } L_{max} = G_m * \frac{1 + \frac{2}{m}}{1,3 + m * 0,018}$$

$$\text{Untergrenze: } L_{min} = G_m * \frac{0,2 + \frac{4,5}{m}}{1,3 + m * 0,018}$$

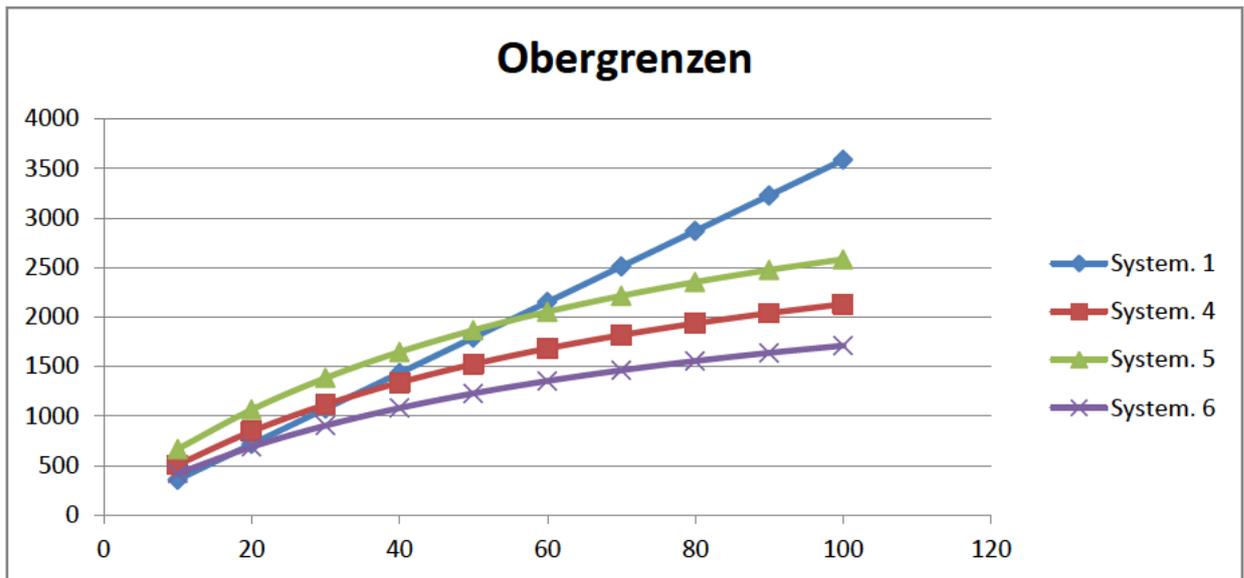


Abbildung 13: Mittlere Höhe der Obergrenzen in Abhängigkeit der Standortanzahl

Grafik 13 zeigt die mittlere Höhe der Obergrenzen, welche mit den vier Systematiken bei 10 bis 100 zufällig verteilten Orten ermittelt werden. Als Berechnungsgrundlage diente wieder der mittlere Abstand zweier zufällig generierter Orte von ca. 52 Einheiten. Die Grafik zeigt, dass die Graphen der flexibel generierten Obergrenzen einen gekrümmten Verlauf aufweisen, welcher an den Verlauf der Graphen aus Abbildung 11 erinnert.

5.2 Untersuchte Einflussfaktoren

In diesem Teil werden die Einflussfaktoren vorgestellt, welche innerhalb der Analyse betrachtet werden sollen. Dazu zählen einerseits die vorgestellten Lösungswege, jedoch auch die für die Generierung der Probleme relevanten Faktoren.

Entscheidungskriterien:

1. Kostenkriterium
2. Pufferkriterium
3. Gewichtetes Kriterium 1:1
4. Gewichtetes Kriterium 1:2 (Puffer 1: Kosten 2)
5. Gewichtetes Kriterium 1:4 (Puffer 1: Kosten 4)
6. Gewichtetes Kriterium 1:8 (Puffer 1: Kosten 8)

Einfügereihenfolgen:

1. **Aufsteigend nach Höhe des Limits sortiert**
2. **Absteigend nach Höhe des Limits sortiert**
3. **Aufsteigend nach „Entfernung vom Startpunkt“ sortiert**
4. **Absteigend nach „Entfernung vom Startpunkt“ sortiert**
5. **Aufsteigend nach „mittlerer Entfernung zu allen anderen Punkten“ sortiert**
6. **Absteigend nach „mittlerer Entfernung zu allen anderen Punkten“ sortiert**
7. **Größte Entfernung zur bestehenden Tour**
8. **Zufällige Reihenfolge**

Anzahl der Standorte: Die Standortanzahl ist ein signifikanter Faktor besonders für den Lösungsaufwand, jedoch kann sie auch einen erheblichen Einfluss auf die Dynamik des Lösungsverfahrens haben. So fallen zum Beispiel einzelne Ineffizienzen bei einem Problem mit vielen Standorten deutlich weniger ins Gewicht als bei einem kleinen Problem. Für die Messreihen werden folgende Ausprägungen gewählt:

M=20/ n=21: Ein Startpunkt plus 20 zu besuchende Punkte

M=50/ n=51: Ein Startpunkt plus 50 zu besuchende Punkte

M=100/ n=101: Ein Startpunkt plus 100 zu besuchende Punkte

Höhe der Limits: Untersucht werden hierbei die vier in Teil 5.1.2 vorgestellten Systematiken zur Ermittlung von Ober- und Untergrenze. Konkret stellt sich die Frage, in wie weit Erfolgsquote und Gesamtkosten von diesen beeinflusst werden:

1. **Limitsystematik 1: Unflexible Limits Basis**
2. **Limitsystematik 4: Flexible Limits Basis**
3. **Limitsystematik 5: Flexible Limits mit vergrößertem Spektrum**
4. **Limitsystematik 6: Flexible Limits mit reduziertem Spektrum**

Verteilung der Standorte: Die Standorte werden in Form von Koordinaten in einer euklidischen Ebene dargestellt. Je nach Systematik der Generierung können die Koordinaten auf unterschiedliche Weise in der Ebene verteilt liegen. Betrachtet werden vier verschiedene Varianten der Standortverteilung. Zu diesen wird jeweils eine beispielhafte Verteilung von 100 Orten grafisch dargestellt:

1. **Zufällige Verteilung:** Die Koordinaten der Punkte werden hier komplett zufällig in einer 100x100 Einheiten großen Ebene verteilt. Bestimmt werden diese, indem für jeden Punkt eine x und eine y-Koordinate als Zufallszahl im Bereich von 0-100 generiert werden.

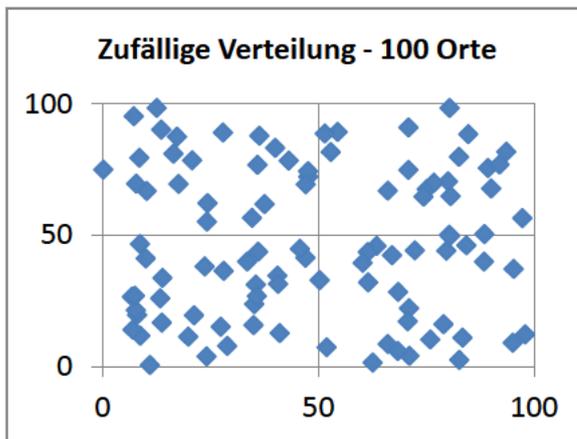


Abbildung 14: Beispielhafte zufällige Verteilung der Orte

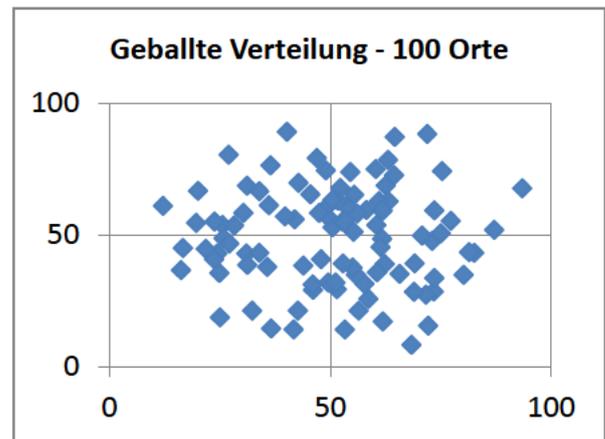


Abbildung 15: Beispielhafte Verteilung mit zentraler Ballung

2. **Zentrale Ballung:** Diese Verteilung favorisiert die Generierung von Punkten nahe der Mitte einer 100x100 Einheiten großen Ebene. Dies wird erreicht, indem die Koordinaten als Summe von drei Zufallszahlen bestimmt werden: Zwei Zufallszahlen von 0-40 und eine von 0-20. Dadurch werden die Extremwerte erheblich unwahrscheinlicher und Werte im Bereich von 50 deutlich wahrscheinlicher. Diese Art der Verteilung soll im Groben eine Stadt simulieren, bei der im Zentrum die Bevölkerungsdichte – hier die Dichte der Punkte - hoch ist und in den Randgebieten abnimmt.
3. **Zufällige Verteilung im gestreckten Bereich:** In diesem Fall hat die Ebene die Dimensionen 50x200. Das bedeutet, die x-Koordinaten werden als Zufallszahl zwischen 0 und 50 generiert, die y-Koordinaten als Zufallszahl zwischen 0 und 200.

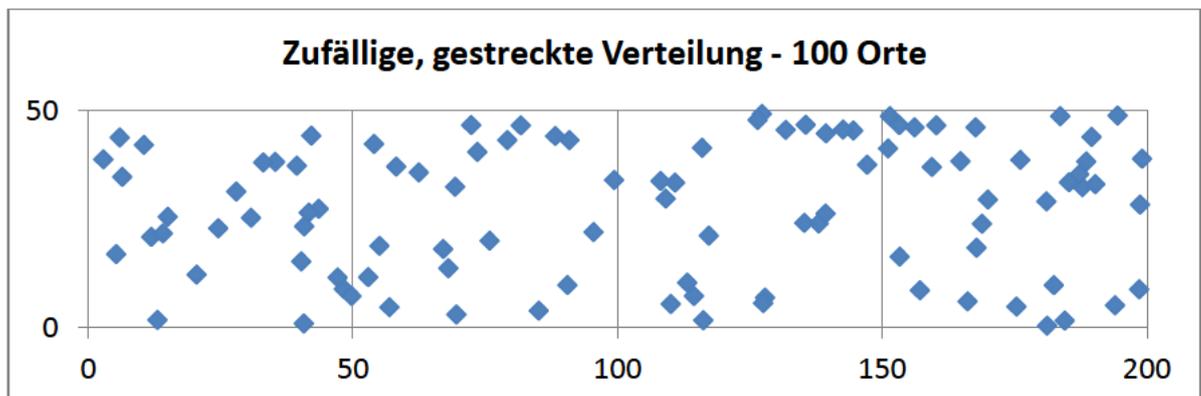


Abbildung 16: Beispielhafte Zufällige Verteilung der Orte im gestreckten Raum (50x200 Einheiten)

4. **Städtecluster:** Die Generierung nach Städteclustern ist mit Abstand am kompliziertesten. Im Gegensatz zur zentralen Ballung soll hierdurch nicht nur der Aufbau einer Stadt, sondern ein ganzer Landstrich mit mehreren Städten simuliert werden. Die zu besuchenden Punkte – zum Beispiel Kunden – befinden sich in einer der Städte, während die Regionen zwischen den Städten kaum oder keine Punkte aufweist. Das heißt die Punkte sind in Clustern über eine 100x100 Ebene verteilt. Diese werden so generiert, dass zunächst eine beliebige Anzahl von Stadtzentren in Form von zufälligen Koordinaten über die Ebene verteilt wird. Für diese Untersuchung wird immer von 10 Städten ausgegangen. Um eine zu starke Ballung an einem Ort zu vermeiden, werden diese gleichmäßig auf alle Quadranten⁶² aufgeteilt.
- Die eigentlichen Punkte werden dann zunächst zufällig einer der Städte zugeordnet und anschließend in einem Bereich von 16 Einheiten um das Stadtzentrum herum – auch hier mit zentraler Ballung – generiert.
- Die Koordinaten der Städte werden ermittelt als Zufallszahl zwischen 8 und 92 (damit die eigentlichen Punkte die Werte 0 und 100 nicht über- bzw. unterschreiten).
- Die Koordinaten der Punkte ermitteln sich aus der Koordinate der ausgewählten Stadt addiert mit der Summe zweier Zufallszahlen zwischen -4 und 4. Ein Punkt kann somit in beiden Dimensionen maximal 8 Einheiten vom zugehörigen „Stadtzentrum“ entfernt liegen.

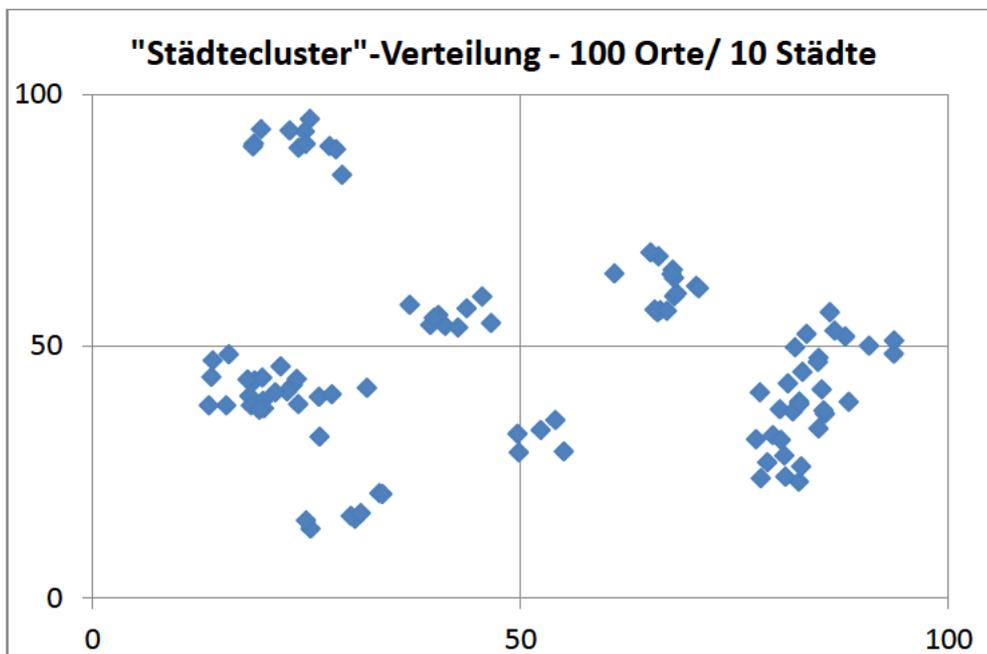


Abbildung 17: Beispielhafte Verteilung der Orte bei Generierung in „Städteclustern“

⁶² Hiermit ist jeweils ein Viertel der Ebene, also ein 50x50 Einheiten großer Bereich gemeint.

Lage des Startpunktes: Im geringeren Maße wie die Verteilung der Standorte kann auch die relative Lage des Startpunktes einen Einfluss auf die Lösungen haben. Dies betrifft besonders die Einfügereihenfolgen, welche in Abhängigkeit von der Entfernung vom Startpunkt generiert werden. Demnach soll untersucht werden, in wie weit sich Unterschiede bemerkbar machen. Da sich der Einfluss dieses Faktors in Problemen mit vielen Standorten mit großer Wahrscheinlichkeit sehr gering auswirkt, wird die Lage des Startpunktes nur bei relativ kleinen Problemen mit 20 Standorten untersucht.

1. **Zufällige Lage:** Der Startpunkt wird wie die restlichen Punkte zufällig nach einem der oben genannten Verfahren bestimmt.
2. **Zentrale Lage:** Der Startpunkt liegt zentral in der Euklidischen Ebene. Für die Standortverteilungen 1, 3 und 4 erhält er also die Koordinaten $x_0=50$; $y_0=50$. In der gestreckten Ebene wäre dies angepasst zu $x_0=25$; $y_0=100$.
3. **Dezentrale Lage:** Eine der Ecken der Ebene wird für die Lage des Startpunktes ausgewählt. Er erhält die Koordinaten $x_0=0$; $y_0=0$.

5.2.1 Zusammenfassung der zu untersuchenden Faktoren

Zusammengefasst sollen in der Analyse folgende Varianten und Einflussfaktoren untersucht werden:

- 6 Varianten des Entscheidungskriteriums (Kostenkriterium, Pufferkriterium, verschiedene Gewichtungen)
- 8 Einfügereihenfolgen
- 3 Ausprägungen der Standortanzahl
- 4 Arten der Limitgenerierung
- 4 Arten der Standortverteilung
- 3 Möglichkeiten der Lage des Startpunktes

Ein Test von allen Varianten unter sämtlichen Bedingungen würde daher eine Gesamtanzahl von $6 \times 8 \times 3 \times 4 \times 4 \times 3 = 6912$ Kombinationen bedeuten. Da dies den Rahmen dieser Arbeit sprengt, muss diese Auswahl deutlich eingeschränkt werden. Zunächst ist bei den Faktoren entscheidend, ob durch sie die Generierung der Kostenmatrix beeinflusst wird oder nicht. Die Varianten des Lösungsverfahrens, die Einfügereihenfolgen sowie die Generierung der Limits bewirken dies nicht. Demnach können bei einer einzigen generierten Kostenmatrix eine beliebige Anzahl von Kombinationen dieser Faktoren nacheinander durchgerechnet werden. Eine Kombination aus Entscheidungskriterium und der Einfügereihenfolge wird als **Lösungsweg** bezeichnet. Die Ergebnisse von verschiedenen Lösungswegen bei gleicher Kostenmatrix und gleichen Limits werden von nun an als ein **Datensatz** bezeichnet. Für eine Kostenmatrix können mehrere Datensätze unter Verwendung verschiedener Arten der Limitgenerierung ermittelt werden. Eine solche Gruppe aus Datensätzen wird hier als ein **Durchlauf** bezeichnet. Die Faktoren, welche die Generierung der Kostenmatrix beeinflussen, werden **Rahmenbedingungen** genannt. Eine Reihe aus Durchläufen mit gleichen Rahmenbedingungen wird als eine **Messreihe** bezeichnet.

Die Limits sind eigentlich auch eine Rahmenbedingung, jedoch beeinflussen sie die Generierung der Kostenmatrix nicht, wodurch mehrere Datensätze mit unterschiedlichen Limits nacheinander getestet werden können ohne die Kostenmatrix neu generieren zu müssen.

Zusammenfassend müssen zunächst die Rahmenbedingungen für eine Messreihe festgelegt werden, also die Anzahl der Standorte, deren Verteilung und die Lage des Startpunktes. Anschließend muss festgelegt werden, aus wie vielen Durchläufen die Messreihe bestehen soll und welche Limits untersucht werden. Zuletzt müssen die **Lösungswege** definiert werden, aus welchen sich ein Datensatz zusammensetzt.

Zur Übersichtlichkeit werden die Rahmenbedingungen einer Messreihe a wie folgt abgekürzt:

Messreihe a: (Anzahl der Punkte = m; Verteilung der Punkte; Lage des Startpunktes)

Beispiel: Messreihe 1: (20; Städtecluster; zufällig)

Alle Messreihen werden dimensioniert mit 500 Durchläufen, in welchen mehrere Lösungswege unter verschiedenen Limitsystematiken berechnet werden.

5.3 Fokus der Analyse

Zur Begrenzung der Datenmenge wird die Analyse in zwei Schritte unterteilt. Im ersten Analyseteil werden zunächst gezielt die Einfügereihenfolgen untersucht, um festzustellen, ob und in wie weit die unterschiedlichen Systematiken überhaupt signifikante Unterschiede bei den Bewertungsfaktoren aufweisen. Es werden daher zunächst nur die einfachsten Entscheidungskriterien und nur wenige Rahmenbedingungen untersucht.

Im zweiten Analyseteil werden dann nur die besseren Einfügereihenfolgen in Form verschiedener Lösungswege gegenüber gestellt und deren Leistung unter unterschiedlichen Rahmenbedingungen (auch Limits) verglichen.

5.4 Bewertungsfaktoren der Analyse

Wie bereits zu Beginn dieses Teils erläutert, sind die entscheidenden Bewertungsfaktoren die Erfolgsquote und die Höhe der Gesamtkosten. Um beide Faktoren bewerten zu können, müssen die verschiedenen Lösungswege eines Datensatzes zueinander in Relation betrachtet werden. Im Folgenden wird erklärt, wie Erfolgsquote und Gesamtkosten errechnet werden.

5.4.1 Erfolgsquote:

Die Erfolgsquote ist der wichtigste Bewertungsfaktor, da der Erhalt einer zulässigen Lösung das übergeordnete Ziel der Methode darstellt. Die Erfolgsquote eines Lösungsweges lässt sich nach Durchführung einer Messreihe wie folgt berechnen:

$$\text{Erfolgsquote} = \frac{\text{Anzahl zulässig gelöster Probleme}}{\text{Anzahl lösbarer Probleme}}$$

Dabei ist es wichtig zu beachten, dass die Anzahl lösbarer Probleme nicht unbedingt der Anzahl an Durchläufen entspricht. Wie in Teil 5.1.2 erläutert, kann nicht ohne weiteres bestimmt werden, ob ein automatisch generiertes Problem lösbar ist oder nicht. Daher wird vereinfacht angenommen, dass ein Problem nur dann lösbar ist, wenn es mit mindestens einem Lösungsweg des Datensatzes erfolgreich gelöst werden konnte.

Beispiel: Konnten mit einem Lösungsweg 300 von 500 Problemen gelöst werden, während 100 Probleme von keinem der Lösungswege gelöst werden konnten, so ergibt sich für die Erfolgsquote des Lösungsweges: $\frac{300}{500-100} = 75\%$.

5.4.2 Gesamtkosten:

Bei den Gesamtkosten ist es besonders wichtig, diese in Relation zu den anderen Ergebnissen zu betrachten, da ein absoluter Wert ohne Referenz keine Aussage hat. Außerdem sind dabei nur zulässige Ergebnisse relevant, da unzulässige Ergebnisse erheblich niedrigere Gesamtkosten aufweisen können, welche das Bild verfälschen würden. Daher wird hier so vorgegangen, dass aus jedem **Datensatz** das beste zulässige Ergebnis K_{opt} ermittelt wird. Alle zulässigen Ergebnisse dieses Datensatzes werden dann prozentual ins Verhältnis zu diesem „Optimum“ gesetzt. Aus dem Mittelwert der prozentualen Ergebnisse errechnen sich schließlich die mittleren Gesamtkosten K_{\emptyset} eines Lösungsweges in einer Messreihe. Die Berechnung soll am nachfolgenden Beispiel verdeutlicht werden:

	Absolute Gesamtkosten				Erfolgsquote
	Datensatz 1	Datensatz 2	Datensatz 3	Datensatz 4	
Lösungsweg A	10	13	10	12	75%
Lösungsweg B	14	14	12	12	75%
Lösungsweg C	11	10	13	11	25%
Lösungsweg D	13	12	15	11	100%
Optimum*	10	12	10	11	

Abbildung 18: Beispielhafte absolute Gesamtkosten, welche durch verschiedene Lösungswege ermittelt wurden⁶³

⁶³ *Die rot hinterlegten Werte sollen nicht zulässige Ergebnisse darstellen. Für die Bildung des Optimums sind nur zulässige Werte relevant.

	Gesamtkosten im Verhältnis zum Spaltenoptimum				Mittelwert K_{ϕ}
	Datensatz 1	Datensatz 2	Datensatz 3	Datensatz 4	
Lösungsweg A	100%	108%	100%	-	103%
Lösungsweg B	140%	117%	-	109%	122%
Lösungsweg C	-	-	130%	-	130%
Lösungsweg D	130%	100%	150%	100%	120%

Abbildung 19: Darstellung der Gesamtkosten in Relation zum Optimum des jeweiligen Datensatzes

Die erste Tabelle zeigt die Ergebnisse einer Messreihe mit vier Datensätzen. Das heißt, eine Spalte repräsentiert ein Problem (aus Kostenmatrix und Limits), welches durch die vier Lösungswege unterschiedlich gelöst wurde. Bei den rot hinterlegten wird eine unzulässige Lösung angenommen. In der untersten Zeile sind die jeweils besten zulässigen Gesamtkosten aufgeführt. Die zweite Tabelle zeigt, wie die Daten für die Bestimmung der mittleren Gesamtkosten aufgearbeitet wurden, indem die absoluten Werte durchs Spaltenoptimum geteilt wurden. Die letzte Spalte enthält den Mittelwert der Zeilen und somit die für die Analyse verwendeten mittleren Gesamtkosten K_{ϕ} .

6. Erster Teil der Analyse

6.1 Messreihen im ersten Teil

Für die erste Analyse werden nur drei Messreihen durchgeführt. Alle verwenden eine zufällige Lage der Punkte, inklusive Startpunkt. Hinsichtlich der Standortanzahl werden die drei Ausprägungen mit 20, 50 und 100 Standorten untersucht:

Messreihe 1.01: (20; zufällig; zufällig)

Messreihe 1.02: (50; zufällig; zufällig)

Messreihe 1.03: (100; zufällig; zufällig)

In einer Messreihe werden hier die folgenden Lösungswege und Limits untersucht:

Lösungsverfahren: Kostenkriterium, Pufferkriterium, Gewichtetes Kriterium 1:1

Einfügereihenfolgen: alle

Limits: Systematiken 1, 4, 5, 6

Das ergibt $3 \times 8 = 24$ verschiedene Lösungswege, und somit 96 Ergebnisse, welche in einem Durchlauf berechnet werden. Die Messreihen bestehen aus je 500 Durchläufen und liefern somit jeweils fast 50.000 Messwerte.

6.2 Auswertung erster Teil

Der Fokus des ersten Teils der Auswertung liegt auf den Unterschieden zwischen den 24 Lösungswegen. Da jede Messreihe mit vier verschiedenen Arten der Limitgenerierung und je 500 Durchläufen durchgeführt wurde, liegen für jeden Lösungsweg 6000 Ergebnisse (2000 je Messreihe und Standortanzahl) vor, auf deren Basis eine Bewertung vorgenommen werden kann. Die Tabellen 20 und 21 stellen die Ergebnisse der drei Messreihen dar. Die verschiedenen Arten der Limitgenerierung werden erst im zweiten Teil der Auswertung untersucht. In diesem Teil werden die Mittelwerte aus den Datensätzen der verschiedenen Limits verwendet um auf eine größere Datenbasis zurückgreifen zu können.

6.2.1 Auswertung der Erfolgsquoten

Kriterium	Einfügereihenfolge	Erfolgsquote			Erfolgsquote
		20 Orte	50 Orte	100 Orte	Mittelwert
Kosten	1. Limit min-max	91%	94%	80%	88%
	2. Limit max-min	28%	4%	0%	11%
	3. Entf. zu 0 min-max	30%	1%	0%	10%
	4. Entf. zu 0 max-min	48%	8%	0%	19%
	5. ∅ Entf. min-max	34%	3%	0%	12%
	6. ∅ Entf. max-min	41%	7%	0%	16%
	7. max Entf. zur Tour	39%	5%	0%	15%
	8. Zufallsreihenfolge	40%	4%	0%	15%
Puffer	1. Limit min-max	89%	83%	34%	69%
	2. Limit max-min	83%	70%	43%	65%
	3. Entf. zu 0 min-max	74%	46%	9%	43%
	4. Entf. zu 0 max-min	92%	81%	31%	68%
	5. ∅ Entf. min-max	73%	40%	8%	40%
	6. ∅ Entf. max-min	94%	95%	60%	83%
	7. max Entf. zur Tour	92%	91%	65%	83%
	8. Zufallsreihenfolge	87%	85%	47%	73%
Gewichtet 1:1	1. Limit min-max	97%	98%	89%	95%
	2. Limit max-min	62%	42%	19%	41%
	3. Entf. zu 0 min-max	80%	73%	34%	62%
	4. Entf. zu 0 max-min	91%	86%	55%	78%
	5. ∅ Entf. min-max	81%	71%	31%	61%
	6. ∅ Entf. max-min	90%	82%	62%	78%
	7. max Entf. zur Tour	90%	89%	82%	87%
	8. Zufallsreihenfolge	86%	86%	70%	81%

Abbildung 20: Erfolgsquoten der Lösungswege im ersten Analyseteil

Die Erfolgsquoten der verschiedenen Lösungswege weisen erhebliche Unterschiede auf, welche sich sowohl auf das Entscheidungskriterium, als auch auf die verwendete Einfügereihenfolge zurückführen lassen. Die größte Auffälligkeit ist, dass die auf dem Kostenkriterium basierenden Lösungswege extrem schlechte Erfolgsquoten aufweisen, mit

Ausnahme der Einfügereihenfolge nach aufsteigenden Limits, welche sehr gute Erfolgsaussichten bietet. Des Weiteren fällt auf, dass mit zunehmender Anzahl Orte die Erfolgsquote abnimmt. Da die Erfolgsquote an der Anzahl „lösbarer Probleme“⁶⁴ gemessen wird, bedeutet das, dass es mehr Probleme gab, welche nur von wenigen der Verfahren gelöst werden konnten. Mit dem Kostenkriterium und den Einfügereihenfolgen 2-8 konnte bei 100 Orten nur eine einzige zulässige Lösung (mit der Zufallsreihenfolge) erzielt werden. Die schlechten Ergebnisse lassen sich dadurch erklären, dass die Methodik des Kostenkriteriums das Einfügen von Punkten mit sehr geringem Zeitpuffer zulässt. Wird ein Punkt so weit hinten in die bestehende Tour eingefügt, dass das Zeitlimit gerade so eingehalten wird, so sind alle davor liegenden Einfügepositionen effektiv blockiert.⁶⁵ Wenn in einem späteren Schritt ein weiterer Punkt hinzugefügt werden muss, dessen Limit noch niedriger ist – für den also nur die blockierten Einfügepositionen in Frage kommen - so scheitert das Verfahren. Je mehr Punkte besucht werden müssen, desto wahrscheinlicher ist es, dass sich das Verfahren auf diese Weise selbst blockiert. Diese Problematik kann weitgehend vermieden werden, wenn verhindert wird, dass ein Punkt mit niedrigerem Limit nach einem Punkt mit höherem Limit eingefügt wird. Demnach führt nur die nach aufsteigenden Limits sortierte Einfügereihenfolge zu guten Ergebnissen. Nur mit dem gewichteten Kriterium wird unter Verwendung dieser Einfügereihenfolge mit 95% im Mittel eine noch bessere Erfolgsquote erzielt.

Pufferkriterium und gewichtetes Kriterium liefern ähnliche Erfolgsquoten mit den meisten Einfügereihenfolgen. Die Reihenfolgen 2, 3 und 5, welche - wie in Teil 4.3 erläutert - als Gegenbeispiel dienen sollen, liefern deutlich schlechtere Ergebnisse als ihre jeweiligen Counterparts. Dabei ist jedoch auch anzumerken, dass sich auch diese von der Erfolgsquote her nicht stark (wenn überhaupt) von der zufälligen Reihenfolge abheben können. Allgemein werden durch Anwendung von Puffer- und gewichtetem Kriterium gute Ergebnisse erzielt. Dies gilt besonders für die Einfügereihenfolgen 6. (\emptyset Entfernung zu anderen Punkten absteigend sortiert) und 7. (maximale Entfernung zur Tour) sowie in etwas geringerem Maße für Reihenfolge 4. (Entfernung vom Startpunkt absteigend sortiert). Das gewichtete Kriterium hat besonders große Schwierigkeiten im Zusammenspiel mit der Sortierung nach absteigenden Limits, was sich ebenfalls auf die Problematik des Kostenkriteriums zurückführen lässt. Dadurch, dass hier jedoch auch die Puffer ins Gewicht fallen, ist das Ergebnis immer noch deutlich besser als beim Kostenkriterium.

Letztlich fällt auf, dass die Leistungsfähigkeit des Pufferkriteriums bei 100 Orten erheblich abfällt. Dessen beste Erfolgsquote ist mit 65% nur noch mittelmäßig. Generell können nur drei Lösungswege mit einer Erfolgsquote von über 80% überzeugen: Kostenkriterium und gewichtetes Kriterium mit aufsteigenden Limits und gewichtetes Kriterium mit der max. Entfernung zur Tour. Wie und ob sich der Trend der sinkenden Erfolgsquoten bei einer

⁶⁴ Wie in Teil 5.4 erläutert, sind das Probleme, welche von mindestens einem Lösungsverfahren gelöst worden sind.

⁶⁵ Weil ein zusätzlicher Umweg vor Erreichung des kritischen Punktes dessen knappen Restpuffer überschreiten würde.

weiteren Erhöhung der Standortanzahl fortsetzen würde, ist schwer zu sagen, da dies im Wesentlichen auch von den Limits und der Effizienz der Strecken⁶⁶ abhängt, deren Zusammenhang nicht als symmetrisch angenommen werden kann. Es ist allerdings damit zu rechnen, dass die Leistungsfähigkeit der Lösungswege, welche bereits bei 100 Orten ein schwaches Ergebnis erzielt haben, weiter hinter die besseren Wege zurückfallen wird.

Zusammenfassend kann das Ergebnis formuliert werden, dass die Einfügereihenfolge einen erheblichen Einfluss auf den Erfolg des Verfahrens haben kann. Dabei bietet jedoch die Wahl einer „guten“ Einfügereihenfolge (1, 4, 6, 7) einen verhältnismäßig geringen Mehrwert im Vergleich zur zufälligen Reihenfolge. Die Wahl einer „schlechten“ Reihenfolge (2, 3, 5) kann jedoch zu erheblich schlechteren Ergebnissen führen und ist daher zu vermeiden. Besonders hervorzuheben ist dabei das Kostenkriterium, welches nur mit einer einzigen Einfügereihenfolge kompatibel ist.

6.2.2 Auswertung der Gesamtkosten

Bei der Betrachtung der Gesamtkosten (siehe Abbildung 21) fällt besonders auf, dass es sehr große Unterschiede zwischen den drei Entscheidungskriterien gibt. So bewegen sich die durchschnittlichen Gesamtkosten unter Verwendung des Kostenkriteriums zwischen 100 und 120%. Gesamtkosten von 100% sind dabei besonders hervorzuheben, da sie bedeuten, dass jedes Mal, wenn ein zulässiges Ergebnis erzielt wurde, dieses Ergebnis nicht von einem anderen Lösungsweg übertroffen werden konnte. Es ist des Weiteren davon auszugehen, dass bei 20 und 50 Orten nahezu immer ein auf dem Kostenkriterium basierendes Lösungsverfahren das kostentechnisch beste Ergebnis geliefert hat. Da nur zulässige Ergebnisse bewertet werden, kann bei 100 Orten kein Urteil über die Einfügereihenfolgen 2-8 in Verbindung mit dem Kostenkriterium getroffen werden, weil diese kein (bzw. nur ein einziges) zulässiges Ergebnis ermitteln konnten.

Die mit dem Pufferkriterium errechneten, zulässigen Lösungen liegen im Durchschnitt 70-90% über dem besten erzielten Wert. Die Kosten sind also im Schnitt fast doppelt so hoch wie notwendig. Wie anhand dieser beiden Extreme zu erwarten ist, liegen die Ergebnisse des gewichteten Kriteriums im Mittelfeld. Die Werte liegen zwischen 30 und 60% oberhalb des Optimums und somit geringfügig näher an den Ergebnissen des Pufferkriteriums.

Im Gegensatz zur Erfolgsquote haben die unterschiedlichen Einfügereihenfolgen nur einen relativ geringen Einfluss auf die Gesamtkosten. Die „guten“, entfernungsbasierten Einfügereihenfolgen (4, 6, 7) können Gesamtkosten erzielen, welche nur um etwa 5% besser sind als die der Zufallsreihenfolge und ca. 10 % besser als die entgegengesetzten Einfügereihenfolgen (3 und 5).

⁶⁶ Mit der Effizienz der Einzelstrecken ist hier gemeint, wie lang durchschnittlich eine Strecke der vollendeten Tour ist. Diese nimmt mit zunehmender Standortanzahl ab.

Kriterium	Einfügereihenfolge	Ø Gesamtkosten			Ø Gesamtkosten
		20	50	100	Mittelwert
Kosten	1. Limit min-max	119%	107%	100%	108,8%
	2. Limit max-min	104%	102%	Kein Ergebnis	103,2%
	3. Entf. zu 0 min-max	119%	108%	Kein Ergebnis	114,2%
	4. Entf. zu 0 max-min	104%	102%	Kein Ergebnis	102,9%
	5. Ø Entf. min-max	117%	111%	Kein Ergebnis	113,9%
	6. Ø Entf. max-min	104%	102%	Kein Ergebnis	103,1%
	7. max Entf. zur Tour	105%	105%	Kein Ergebnis	105,1%
	8. Zufallsreihenfolge	110%	105%	100%	106,7%
Puffer	1. Limit min-max	174%	184%	175%	177,7%
	2. Limit max-min	168%	171%	161%	166,5%
	3. Entf. zu 0 min-max	181%	195%	186%	187,3%
	4. Entf. zu 0 max-min	172%	183%	175%	176,8%
	5. Ø Entf. min-max	183%	196%	187%	188,8%
	6. Ø Entf. max-min	169%	178%	170%	172,1%
	7. max Entf. zur Tour	171%	180%	170%	173,6%
	8. Zufallsreihenfolge	174%	183%	173%	176,6%
Gewichtet 1:1	1. Limit min-max	150%	160%	154%	154,5%
	2. Limit max-min	136%	133%	127%	131,9%
	3. Entf. zu 0 min-max	151%	163%	155%	156,5%
	4. Entf. zu 0 max-min	142%	151%	147%	146,6%
	5. Ø Entf. min-max	152%	164%	159%	158,3%
	6. Ø Entf. max-min	141%	151%	147%	146,3%
	7. max Entf. zur Tour	145%	157%	152%	151,5%
	8. Zufallsreihenfolge	147%	158%	152%	152,2%

Abbildung 21: Ø Gesamtkosten der Lösungswege im ersten Analyseteil

Die im Schnitt besten Ergebnisse liefert überraschenderweise die Einfügereihenfolge nach absteigenden Limits und somit keine der Reihenfolgen, deren Methodik, wie in Teil 4.3 erläutert, bessere Gesamtkosten versprochen.

6.2.3 Betrachtung beider Bewertungsfaktoren in Relation:

Grafik 22 stellt für jeden Lösungsweg die Gesamtkosten und Erfolgsquoten in einem Balkendiagramm dar. Der ideale Wert beider Faktoren liegt bei 100%. Der einzige Lösungsweg, welcher für beide ein gutes Ergebnis erzielt ist die Kombination aus Kostenkriterium mit aufsteigenden Limits. Das Pufferkriterium weist tendenziell vergleichbare oder leicht schlechtere Erfolgsquoten auf als das gewichtete Kriterium, hat jedoch überall deutlich höhere Gesamtkosten. Das gewichtete Kriterium schafft es, gute Erfolgsquoten mit vergleichsweise niedrigen Gesamtkosten zu verbinden. Im zweiten Teil der Auswertung gilt es daher, unterschiedliche Gewichtungen von Kosten zu Puffern zu testen. Die zentrale Fragestellung dabei ist, in wie weit es möglich ist, die Gesamtkosten zu senken, ohne dabei Einbußen bei der Erfolgsquote hinnehmen zu müssen.

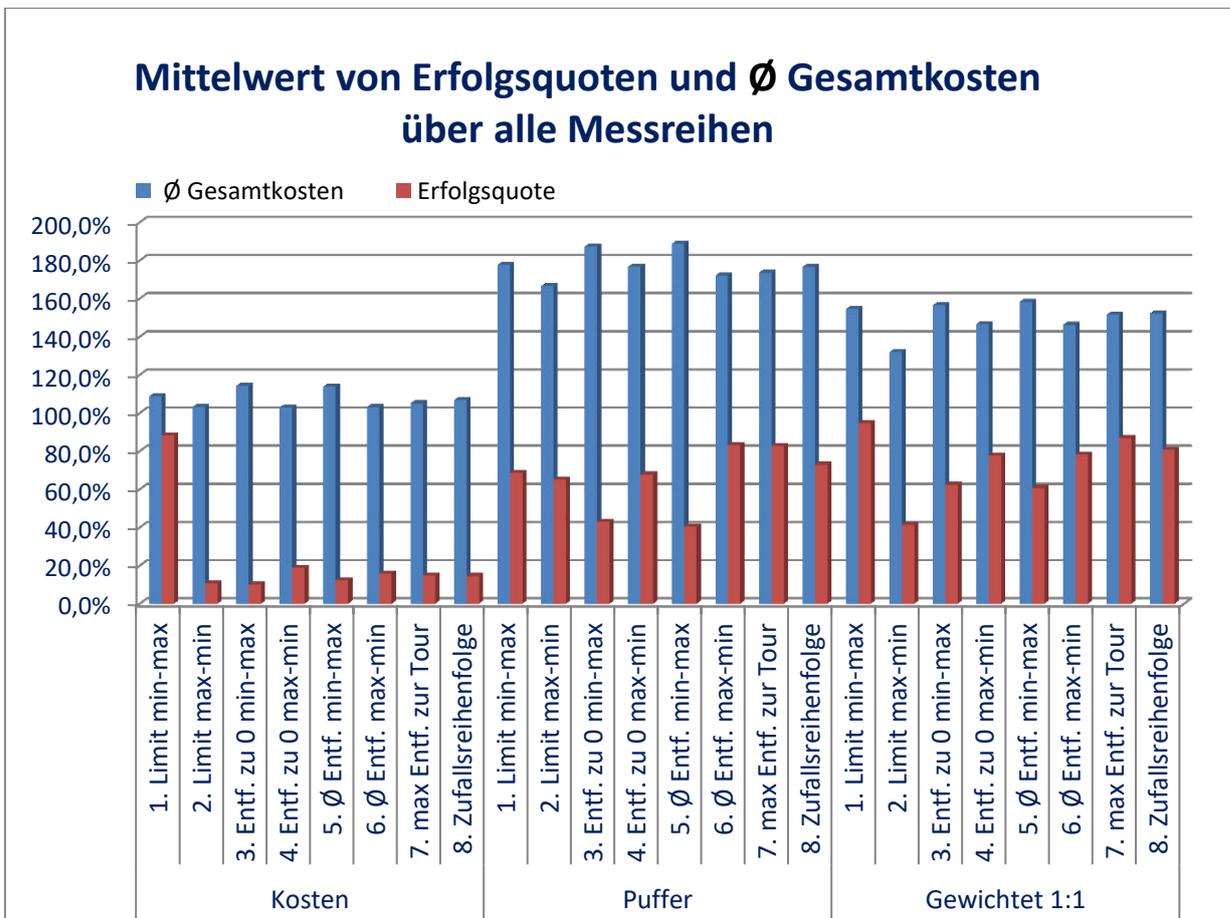


Abbildung 22: Grafische Darstellung der Erfolgsquoten und \emptyset Gesamtkosten als Mittelwert aller Messreihen

Hinsichtlich der Einfügereihenfolgen hat sich insbesondere gezeigt, dass das Kostenkriterium nur sinnvoll mit der Reihenfolge der aufsteigenden Limits angewendet werden kann. Für den zweiten Teil der Auswertung werden die anderen Einfügereihenfolgen in Verbindung mit dem Kostenkriterium daher nicht weiter verfolgt. Abseits des Kostenkriteriums werden die folgenden Bewertungen und Schlüsse für die Reihenfolgen getroffen:

1. **Aufsteigend nach Höhe des Limits sortiert**: Liefert die bisher besten Erfolgsquoten und funktioniert sehr gut mit kostenorientierten Entscheidungskriterien. Besonders hier ist die Auswirkung von unterschiedlichen Gewichtungen interessant.
2. **Absteigend nach Höhe des Limits sortiert**: Die Gesamtkosten sind im Mittel zwar knapp die Besten, jedoch sind die Erfolgsquoten - besonders im Zusammenhang mit einer Entscheidung nach Kosten – sehr schlecht. Da sich der zweite Teil primär mit Gewichtungen zugunsten von Kosten beschäftigen soll, wird diese Reihenfolge nicht weiter verfolgt.
3. **Aufsteigend nach „Entfernung vom Startpunkt“ sortiert**: Die Ergebnisse bei Erfolgsquote und Gesamtkosten sind durchweg denen der absteigenden Sortierung (Reihenfolge 4) unterlegen. Daher macht es keinen Sinn diese Systematik weiter zu verfolgen.
4. **Absteigend nach „Entfernung vom Startpunkt“ sortiert**: Diese Reihenfolge liefert solide Ergebnisse bei 20 und 50 Standorten. Mit zunehmender Standortanzahl kann

sie sich allerdings kaum von der Zufallsreihenfolge abheben und liefert zum Teil sogar schlechtere Erfolgsquoten. Dennoch wäre für den zweiten Teil interessant, wie die Ergebnisse auf veränderte Rahmenbedingungen reagieren.

5. **Aufsteigend nach „mittlerer Entfernung zu allen anderen Punkten“ sortiert**: Wie die Reihenfolge 3 ist auch diese Systematik ihrem Gegenstück (Folge 6) sowohl hinsichtlich der Erfolgsquote als auch bei den Gesamtkosten unterlegen und wird daher ebenfalls nicht weiter verfolgt.
6. **Absteigend nach „mittlerer Entfernung zu allen anderen Punkten“ sortiert** und
7. **Größte Entfernung zur bestehenden Tour**: Die Ergebnisse der beiden Einfügereihenfolgen sind erwartungsgemäß relativ ähnlich, beide erzielen mit Puffer- und gewichtetem Kriterium generell gute Ergebnisse. Es gibt jedoch eine nennenswerte Differenz der Erfolgsquote von 20% beim gewichteten Kriterium in der Messreihe mit 100 Standorten. Im zweiten Teil wäre es von Interesse, ob sich diese Differenz wiederholt und wie sich veränderte Rahmenbedingungen auswirken.
8. **Zufällige Reihenfolge**: Die zufällige Reihenfolge wird auch im zweiten Teil der Auswertung als Referenzwert verwendet.

7. Zweiter Teil der Analyse

7.1 Messreihen im zweiten Analyseteils

Im zweiten Teil der Analyse wird primär der Einfluss der Rahmenbedingungen untersucht. Da allerdings jede Veränderung der Rahmenbedingungen eine weitere, aufwendig durchzuführende Messreihe bedeutet, werden nicht alle Kombinationen untersucht. So wird der Einfluss der Lage des Startpunktes nur bei 20 zufällig verteilten Standorte untersucht. In allen weiteren Messreihen wird der Startpunkt zufällig generiert. Bei der gestreckten Verteilung ist dies angepasst.

Der Einfluss der Limits steht im Fokus des zweiten Teils der Untersuchung. Aber auch die Rahmenbedingungen der Standortanzahl und -verteilung werden näher untersucht. Es resultieren die folgenden 14 Messreihen:

- Messreihe 2.01: (20; zufällig; dezentral)
- Messreihe 2.02: (20; zufällig; zentral)
- Messreihe 2.03: (20; zufällig; zufällig)
- Messreihe 2.04: (50; zufällig; zufällig)
- Messreihe 2.05: (100; zufällig; zufällig)
- Messreihe 2.06: (20; zufällig gestreckt; zufällig)
- Messreihe 2.07: (50; zufällig gestreckt; zufällig)
- Messreihe 2.08: (100; zufällig gestreckt; zufällig)
- Messreihe 2.09: (20; zentrale Ballung; zufällig)
- Messreihe 2.10: (50; zentrale Ballung; zufällig)
- Messreihe 2.11: (100; zentrale Ballung; zufällig)

- Messreihe 2.12: (20; Städtecluster; zufällig)
- Messreihe 2.13: (50; Städtecluster; zufällig)
- Messreihe 2.14: (100; Städtecluster; zufällig)

Die in den einzelnen Messreihen untersuchten Einfügereihenfolgen wurden basierend auf den Ergebnissen des ersten Analyseteils eingeschränkt. Hinzugefügt werden hingegen einige Lösungsverfahren, welche weitere Ausprägungen des gewichteten Kriteriums verwenden.

7.2 Auswertung zweiter Teil

Zunächst werden kurz die durchgeführten Messreihen betrachtet. Tabelle 23 zeigt, wie viele der jeweils 500 generierten Probleme durch mindestens einen der Lösungswege gelöst werden konnten. Dabei wird unterschieden zwischen den Messreihen und deren Kombination mit den vier Arten der Limitgenerierung.

Anzahl durch min. einen Lösungsweg gelöste Probleme	Limitsys. 1 (unflexibel)	Limitsys. 4 (flex, Basis)	Limitsys. 5 (flex, gr. Raum)	Limitsys. 6 (flex, kl. Raum)
2.01: (20; zufällig; dezentral)	459	500	500	500
2.02: (20; zufällig; zentral)	478	500	498	500
2.03: (20; zufällig; zufällig)	482	500	500	500
2.04: (50; zufällig; zufällig)	484	500	500	500
2.05: (100; zufällig; zufällig)	491	499	499	487
2.06: (20; zufällig gestreckt; zufällig)	496	500	500	500
2.07: (50; zufällig gestreckt; zufällig)	488	500	500	500
2.08: (100; zufällig gestreckt; zufällig)	490	500	500	500
2.09: (20; zentrale Ballung; zufällig)	405	497	489	500
2.10: (50; zentrale Ballung; zufällig)	422	496	493	491
2.11: (100; zentrale Ballung; zufällig)	440	374	458	132
2.12: (20; Städtecluster; zufällig)	480	500	500	500
2.13: (50; Städtecluster; zufällig)	488	500	499	500
2.14: (100; Städtecluster; zufällig)	489	500	500	500
Mittelwert	470,8571429	490,4285714	495,4285714	472,1428571

Abbildung 23: Anzahl zulässig gelöster Probleme aller im zweiten Analyseteil vorgenommenen Messreihen

Es ist schnell ersichtlich, dass unter Verwendung der flexiblen Limits in den meisten Messreihen alle oder fast alle der generierten Probleme gelöst werden konnten. Da nur die kontrollierbar lösbar Probleme für die Auswertung genutzt werden können, hat dies den positiven Effekt einer großen Stichprobe. Auf der anderen Seite wird die Bewertung von Lösungswegen mit guten Erfolgsquoten erschwert. Wenn mithilfe eines Lösungsweges alle 500 Probleme einer Messreihe gelöst werden konnten, bleibt offen, wo die Grenzen des Lösungsweges liegen und ob die guten Ergebnisse primär durch eine „zu einfache“ Problemstellung zustande kamen. Auch lassen sich mehrere Lösungswege mit 100 prozentiger Erfolgsquote innerhalb einer Messreihe nicht gegeneinander abgrenzen.

Auf der anderen Seite des Spektrums lässt sich bei einer großen Anzahl nicht gelöster Probleme nicht feststellen, ob diese unlösbar sind oder lediglich alle Lösungswege scheiterten. Es ist jedoch damit zu rechnen, dass sich die guten Lösungswege stärker abheben können, je schwieriger es ist überhaupt eine zulässige Lösung zu erzielen.

Messreihe 2.11 mit einer zentralen Ballung und 100 Orten sticht besonders heraus. Diese Messreihe weist mit Abstand die geringste Anzahl gelöster Probleme auf, insbesondere in Kombination mit Limit 6, wo nur ca. ein Viertel aller Probleme gelöst werden konnten.

7.2.1 Anmerkung zum Einfluss der Standortverteilung

Die Idee hinter der Erhebung von Datenreihen mit unterschiedlichen Arten der Standortverteilung war es, deren Einfluss auf die Leistung der verschiedenen Lösungswege zu untersuchen. Leider hat sich jedoch gezeigt, dass sich nicht ohne weiteres gleiche Rahmenbedingungen für den Vergleich herstellen lassen. Das liegt vor allem an der Generierung der Limits, welche, wie in Teil 5.1.2 erklärt, auf die zufällige Generierung von Standorten auf einer 100x100 Einheiten großen Ebene ausgelegt ist.⁶⁷ Dabei werden die Limits in Abhängigkeit von der „Matrixgröße“ – bzw. der mittleren Entfernung zweier Punkte – und der Anzahl Standorte bestimmt. Dabei wird ein Verhältnis von Matrixgröße zur Limit-Obergrenze vorgeschrieben, welches nur von der Anzahl der Standorte abhängig ist. Es zeigte sich leider, dass dieses Verhältnis bei anderen Arten der Standortgenerierung nicht besonders sinnvoll ist.

An Grafik 23 ist ersichtlich, dass die Systematiken der Limitgenerierung in Zusammenspiel mit der geballten Standortgenerierung insgesamt zu wenigen gelösten Ergebnissen geführt haben. Demnach wurden in der Summe zu schwere Probleme generiert. Das Gegenteil ist bei der Generierung im gestreckten Raum sowie der „Städtecluster“-Generierung zu sehen. Hier wurden zu leicht lösbare Probleme generiert, was dazu führt, dass eine Vielzahl der Lösungswege extrem gute Erfolgsquoten erzielen konnten. Dadurch ist leider kaum Vergleichbarkeit zwischen Messreihen mit verschiedenen Arten der Standortgenerierung möglich. Die Werte innerhalb einer Messreihen lassen sich jedoch miteinander vergleichen. Da die erhobene Menge an Daten sehr groß ist, werden gezielt bestimmte Faktoren und Messreihen untersucht, während weniger aussagekräftige Daten nicht betrachtet werden.

7.2.2 Gegenüberstellung der Lösungswege allgemein

Zunächst wird eine allgemeine Untersuchung der aller Lösungswege hinsichtlich ihrer Erfolgsquoten und Gesamtkosten vorgenommen. Da dieser Schritt bereits im ersten Teil vorgenommen wurde, werden primär die neu hinzugefügten Lösungswege mit neuen Gewichtungen untersucht. Dargestellt wird hier ein Mittelwert aller Limitsystematiken bei zufälliger Generierung der Standorte inklusive des Starts (Messreihen 2.03 bis 2.05).

⁶⁷ Siehe dazu die zufällige Generierung in Teil 5.2

Lösungsweg	Erfolgsquote			Ø Gesamtkosten		
	20	50	100	20	50	100
Pufferkriterium						
1. Limit min-max	88%	84%	33%	172%	183%	182%
4. Entf. zu 1 max-min	91%	82%	30%	170%	182%	179%
6. Ø Entf. max-min	94%	94%	62%	167%	177%	176%
7. max Entf. zur Tour	92%	91%	64%	169%	180%	177%
8. Zufallsreihenfolge	86%	83%	48%	171%	182%	179%
Gewichtet gleich						
1. Limit min-max	96%	98%	88%	148%	160%	160%
4. Entf. zu 1 max-min	90%	88%	55%	140%	150%	152%
6. Ø Entf. max-min	87%	83%	61%	139%	150%	152%
7. max Entf. zur Tour	91%	89%	80%	145%	157%	158%
8. Zufallsreihenfolge	85%	85%	66%	145%	157%	159%
Gewichtet 1 zu 2						
1. Limit min-max	96%	98%	92%	134%	146%	148%
4. Entf. zu 1 max-min	82%	68%	33%	129%	135%	136%
6. Ø Entf. max-min	77%	60%	32%	127%	136%	138%
7. max Entf. zur Tour	82%	75%	57%	131%	142%	146%
8. Zufallsreihenfolge	77%	70%	49%	132%	142%	145%
Gewichtet 1 zu 4						
1. Limit min-max	96%	97%	91%	123%	129%	133%
4. Entf. zu 1 max-min	70%	40%	12%	119%	122%	121%
6. Ø Entf. max-min	66%	39%	14%	115%	123%	126%
7. max Entf. zur Tour	69%	58%	30%	117%	128%	134%
8. Zufallsreihenfolge	68%	50%	25%	119%	127%	130%
Gewichtet 1 zu 8						
1. Limit min-max	95%	97%	89%	117%	116%	119%
4. Entf. zu 1 max-min	60%	23%	3%	110%	111%	112%
6. Ø Entf. max-min	59%	23%	6%	107%	109%	115%
7. max Entf. zur Tour	59%	37%	12%	106%	112%	121%
8. Zufallsreihenfolge	57%	28%	10%	111%	113%	118%
Kostenkriterium						
1. Limit min-max	90%	93%	80%	118%	108%	103%
Gesamtergebnis	81%	71%	47%	134%	141%	143%

Abbildung 24: Erfolgsquoten und Ø Gesamtkosten bei zufälliger Generierung der Standorte

Tabelle 24 zeigt die untersuchten Lösungswege sortiert nach der Gewichtung der Kosten gegenüber den Puffern. Es ist schnell ersichtlich, dass die Gesamtkosten mit zunehmender Gewichtung der Kosten deutlich abnehmen, sich allerdings die Erfolgsquoten ebenfalls deutlich absenken. Ausnahme davon ist erneut die nach aufsteigenden Limits sortierte Einfügereihenfolge. Mit dieser Reihenfolge können besonders bei einer höheren Gewichtung der Kosten gute Ergebnisse erzielt werden. Aufbauend auf den Ergebnissen des ersten Teils lässt sich also sagen, dass sich die gewichteten Kriterien mit zunehmender Gewichtung

zugunsten der Kosten immer weiter an das Kostenkriterium annähern. Das gilt sowohl für die Nachteile bezüglich der Kompatibilität mit den meisten Einfügereihenfolgen, jedoch auch für das sehr gute Zusammenspiel mit der limitorientierten Einfügereihenfolge.

Hinsichtlich der Gegenüberstellung der entfernungsbasierten Einfügereihenfolgen 6 und 7 lässt sich feststellen, dass sich bei den gewichteten Kriterien der zusätzliche Berechnungsaufwand der Reihenfolge 7 in geringfügig, jedoch konsistent besseren Erfolgsquoten niederschlägt. Interessanterweise ist dies in allen drei Messreihen unter Verwendung des Pufferkriteriums exakt gegensätzlich.

7.2.3 Analyse des Einflusses der Limits

In diesem Teil soll der Einfluss der verschiedenen Arten der Limitgenerierung untersucht werden. Dafür werden die Messreihen 2.03 bis 2.05 (Zufällige Generierung von 20/50/100 Orten) betrachtet, da die Systematiken zur Limitgenerierung auf diese Rahmenbedingungen ausgelegt sind. Tabellen 25 bis 27 zeigen Gesamtkosten und Erfolgsquoten von ausgewählten Lösungswegen⁶⁸ sowie die absolute Höhe, welche die Ober- und Untergrenze im Mittel annehmen⁶⁹.

20 Orte (Tabelle 25): Die Ergebnisse hinsichtlich der Erfolgsquote sind bei den flexiblen Limitsystematiken 4, 5 und 6 sehr ähnlich, nur die Systematik 1 weist hier deutlich schlechtere Ergebnisse auf. Auffällig dabei ist, dass diese schlechteren Erfolgsaussichten sowohl auf die Ober- als auch auf die Untergrenze zurückzuführen sind. Systematik 5 generiert im Schnitt eine tiefere Untergrenze und Systematik 6 eine geringere Obergrenze. Dennoch können bei beiden Systematiken signifikant bessere Ergebnisse erzielt werden. Folglich können trotz schwieriger Ober- bzw. Untergrenzen bedingt gute Ergebnisse erzielt werden, sofern die jeweils andere Grenze besonders großzügig ist. Dies macht Sinn, da eine hohe Obergrenze eine größere Streckenlänge innerhalb der finalen Tour und somit auch die Wahl ineffizienterer Strecken erlaubt. Die Höhe der Untergrenze hingegen gibt vor, wie früh die Punkte mit niedrigen Limits in der finalen Tour angesteuert werden müssen. Hohe Untergrenzen führen somit zu deutlich mehr Flexibilität, weil sich die Reihenfolge innerhalb der finalen Tour weniger stark an der Höhe der Limits orientieren muss. Folglich können im Schnitt effizientere Touren gebildet werden, welche geringere mittlere Entfernungen aufweisen. Es ist jedoch extrem schwer diese Dynamik genauer einzuschätzen, insbesondere dann, wenn die Limits nicht - wie für diese Datenerhebung - gleichmäßig zwischen Ober- und Untergrenze verteilt sind.

⁶⁸ Gewählt wurden nur die Einfügereihenfolgen, welche mit den einzelnen Kriterien generell die besten Ergebnisse erzielen.

⁶⁹ Für die mittlere Höhe der Limits wurde mit einer mittleren Entfernung zwischen zwei Punkten von 52 Einheiten gerechnet.

	20 Orte Erfolgsquote				20 Orte Ø Gesamtkosten (absolut)			
	Lim 1	Lim 4	Lim 5	Lim 6	Lim 1	Lim 4	Lim 5	Lim 6
Ø Obergrenze	717,24	845,78	1065,06	689,16	717,24	845,78	1065,06	689,16
Ø Untergrenze	171,60	219,28	140,96	266,27	171,60	219,28	140,96	266,27
Pufferkriterium								
1. Limit min-max	58%	99%	99%	97%	705,73	773,04	911,26	641,22
6. Ø Entf. max-min	76%	100%	100%	99%	690,36	749,74	881,20	631,59
7. max Entf. zur Tour	71%	99%	99%	98%	694,06	757,58	899,23	636,56
Gewichtet gleich								
1. Limit min-max	84%	100%	100%	99%	622,85	662,07	773,84	562,90
7. max Entf. zur Tour	71%	98%	97%	97%	609,20	648,67	737,20	560,37
Gewichtet 1 zu 2								
1. Limit min-max	84%	100%	100%	99%	575,53	600,33	691,00	521,60
7. max Entf. zur Tour	53%	91%	92%	90%	548,51	582,54	659,86	508,65
Gewichtet 1 zu 4								
1. Limit min-max	84%	100%	100%	99%	552,77	538,60	604,75	487,35
7. max Entf. zur Tour	38%	80%	83%	77%	486,38	507,63	579,75	458,52
Gewichtet 1 zu 8								
1. Limit min-max	81%	100%	100%	99%	546,69	507,45	541,28	484,12
7. max Entf. zur Tour	29%	68%	71%	66%	445,87	457,35	496,01	432,44
Kostenkriterium								
1. Limit min-max	69%	98%	99%	94%	558,72	523,24	519,49	506,15

Abbildung 25: Erfolgsquoten und Ø Gesamtkosten ausgewählter Lösungswege bei 20 zu besuchenden Orten

Hinsichtlich der Gesamtkosten lässt sich ein sehr differenziertes Bild erkennen. Neben den bereits im ersten Teil der Auswertung erkannten Unterschieden zwischen den Lösungswegen lassen sich hier auch erhebliche Unterschiede zwischen den mit unterschiedlichen Limits durchgeführten Reihen feststellen. Im Allgemeinen führt die Anhebung einer oder beider Grenzen ebenfalls zur Anhebung aller Limits⁷⁰, da diese gleichmäßig im Bereich zwischen den beiden Grenzen generiert werden. Die höheren Limits führen wiederum dazu, dass mehr Freiheiten bei der Bildung der Tour vorhanden sind. Im Schnitt sollte es daher möglich sein, effizientere Touren mit geringeren Gesamtkosten zu konstruieren. Da sich durch Anhebung einer Grenze auch kein Wert verschlechtert, sollten auch die Gesamtkosten im schlechtesten Fall auf dem gleichen Niveau bleiben. Am Vergleich der Ergebnisse der Limitsystematiken 1 und 4 lässt sich jedoch erkennen, dass diese Erwartung hier zum Großteil nicht erfüllt werden konnte. Nur unter Verwendung der Einfügereihenfolge 1 in Verbindung mit dem Kostenkriterium sowie den Gewichtungen 1 zu 4 und 1 zu 8 führt die zusätzliche Flexibilität zu durchschnittlich niedrigeren Kosten. Mit stärkerer Gewichtung der Puffer ist sogar das Gegenteil sichtbar, da die Gesamtkosten unter den großzügigeren Limits von Systematik 4 deutlich höher sind. Am deutlichsten sichtbar ist

⁷⁰ Davon ausgenommen ist das Limit, welches genau der anderen Grenze entspricht. Also bei Anhebung der Obergrenze verändern sich alle Limits außer dem niedrigsten und umgekehrt.

dies bei den Lösungswegen des Pufferkriteriums, wo sich die Höhe der Gesamtkosten dem Wert der Obergrenze annähern. Dies ist noch eindeutiger zu sehen an den Ergebnissen der Datenreihen mit 50 und 100 Orten (Tabellen 26 und 27).

Diese Annäherung an die Obergrenze lässt sich aufgrund der Vorgehensweise des Pufferkriteriums erklären, da diese keine Systematik enthält, welche zu einer Reduktion der Kosten führt. Im Gegenteil werden solche Lösungen begünstigt, welche die Limits weitgehend ausschöpfen. Dieser Effekt wird als Exkurs in Teil 7.2.6 genauer aufgezeigt und erklärt. Auch beim gewichteten Kriterium wird der Effekt zwar nicht aufgehoben, allerdings dadurch abgeschwächt, dass die Höhe der Gesamtkosten ebenfalls zur Wahl der Einfügeposition beiträgt. Je höher die Gewichtung der Puffer gegenüber den Kosten ist, desto stärker ist der Effekt in den Ergebnissen zu erkennen.

50 Orte (Tabelle 26):

	50 Orte Erfolgsquote				50 Orte Ø Gesamtkosten (absolut)			
	Lim 1	Lim 4	Lim 5	Lim 6	Lim 1	Lim 4	Lim 5	Lim 6
Ø Obergrenze	1793,10	1524,55	1867,27	1229,09	1793,10	1524,55	1867,27	1229,09
Ø Untergrenze	171,60	271,82	195,00	342,73	171,60	271,82	195,00	342,73
Pufferkriterium								
1. Limit min-max	69%	89%	94%	85%	1762,18	1488,80	1796,70	1214,07
6. Ø Entf. max-min	88%	98%	99%	92%	1704,41	1445,94	1731,52	1188,32
7. max Entf. zur Tour	78%	97%	97%	93%	1729,24	1462,70	1763,68	1188,97
Gewichtet gleich								
1. Limit min-max	94%	100%	100%	99%	1526,00	1321,93	1540,51	1090,77
7. max Entf. zur Tour	75%	95%	93%	95%	1476,12	1296,58	1490,68	1087,13
Gewichtet 1 zu 2								
1. Limit min-max	93%	100%	100%	98%	1379,03	1208,59	1385,20	1014,95
7. max Entf. zur Tour	52%	84%	79%	85%	1328,04	1179,13	1330,64	1005,28
Gewichtet 1 zu 4								
1. Limit min-max	93%	99%	100%	98%	1215,45	1070,84	1210,74	933,87
7. max Entf. zur Tour	39%	65%	65%	65%	1168,03	1054,03	1183,47	901,13
Gewichtet 1 zu 8								
1. Limit min-max	93%	99%	100%	97%	1075,29	956,60	1053,79	877,27
7. max Entf. zur Tour	23%	41%	46%	38%	1012,27	896,78	1025,57	783,43
Kostenkriterium								
1. Limit min-max	84%	98%	99%	92%	953,17	920,93	898,11	909,41

Abbildung 26: Erfolgsquoten und Ø Gesamtkosten ausgewählter Lösungswege bei 50 zu besuchenden Orten

Auch bei 50 Orten weisen die flexiblen Limitsystematiken sehr ähnliche Erfolgsquoten auf. Interessant ist hier jedoch besonders der Vergleich zwischen den Limitsystematiken 1 und 5, welcher aufzeigt, wie komplex die Dynamik von Grenzen und resultierenden Erfolgsquoten

ist. Während beide Grenzen bei Systematik 5 im Durchschnitt nur geringfügig (ca. 4% bzw. 14%) höher sind als bei Systematik 1, sind die Differenzen der Erfolgsquoten mit bis zu 25% erheblich. Auch Limitsystematik 6 hebt sich gegen Systematik 1 erneut mit durchweg besseren Erfolgsquoten ab, obwohl die generierten Limits im Schnitt 20% niedriger liegen.⁷¹ Demzufolge lässt sich aus der durchschnittlichen Höhe der Limits nur bedingt eine Schlussfolgerung bezüglich der Lösbarkeit eines Problems ziehen.

100 Orte (Tabelle 27) und Vergleich aller Messreihen:

	100 Orte Erfolgsquote				100 Orte Ø Gesamtkosten (absolut)			
	Lim 1	Lim 4	Lim 5	Lim 6	Lim 1	Lim 4	Lim 5	Lim 6
Ø Obergrenze	3586,21	2130,32	2583,23	1710,97	3586,21	2130,32	2583,23	1710,97
Ø Untergrenze	171,60	318,71	243,23	410,97	171,60	318,71	243,23	410,97
Pufferkriterium								
1. Limit min-max	67%	17%	32%	14%	3537,16	2194,04	2612,95	1787,01
6. Ø Entf. max-min	91%	51%	80%	27%	3418,97	2150,65	2566,86	1764,15
7. max Entf. zur Tour	78%	58%	69%	52%	3474,77	2146,31	2580,36	1743,28
Gewichtet gleich								
1. Limit min-max	95%	88%	97%	69%	3032,54	2022,19	2385,86	1656,16
7. max Entf. zur Tour	79%	80%	82%	80%	2928,22	2010,28	2362,03	1642,97
Gewichtet 1 zu 2								
1. Limit min-max	96%	94%	98%	81%	2717,91	1902,45	2197,96	1587,40
7. max Entf. zur Tour	54%	57%	54%	64%	2587,94	1874,75	2156,27	1555,35
Gewichtet 1 zu 4								
1. Limit min-max	95%	91%	96%	81%	2358,20	1723,92	1956,81	1490,91
7. max Entf. zur Tour	31%	29%	29%	32%	2299,63	1698,51	1942,51	1437,48
Gewichtet 1 zu 8								
1. Limit min-max	95%	91%	95%	75%	2007,49	1575,96	1712,39	1430,84
7. max Entf. zur Tour	18%	10%	13%	8%	2069,05	1485,55	1709,42	1250,66
Kostenkriterium								
1. Limit min-max	88%	81%	89%	61%	1389,32	1486,34	1473,53	1442,40

Abbildung 27: Erfolgsquoten und Ø Gesamtkosten ausgewählter Lösungswege bei 100 zu besuchenden Orten

Bei der Messreihe mit 100 Orten weisen die die unter Limitsystematik 6 erzielten Erfolgsquoten im Vergleich mit den anderen Systematiken deutlich schlechtere Werte auf. Die niedrige Obergrenze wirkt sich hier besonders stark auf die Ergebnisse der Einfügereihenfolge 1 aus. Auffällig ist jedoch besonders Einfügereihenfolge 7 in Zusammenhang mit den gewichteten Kriterien. Im Gegensatz zu Reihenfolge 1 zeigen dessen Erfolgsquoten nahezu keine Unterschiede zwischen den Systematiken. Interessant ist dies besonders in Zusammenhang mit den Ergebnissen der Messreihen mit 20 und 50 Orten, bei

⁷¹ Da die Limits gleichmäßig zwischen den Grenzen generiert werden, liegt der Mittelwert der Limits genau in der Mitte. (Mittelwert = Untergrenze + (Obergrenze – Untergrenze)/2) Bei 50 Orten ergibt sich für Systematik 1 ein Mittelwert von 982,35 und für Systematik 6 ein Mittelwert von 785,91.

denen die Erfolgsquoten von Reihenfolge 7 eine besonders starke negative Reaktion auf die niedrigen Limits von Systematik 1 aufweisen.

Eine mögliche Begründung dafür lässt sich an zwei Punkten ansetzen. Einerseits könnte besonders die Untergrenze einen sehr großen Einfluss auf die Leistung dieser Einfügereihenfolge haben. Das heißt bei 20 und 50 Orten nimmt die Leistung durch die niedrige Untergrenze stark ab, während durch die hohe Untergrenze bei 100 Orten (und Limitsystematik 6) auch die knappe Obergrenze gut kompensiert werden kann. Alternativ oder ergänzend könnte auch die erhöhte Anzahl von Orten einen stabilisierenden Effekt auf die Leistung der Einfügereihenfolge haben. Die Höhe der Limit-Obergrenze scheint auf jeden Fall einen deutlich geringeren Einfluss auf Reihenfolge 7 zu haben als auf Reihenfolge 1.

Hinsichtlich der Entscheidungskriterien können die 1 zu 2 und 1 zu 4 gewichteten Kriterien unter der kritischen Limitsystematik 6 (bei 100 Orten) mit der limitorientierten Reihenfolge die besten Ergebnisse erzielen. Die Differenz zu den anderen Kriterien ist dabei erheblich größer als bei anderen Messreihen und Limits. Der Einfluss der Puffer bei der Wahl der Einfügeposition macht sich hier also definitiv positiv bemerkbar.

Die Ergebnisse bei 100 Orten weisen des Weiteren einen Umschwung der Leistung unter Limitsystematik 1 auf. Durch die fixierte Generierung der Obergrenze liegt diese nun erheblich höher generiert, was sich in Form von deutlich höheren Erfolgsquoten bemerkbar macht. Der Vergleich mit Limitsystematik 5 zeigt, dass dessen um ca. 70 Einheiten höhere Untergrenze die Differenz von ca. 1000 Einheiten zwischen den Obergrenzen ausgleichen kann, wodurch sehr ähnliche Erfolgsquoten resultieren.

Bezieht man nun allerdings die Gesamtkosten in die Betrachtung ein, so wird die Ineffizienz der pufferorientierten Entscheidung hier besonders deutlich. Unter Limitsystematik 1 weisen sogar die stärker zugunsten der Kosten gewichteten Kriterien stark überdurchschnittliche Gesamtkosten auf. Dazu im Gegensatz stehen die Ergebnisse unter Systematik 6, bei der sogar die ineffizienteren Lösungswege bessere Gesamtkosten vorbringen können als die besten bei Systematik 1 (ausgenommen des Kostenkriteriums). Auch werden unter Systematik 6 mit den 1 zu 4 und 1 zu 8 gewichteten Kriterien Gesamtkosten erzielt, welche vergleichbar oder besser als die des Kostenkriteriums sind. Daraus kann abgeleitet werden, dass mit zunehmendem Einfluss der Puffer auf die Entscheidung nicht zwangsläufig auch die Kosten ansteigen. Dennoch ist dieser Trend klar zu erkennen, besonders bei den Systematiken, welche eine hohe Obergrenze generieren.

Es muss folglich der Schluss gezogen werden, dass der Einfluss der Puffer auf das Entscheidungskriterium zu einer Annäherung der Gesamtkosten an die gegebenen Limitobergrenzen führt. Dieser Effekt verstärkt sich mit zunehmender Gewichtung. Es ist hier anzumerken, dass auch die durch die Untergrenze bedingte Flexibilität einen Einfluss auf die Gesamtkosten haben kann, jedoch scheint dies nur einen geringen Effekt zu haben. Erkennen lässt sich dies vor allem an den Ergebnissen des Kostenkriteriums, da diese nur von den Kosten beeinflusst werden. Bei allen drei Messreihen lassen sich hier hinsichtlich der

Gesamtkosten nur geringfügige Unterschiede erkennen. Bei 100 Orten konnten unter der großzügigen Obergrenze von Limitsystematik 1 die besten Ergebnisse erzielt werden. Weil durch diese Systematik die durchschnittlich mit Abstand höchsten Limits generiert werden, macht es Sinn, dass sich die erhöhte Flexibilität in den Gesamtkosten widerspiegelt. Es ist jedoch auch anzumerken, dass der Unterschied der Gesamtkosten nicht annähernd so groß ist, wie die Differenz der durchschnittlichen Limits impliziert. So sind die mittleren Limits im Vergleich von Limitsystematik 1 zu 4 um über 50% höher⁷², während die Differenz der Gesamtkosten unter 10% beträgt. Folglich müssen die hohen Differenzen des Pufferkriteriums sowie der gewichteten Kriterien zu einem sehr großen Teil auf die Vorgehensweise der pufferorientierten Entscheidung zurückzuführen sein.

Eine interessante Beobachtung lässt sich noch bei den durchs Pufferkriterium erzielten Ergebnissen vornehmen. Bei den Lösungswegen mit niedrigen Erfolgsquoten liegen die durchschnittlichen Gesamtkosten zum Teil oberhalb der durchschnittlichen Limit-Obergrenze. Da nur zulässig gelöste Ergebnisse einbezogen werden, bedeutet das, dass hauptsächlich Probleme gelöst werden konnten, welche eine überdurchschnittlich große Matrixgröße und somit Obergrenze haben.

Der Einfluss der Puffer auf die Entscheidung kann besonders bei Problemen mit knappen Limits einen positiven Effekt erzielen. So weisen die gewichteten Kriterien in Zusammenhang mit Einfügereihenfolge 1 meist bessere Erfolgsquoten auf als das Kostenkriterium. Hier zeigt sich jedoch, dass ohne Einfluss der Kosten gerade die Probleme mit geringer Obergrenze besonders selten gelöst werden können.

7.2.4 Untersuchung der Ergebnisse bei zentraler Ballung der Standorte

Die Ergebnisse der Messreihen mit zentraler Ballung der Standorte sind besonders interessant, da nur ein geringer Teil der generierten Probleme überhaupt gelöst werden konnte. Demnach sind die generierten Probleme allgemein deutlich schwieriger zu lösen als die der anderen Messreihen. Folglich lassen sich die Grenzen der einzelnen Lösungswege hier besser abschätzen als bei den leichter zu lösenden Problemen. Diese Teil der Untersuchung fokussiert sich lediglich auf die Erfolgsquoten, da die Gesamtkosten im letzten Teil bereits ausführlich untersucht wurden. Tabelle 28 zeigt die resultierenden Erfolgsquoten der Messreihen 2.09 und 2.10 (zentrale Ballung mit 20/ 50 Orten).

⁷² Für Systematik 1 ergibt sich ein Mittelwert von $171,60 + \frac{3586,21-171,60}{2} = 1878,9$, während Systematik 4 einen Mittelwert von 1224,5 Einheiten aufweist.

Lösungsweg	Erfolgsquote 20 Orte				Erfolgsquote 50 Orte			
	Lim 1	Lim 4	Lim 5	Lim 6	Lim 1	Lim 4	Lim 5	Lim 6
Pufferkriterium								
1. Limit min-max	32%	89%	96%	80%	24%	38%	61%	24%
4. Entf. zu 0 max-min	51%	93%	98%	83%	49%	57%	80%	31%
6. \emptyset Entf. max-min	67%	95%	99%	93%	78%	89%	94%	70%
7. max Entf. zur Tour	65%	96%	99%	92%	67%	84%	91%	73%
8. Zufallsreihenfolge	37%	89%	93%	77%	45%	54%	76%	31%
Gewichtet gleich								
1. Limit min-max	69%	97%	99%	91%	79%	91%	95%	72%
4. Entf. zu 0 max-min	56%	92%	95%	85%	65%	75%	86%	55%
6. \emptyset Entf. max-min	48%	88%	89%	83%	44%	68%	72%	59%
7. max Entf. zur Tour	57%	92%	94%	88%	64%	85%	85%	76%
8. Zufallsreihenfolge	47%	86%	92%	81%	56%	76%	83%	57%
Gewichtet 1 zu 2								
1. Limit min-max	70%	97%	99%	93%	82%	95%	96%	76%
4. Entf. zu 0 max-min	38%	77%	86%	74%	39%	52%	63%	37%
6. \emptyset Entf. max-min	32%	72%	79%	67%	23%	37%	46%	29%
7. max Entf. zur Tour	39%	79%	82%	71%	42%	60%	66%	54%
8. Zufallsreihenfolge	32%	79%	83%	70%	36%	55%	62%	43%
Gewichtet 1 zu 4								
1. Limit min-max	72%	97%	99%	92%	84%	93%	96%	77%
4. Entf. zu 0 max-min	27%	65%	73%	58%	15%	24%	34%	14%
6. \emptyset Entf. max-min	22%	57%	67%	51%	15%	19%	27%	13%
7. max Entf. zur Tour	26%	62%	70%	56%	17%	34%	40%	27%
8. Zufallsreihenfolge	26%	65%	72%	54%	21%	32%	39%	22%
Gewichtet 1 zu 8								
1. Limit min-max	71%	97%	98%	91%	83%	92%	96%	71%
4. Entf. zu 0 max-min	20%	51%	62%	49%	7%	10%	15%	5%
6. \emptyset Entf. max-min	18%	48%	54%	43%	7%	8%	14%	4%
7. max Entf. zur Tour	18%	48%	57%	45%	8%	13%	20%	5%
8. Zufallsreihenfolge	17%	52%	59%	43%	12%	14%	24%	7%
Kostenkriterium								
1. Limit min-max	59%	93%	97%	83%	78%	88%	90%	58%

Abbildung 28: Erfolgsquoten bei zentral geballter Generierung von 20/ 50 Orten

Hier lässt sich allgemein das gleiche Muster erkennen, dass die besten Ergebnisse mit der Einfügereihenfolge 1 erzielt werden können. Ähnlich wie im letzten Teil liegen die unter Verwendung von Limitsystematik 1 generierten Ergebnisse auch hier deutlich unterhalb des Durchschnitts. Wie in Tabelle 23 ersichtlich ist, konnten bei den Messreihen mit 20 und 50 Orten nur 405 bzw. 422 von den 500 generierten Problemen gelöst werden. Da die Lösungsquoten nur bei ca. 70% (von 405 lösbaren Ergebnissen) und ca. 80% (von 422 lösbaren Ergebnissen) liegen, wurden folglich mit den besten Lösungswegen nur um die 280 bzw. 340 zulässige Ergebnisse erzielt. Daher kann hier der Schluss gezogen werden, dass

auch die Leistung der besseren Lösungsverfahren unter schwierigen Bedingungen abnimmt. Das lässt sich dadurch erklären, dass sich der Spielraum, in dem Ineffizienzen hingenommen werden können, verkleinert. Somit steigt die Wahrscheinlichkeit, dass sich das Verfahren durch eine schlechte Einfügeentscheidung selbst blockiert und in der Folge scheitert.

Lösungsweg	100 Orte			
	Erfolgsquote			
	Lim 1	Lim 4	Lim 5	Lim 6
Pufferkriterium				
1. Limit min-max	22%	0%	0%	0%
4. Entf. zu 0 max-min	52%	3%	5%	2%
6. ∅ Entf. max-min	82%	18%	52%	2%
7. max Entf. zur Tour	70%	26%	50%	25%
8. Zufallsreihenfolge	46%	1%	8%	1%
Gewichtet gleich				
1. Limit min-max	83%	22%	60%	8%
4. Entf. zu 0 max-min	66%	8%	36%	2%
6. ∅ Entf. max-min	46%	19%	35%	10%
7. max Entf. zur Tour	59%	48%	59%	44%
8. Zufallsreihenfolge	52%	18%	41%	5%
Gewichtet 1 zu 2				
1. Limit min-max	88%	44%	74%	14%
4. Entf. zu 0 max-min	36%	6%	19%	1%
6. ∅ Entf. max-min	22%	4%	13%	2%
7. max Entf. zur Tour	38%	26%	30%	29%
8. Zufallsreihenfolge	33%	14%	20%	6%
Gewichtet 1 zu 4				
1. Limit min-max	86%	53%	75%	24%
4. Entf. zu 0 max-min	14%	1%	5%	1%
6. ∅ Entf. max-min	10%	1%	2%	1%
7. max Entf. zur Tour	15%	5%	10%	5%
8. Zufallsreihenfolge	16%	4%	10%	2%
Gewichtet 1 zu 8				
1. Limit min-max	88%	48%	75%	25%
4. Entf. zu 0 max-min	3%	0%	1%	0%
6. ∅ Entf. max-min	4%	0%	0%	0%
7. max Entf. zur Tour	6%	1%	3%	0%
8. Zufallsreihenfolge	6%	0%	4%	0%
Kostenkriterium				
1. Limit min-max	76%	39%	66%	14%

Abbildung 29: Erfolgsquoten bei zentral geballter Generierung von 100 Orten

Die Wahrscheinlichkeit, dass dies eintritt nimmt im Allgemeinen durch eine geringere Fehlertoleranz zu. Als Konsequenz ist es erheblich mehr dem Zufall überlassen, ob das

Verfahren gelingt, auch wenn das Problem lösbar ist. Noch deutlicher wird dies an der Messreihe 2.11 mit 100 Standorten erkennbar (siehe Tabelle 29).

Während bei 20 Standorten in Kombination mit Limitsystematik 1 nur Erfolgsquoten von bis zu 70% erzielt wurden, zeigt sich hier der Effekt der unflexiblen Limitgenerierung, welche zu deutlich höheren Limit-Obergrenzen und in der Konsequenz zu höheren Erfolgsquoten führt. Die Ergebnisse der Limitsystematik 5 sind ähnlich, da diese Systematik ebenfalls eine angehobene Limit-Obergrenze hervorbringt. Im Gegensatz dazu stehen die Systematiken 4 und besonders 6, bei welchen die Obergrenze deutlich tiefer liegt. Wie aus Tabelle 23 hervorgeht, konnten hier nur 374 (Limitsystematik 4) bzw. 132 (Limitsystematik) der je 500 generierten Probleme überhaupt gelöst werden. Und auch hier setzt sich der Trend weiter fort, dass selbst die besseren Erfolgsquoten sehr unterdurchschnittlich sind. Die jeweils besten Quoten liegen bei gerade einmal 53% und 44%, was gerade einmal 197 und 58 gelösten Ergebnissen entspricht. Besonders auffällig hierbei sind die Resultate bei den reduzierten Limits von Systematik 6, wo die sonst so verlässlichen Kombinationen aus limitorientierter Einfügereihenfolge mit kostenorientiertem Entscheidungskriterium nur noch Erfolgsquoten zwischen 8% und 25% aufweisen können. Das beste Ergebnis liefert hier das gleich gewichtete Kriterium unter Verwendung der Einfügereihenfolge 7 (maximale Entfernung zur Tour). Dies ist überraschend, da die Erfolgsquote dieser Kombination auch bei den großzügigeren Limits nicht wesentlich bessere Quoten aufweisen kann.

Diese Beobachtung geht einher mit der Beobachtungen aus dem letzten Teil, dass die Kombination aus limitorientierter Reihenfolge und kostenorientiertem Entscheidungskriterium zwar bei Problemen mit ausreichend Fehlertoleranz sehr gute Ergebnisse erzielt, jedoch bei knappen Limits besonders fehleranfällig ist. Einfügereihenfolge 7 zeigt sich auch hier als weniger abhängig von der Höhe des Limits, was sich in einer geringeren Schwankung der Erfolgsquote äußert. Leider scheinen jedoch andere Faktoren dazu zu führen, dass diese Quote insgesamt nicht sonderlich gut ist. Zu diesem Schluss ist anzumerken, dass die aus dieser Messreihe resultierende Stichprobe mit nur 132 gelösten Ergebnissen in nicht besonders groß ist, was die Aussagekräftigkeit der Ergebnisse begrenzt.

7.2.5 Lage des Startpunktes

Im Folgenden werden die Messreihen 2.01 bis 2.03 dahingehend untersucht, in wie weit sich die Lage des Startpunktes bei einer relativ kleinen Anzahl zu besuchender Standorte (hier 20) auf die Leistung der Lösungswege auswirkt. Dafür wird die Limitsystematik 1 verwendet, da die restlichen Systematiken durchweg sehr gute Erfolgsquoten liefern. Dadurch können hier leider nur bei den unflexiblen Limits nennenswerte Unterschiede erkannt werden. Tabelle 30 zeigt die ermittelten Erfolgsquoten der Messreihen. Hinsichtlich der Gesamtkosten lassen sich kaum Unterschiede zwischen den Messreihen erkennen, weshalb sie hier nicht aufgeführt sind.

Lösungsweg	20 Orte		
	Erfolgsquote		
	Dezentral	Zentral	Zufällig
Pufferkriterium			
1. Limit min-max	54%	48%	58%
4. Entf. zu 0 max-min	60%	78%	68%
6. ∅ Entf. max-min	74%	77%	76%
7. max Entf. zur Tour	71%	69%	71%
8. Zufallsreihenfolge	54%	51%	54%
Gewichtet gleich			
1. Limit min-max	80%	87%	84%
4. Entf. zu 0 max-min	58%	68%	68%
6. ∅ Entf. max-min	55%	67%	65%
7. max Entf. zur Tour	62%	67%	71%
8. Zufallsreihenfolge	57%	57%	59%
Gewichtet 1 zu 2			
1. Limit min-max	79%	91%	84%
4. Entf. zu 0 max-min	48%	49%	57%
6. ∅ Entf. max-min	44%	49%	49%
7. max Entf. zur Tour	46%	54%	53%
8. Zufallsreihenfolge	43%	42%	46%
Gewichtet 1 zu 4			
1. Limit min-max	78%	88%	84%
4. Entf. zu 0 max-min	39%	40%	42%
6. ∅ Entf. max-min	33%	35%	34%
7. max Entf. zur Tour	35%	42%	38%
8. Zufallsreihenfolge	27%	30%	38%
Gewichtet 1 zu 8			
1. Limit min-max	76%	88%	81%
4. Entf. zu 0 max-min	30%	31%	26%
6. ∅ Entf. max-min	29%	27%	28%
7. max Entf. zur Tour	27%	32%	29%
8. Zufallsreihenfolge	26%	21%	25%
Kostenkriterium			
1. Limit min-max	63%	81%	69%

Abbildung 30: Erfolgsquoten in Abhängigkeit der Lage des Startpunkts

Insgesamt sind die Unterschiede zwischen den einzelnen Messreihen gering und bei einigen Lösungswegen kaum zu erkennen. Auch die Anzahl zulässig gelöster Probleme liegt bei den drei Reihen mit 459, 478 und 482 dabei relativ nah beieinander. Die größten Unterschiede lassen sich bei der vom Startpunkt abhängigen Einfügereihenfolge 4 in Verbindung mit Pufferkriterium und gleich gewichtetem Kriterium erkennen. So ist die Erfolgsquote bei zentraler Lage des Startpunktes um die 20% besser als bei dezentralem Startpunkt. Dabei erreicht die Erfolgsquote jedoch bei zentralem Startpunkt gerade einmal die Güte der anderen entfernungsbasierten Reihenfolgen 6 und 7.

Somit kann hier keine konkrete Ausgangssituation beschrieben werden, in der es sinnvoll ist, die am Startpunkt orientierte Einfügereihenfolge 4 den Folgen 6 oder 7 vorzuziehen. Hinsichtlich der limitorientierten Einfügereihenfolge 1 lassen sich ebenfalls geringfügige Unterschiede bei den Erfolgsquoten erkennen. Diese sind bei zentraler Lage des Starts ca. 5-10% besser als bei zufälliger Generierung und ca. 10-15% besser als bei dezentraler Lage. Eine mögliche Erklärung dafür findet sich darin, dass der Startpunkt bei zentraler Lage mit ungefähr 38 Einheiten im Durchschnitt deutlich näher am ersten zu besuchenden Punkt liegt. Bei zufälliger Generierung sind dies ca. 52 Einheiten, bei dezentraler Lage sogar um die 76 Einheiten. Die einfache Erreichung des ersten Punktes bringt Flexibilität, da die Zeitpuffer des kritischsten Ortes für die nachfolgenden Schritte höher sind. Die Beobachtungen hier würden demnach die Hypothese aus den letzten Teilen der Auswertung unterstützen, dass sich hohe Flexibilität besonders positiv auf die Erfolgsquote der limitorientierten Einfügereihenfolge 1 auswirkt⁷³.

7.2.6 Exkurs zum Einfluss der Puffer im Entscheidungskriterium

An mehreren Stellen der Analyse konnte bereits der negative Einfluss der Puffer (im Entscheidungskriterium) für die Gesamtkosten anhand von Daten belegt werden. Ein Erklärungsansatz für diesen Effekt kann aus der Vorgehensweise des Pufferkriteriums abgeleitet werden:

Da bei jedem Einfügeschritt immer das Minimum aller veränderten Puffer zur Entscheidung herangezogen wird, ist es wahrscheinlicher, dass eine der hinteren Einfügepositionen gewählt wird, weil dessen Minimum aus weniger einzelnen Puffern bestimmt wird. Wird ein Punkt mit sehr hohem Limit eingefügt, so ist es wahrscheinlich, dass dieser ganz am Ende eingefügt wird, weil dort nur der eigene Puffer berücksichtigt wird. So kann es mitunter zur Wahl extrem ungünstiger Verbindungen kommen, wenn das Limit des einzufügenden Punktes ausreichend Spielraum für diese Ineffizienzen lässt.

Besonders kritisch ist dies, wenn Orten kein Limit zugewiesen ist und sie folglich ein unbegrenzt hohes Limit haben. Es würde somit sogar auch beim gewichteten Kriterium in nahezu jedem Fall zur Wahl der letzten Einfügeposition kommen, auch wenn diese extrem schlecht ist.

An der Tendenz zur Wahl der hinteren Einfügepositionen bei hohen Limits lässt sich auch erklären, dass unter Verwendung des Pufferkriteriums die Einfügereihenfolge nach absteigenden Limits bessere Ergebnisse liefert als die sonst weit überlegene aufsteigende Sortierung.⁷⁴

7.3 Fazit der Auswertung

In diesem Teil wird ein zusammenfassendes Fazit aus den Ergebnissen der Evaluation gezogen. Dabei wird primär versucht die Leistung der einzelnen Lösungswege

⁷³ Dies gilt nur für die Kombination mit Kostenkriterium und den getesteten gewichteten Kriterien, nicht fürs Pufferkriterium.

⁷⁴ Siehe dazu Abbildungen 20 und 21 im ersten Teil der Auswertung (Teil 6.2 der Arbeit)

zusammenfassend zu bewerten und für gewisse Rahmenbedingungen Empfehlungen abzugeben.

Hinsichtlich der Entscheidungskriterien haben sich sowohl das reine Kostenkriterium als auch das Pufferkriterium als suboptimal erwiesen. Die Systematik des Pufferkriteriums führt zu durchweg sehr hohen Gesamtkosten, welche sich an die Höhe der Obergrenze annähern, sogar wenn diese extrem hoch liegt. Während die Erfolgsquoten zwar zum Teil gut sind, so sind sie nur selten die besten.

Auf der andern Seite des Spektrums hat es sich gezeigt, dass das Kostenkriterium nur in Zusammenhang mit der limitorientierten Reihenfolge 1 verlässlich ein zulässiges Ergebnis erzielen kann. Die Gesamtkosten dieser Kombination sind zwar im Schnitt die besten, jedoch selten mit großem Abstand gegenüber den gewichteten Kriterien und meistens müssen dafür schlechtere Erfolgsquoten hingenommen werden.

In jeder der untersuchten Datenreihen liefert ein Lösungsweg mit einem der gewichteten Kriterien die beste Erfolgsquote. Generell lassen sich beide kritisierten Effekte von Kosten- und Pufferkriterium auch beim gewichteten Kriterium erkennen. So werden die Lösungswege, welche nicht die Einfügereihenfolge 1 verwenden mit stärkerer Gewichtung der Kosten zunehmend inkompatibler mit dem gewichteten Kriterium. Gleichzeitig reduzieren sich mit dem Einfluss der Puffer jedoch auch die Gesamtkosten. Da dieser Effekt in sehr hohem Maße von der gesetzten Obergrenze abhängt, ist die Güte der mit dem gewichteten Kriterium erzielten Gesamtkosten nicht pauschal beurteilbar. So sind die Gesamtkosten bei einer Gewichtung von 1 zu 8 zum Teil besser, zum Teil jedoch auch deutlich schlechter als die vom Kostenkriterium erzielten. Während unter Verwendung von Limitsystematik 6 aufgrund dessen niedrigen Obergrenzen meistens eins der gewichteten Kriterien die besten Ergebnisse lieferte, werden bei höheren Obergrenzen noch unnötig hohe Gesamtkosten generiert.

Folglich liegt der Schlüssel zum Erhalt einer idealen Lösung in der Wahl der Gewichtung. Da bei hohen Obergrenzen (insbesondere denen von Limitsystematik 1 bei 100 Orten) eine Gewichtung von 1:8 zugunsten der Kosten nicht auszureichen scheint um die pufferbedingte Ineffizienz auszugleichen, muss diese weiter zugunsten der Kosten angepasst werden. Aus dieser Maßnahme ergibt sich jedoch auch, dass dann nur die an den Limits orientierte Einfügereihenfolge 1 verwendet werden kann. Wird eine angemessene Gewichtung gefunden, so ist es zu erhoffen, dass durch diese Kombination sehr gute Ergebnisse erzielt werden können.

Leider ist es an dieser Stelle extrem schwer, eine konkrete Empfehlung für die richtige Wahl der Gewichtung zu geben. Orientieren könnte sich diese an den Gesamtkosten der optimalen Lösung, sofern grob abgeschätzt werden kann, in welcher Größenordnung sich diese bewegt. Aus der Relation dieser zu den Limits könnte dann ein Schluss hinsichtlich einer angemessenen Gewichtung gezogen werden. Die Empfehlung, mit zunehmenden Limits die Gewichtung zugunsten der Kosten zu verändern, lässt sich auch in die umgekehrte

Richtung geben. So zeigten die besonders schwierigen Probleme mit zentraler Ballung tendenziell bessere Ergebnisse bei den ausgeglicheneren Gewichtungen.

Neben der limitorientierten Einfügereihenfolge ist auch noch die Reihenfolge „maximale Entfernung zur Tour“ hervorzuheben, welche besonders bei sehr schwierigen Problemen die besten (wenn auch nicht unbedingt gute) Erfolgsquoten aufweisen konnte.

Neben den Limits muss vor allem auch die Standortanzahl als maßgeblicher Faktor für die Ergebnisfindung aufgeführt werden. So lässt sich ein Problem mit 20 relativ gut mit verschiedenen Ansätzen lösen. Mit zunehmender Anzahl der Standorte werden die Unterschiede zwischen den „guten“ und „schlechten“ Lösungswegen deutlicher. Während sich steigende Komplexität der Probleme zwar generell in sinkenden Erfolgsquoten bemerkbar macht, trifft dies besonders stark auf die ohnehin „schwächeren“ Lösungswege zu.

Wenn anhand der durchgeführten Messreihen ein einzelner Lösungsweg unabhängig von den Rahmenbedingungen als Empfehlung gegeben werden müsste, so würde sich der Autor für die nach aufsteigenden Limits sortierte Einfügereihenfolge 1 in Verbindung mit der Gewichtung von 1 zu 4 zugunsten der Kosten entscheiden. Während diese Variante bei sehr hohen Obergrenzen zwar Gesamtkosten liefert, welche deutlich höher sind als notwendig, so haben sich besonders die Erfolgsquoten dieses Lösungsweges als besonders verlässlich erwiesen. Diese gehörten in jedem getesteten Fall zu den besten und dies im Gegensatz zur Gewichtung von 1 zu 8 auch bei den schwieriger lösbaren Problemen. Da der Autor dem Erfolg des Verfahrens eine höhere Bedeutung zuspricht als den Gesamtkosten, fällt die Wahl auf die Gewichtung von 1 zu 4. Die Empfehlung setzt sich allerdings auch dahin fort, diese Wahl unter Berücksichtigung der Rahmenbedingungen zu hinterfragen und die Gewichtung ggf. anzupassen.

7. Implementierung des Verfahrens

In diesem Teil wird die Implementierung von Problemgenerierung und –Lösung innerhalb von Visual Basic erklärt. Dazu werden zunächst die verwendeten Funktionen erklärt und ein Verzeichnis mit den wichtigsten Variablen aufgeführt.

Die Erläuterung des eigentlichen Programmcodes beschränkt sich auf die wesentlichen Bestandteile zur Generierung der zufälligen Probleme sowie zur Durchführung der Lösungsmethode. Demnach werden die Teile, die der Ausgabe der Daten oder der Formatierung der Arbeitsoberfläche dienen, hier nicht weiter erläutert. Auch einige vorbereitende Schritte, welche zum Beispiel Variablen zurücksetzten, werden nicht in die Erklärung einbezogen.

7.1 Erklärung der verwendeten Funktionen:

- **Dim** Variablenname **As** Variablenart
 - Hierdurch wird eine Variable für spätere Verwendung deklariert. Die Variablenart gibt zum Beispiel an, ob die Variable eine Zahl, einen Text oder einen Wahrheitswert annehmen kann. Im Fall einer numerischen Variablen werden unter anderem auch die Anzahl Nachkommastellen vorgeben.
 - Wird dem Variablennamen eine Nummer angehängt, so wird die Variable zu einem **Array**, welcher mithilfe eines Index mehrere Werte speichern kann
 - Beispiel 1: **Standortanzahl** kann nur einen Wert abspeichern
 - Beispiel 2: **Limits(999)** ist ein eindimensionaler Array und kann bis zu 999 Werte abspeichern.
 - Beispiel 3: **Kostenmatrix(999, 999)** ist ein zweidimensionaler Array und kann bis zu $999 * 999$ Werte speichern.
- **For** **Zählvariable** = a **To** b
Vorgang, welcher (b – a) mal ausgeführt wird
Next
 - Diese Funktion bildet eine Schleife, welche solange wiederholt wird, bis die Zählvariable den Wert von b angenommen hat. Beginnend beim Startwert a zählt die Zählvariable in jedem Schritt um 1 hoch. Alle Vorgänge welche zwischen **For** und **Next** aufgeführt sind, werden in jedem Durchlauf der Schleife durchgeführt. Die Zählvariable kann, sofern sie nicht separat deklariert wurde, nur innerhalb der Schleife verwendet werden.
- **If** Bedingung 1
Vorgang, welcher durchgeführt werden soll, wenn Bedingung 1 erfüllt ist
ElseIf Bedingung 2
Vorgang, welcher durchgeführt werden soll, wenn Bedingung 2 erfüllt ist
Else
Vorgang, welcher durchgeführt werden soll, wenn keine der Bedingungen erfüllt ist
End If
 - Mit der **If**-Funktion kann die Durchführung von Vorgängen an eine Bedingung geknüpft werden. Der Vorgang wird nur durchgeführt, wenn die Bedingung erfüllt ist. Wenn die Bedingung nicht erfüllt ist, können mit **ElseIf** beliebig viele zusätzliche Bedingungen und deren Konsequenzen angereicht werden.
 - Die Bedingungen werden nacheinander geprüft und sobald die erste Bedingung erfüllt ist, werden keine weiteren Bedingungen geprüft. Die nach **Else** aufgeführten Vorgänge werden nur durchgeführt, wenn keine Bedingung erfüllt wurde.
 - **End If** gibt das Ende der **If**-Funktion an.

- **Private Sub** Name des Teilprogramms
Vorgänge des Teilprogramms
End Sub
 - Durch diese Funktion wird ein Teilprogramm deklariert, welches an anderer Stelle abgerufen werden kann. Dies dient primär der Übersichtlichkeit, besonders wenn eine Gruppe von Vorgängen an mehreren Stellen immer wieder durchgeführt wird. Alles was zwischen **Private Sub** und **End Sub** aufgeführt ist, gehört zum Teilprogramm
 - Wird an einer anderen Stelle auf das Teilprogramm verwiesen, so werden alle Vorgänge des Teilprogramms einmal durchgeführt.

7.2 Variablen in Visual Basic:

Im Folgenden sind alle für die Erklärung des Programmcodes relevanten Variablen aufgeführt. In der zweiten Spalte der Tabelle finden sich - sofern vorhanden - die Äquivalente aus den Teilen 4 und 5 aufgeführt.

Generierung der Kostenmatrix		
Kostenmatrix(999,999)	k (a, b)	Kostenmatrix(a,b) enthält die Strecke (Kosten) zwischen a und b
Standortanzahl	m	Anzahl der Orte ohne den Startpunkt
Startpunktlage	-	Beschreibt, auf welche Weise der Startpunkt generiert wird
xvar(999)	x _a	xvar(a) ist die x-Koordinate von a
yvar(999)	y _a	yvar(a) nimmt den Wert der y-Koordinate von Punkt a an
Nur für die Generierung der "Städtekarte"		
Städteanzahl	-	Anzahl der Städte
Stadtwahl	-	Für Stadtwahl wird eine Zufallszahl generiert, welche bestimmt, welcher Stadt ein Ort zugewiesen wird
Stadtxvar (Städteanzahl+1)	-	Stadtxvar(a) beschreibt die x-Koordinate einer Stadt a
Stadtyvar (Städteanzahl+1)	-	Stadtyvar(a) ist die dazugehörige y-Koordinate von a
Einfügereihenfolge		
Einfügereihenfolge (999)	P = (p ₁ , p ₂ , ..., p _m)	Einfügereihenfolge(a) enthält den Punkt, der als a-tes eingefügt wird
Folgesortiert (Standortanzahl)	-	Dienen der Generierung der Zufälligen Einfügereihenfolge
Zufallsfolge (Standortanzahl)	-	
Limitgenerierung		
Limits(999)	L(a)	Limits(a) enthält das Limit von Punkt a
Matrixgröße	G _m	Summe aller mittleren Entfernungen von einem Punkt zu allen anderen
LowerBound	L _{min}	Höhe von unterstem und oberstem Limit
UpperBound	L _{max}	

Durchführung der Methode		
Einfügekosten(999,999)	E	Einfügekosten(a, b) beschreibt die Zusatzkosten, welche durch Einfügen an Position b in Iteration a entstehen würden
Ergebnis	K_{ges}	Gesamtkosten nach Abschluss des Verfahrens
Gesamtpuffer (Standortanzahl)	D_{zu}	Minimum aus Limitselbstpuffer und Limitrestpuffer; Entscheidungskriterium beim Pufferkriterium
Gewichtskosten	g_{kost}	Für gewichtetes Kriterium: gibt die Gewichtung der Kosten an (Entscheidend ist das Verhältnis zu Gewichtpuffer)
Gewichtpuffer	g_{puff}	Gewichtung des verbleibenden Puffers für die Wahl der Einfügeposition
Gkriterium(s)	$G(z)$	Entscheidungskriterium beim gewichteten Kriterium
Index	Z_{opt}	Einfügeposition, welche durch das Entscheidungskriterium ausgewählt wird
Kumkost(999,999)	$K_s(i_s;j)$	Kumkost(a, b) beschreibt die kumulierten Kosten beim b-ten Punkt der Teiltour nach Iteration a
Limitrestpuffer(999)	D_{min}	Minimaler Puffer von einem der nachfolgenden Punkte nach einem Einfügevorgang
Limitselbstpuffer(999)	D_p	Verbleibender Puffer am einzufügenden Punkt nach dem Einfügevorgang
MaxPuffer	D_{opt}	Maximaler Gesamtpuffer (Vergleich zwischen allen Einfügepositionen)
MinKost	E_{opt}	Minimum der ungesperrten Einfügekosten; Entscheidungskriterium beim Kostenkriterium
Restumweg(999,999)	$R_s(i_s;j)$	Restumweg(a, b) beschreibt den Puffer, welcher nach Iteration a am b-ten Punkt der Teiltour noch verfügbar ist
Teiltouren(999,999)	$i_s;j$	Teiltouren(a, b) enthält den Punkt, der nach Iteration a an Stelle b der Tour steht
Verspätung	V_{ges}	Summe aller Limitüberschreitungen (Das Ergebnis ist zulässig, wenn die Verspätung=0 ist)
Hilfsvariablen		
Arbeitseinfügekosten (Standortanzahl)	-	Eindimensionale Hilfsvariablen zur Bestimmung vom Minimum; Werden in jeder Iteration überschrieben
Arbeitsrestumweg (Standortanzahl)	-	
Arbeitsindex	-	Index für die Ermittlung von Einfügereihenfolgen, welcher in jedem Schritt überschrieben wird
Ausschluss	-	Wahrheitswert; wird verwendet zur Bestimmung der "Max. Entfernung zur Tour"- Einfügereihenfolge
Hilfsarray1(999)	-	Werden als Hilfsvariablen bei der "Max. Entfernung zur Tour"- Einfügereihenfolge verwendet
Hilfsarray2(999)	-	
Mittelwertarray(999)	-	Zur Bestimmung von Mittelwerten verwendet
Sortierarray(999)	-	an verschiedenen Stellen zur Sortierung verwendet
Zähler	-	Wird an verschiedenen Stellen genutzt um manuell hochzuzählen

Zählvariablen (werden nur innerhalb von Schleifen verwendet)		
i	-	Werden verwendet, wenn keine der konkreteren Variablen zutrifft. Meistens zählen sie die Standorte durch.
j	-	
pos	z	Zählvariable, welche für die Einfügeposition steht
s	s	Zählvariable, welche für den Iterationsschritt steht
Schritt	-	Zählt die Positionen der Einfügereihenfolge bei der "Größte Entfernung zur bestehenden Tour"-Reihenfolge
Stadt	-	Zählt die "Städte" der "Städtekte"-Generierung

7.3 Generierung von Problemen:

7.3.1 Generierung der Koordinaten

Generierung des Startpunktes:

```

Startpunktlage = Startpunkt.Text

If Startpunktlage = "Zentral" Then
    xvar(0) = 50
    yvar(0) = 50
ElseIf Startpunktlage = "Dezentral" Then
    xvar(0) = 0
    yvar(0) = 0
ElseIf Startpunktlage = "Zufällig" Then
    Randomize()
    yvar(0) = Math.Round(Rnd() * 100, 3)
    Randomize()
    xvar(0) = Math.Round(Rnd() * 100, 3)
End If

```

Die Generierung der Koordinaten beginnt mit dem Startpunkt. Der Variablen **Startpunktlage** wird zunächst der in der Arbeitsoberfläche ausgewählte Wert zugewiesen. In Abhängigkeit davon werden die x- und y-Koordinaten des Startpunktes 0 bestimmt. Bei der zufälligen Lage wird durch *Rnd()* eine Zufallszahl zwischen 0 und 1 generiert, welche anschließend mit 100 multipliziert wird und zuletzt durch *Math.Round()* auf 3 Stellen gerundet wird. Da die generierte Zahl nur eine Pseudozufallszahl ist, wird die Funktion *Randomize()* verwendet, um bei einem Neustart neue Zufallszahlen zu erhalten. Die Koordinaten für die restlichen Orte werden in Abhängigkeit der gewählten Koordinatenverteilung in unterschiedlichen Verfahren generiert:

Zufällige Verteilung:

```
For i = 1 To Standortanzahl
    Randomize()
    yvar(i) = Math.Round(Rnd() * 100, 3)
    Randomize()
    xvar(i) = Math.Round(Rnd() * 100, 3)
Next
```

Bei allen Verteilungen werden Schleifen mit der Zählvariable **i** verwendet, welche, bei 1 beginnend, nach jedem Durchgang hochzählt bis sie den Endwert (hier die Variable **Standortanzahl**) erreicht. Die Koordinaten werden hier mit der gleichen Formel wie der zufällige Startpunkt generiert und erhalten demnach einen Wert zwischen 0 und 100.

Zufällige Verteilung im gestreckten Bereich:

```
For i = 0 To Standortanzahl
    Randomize()
    xvar(i) = Math.Round(Rnd() * 50, 3)
    Randomize()
    yvar(i) = Math.Round(Rnd() * 200, 3)
Next
```

Die Generierung der Koordinaten im gestreckten Bereich erfolgt analog zur zufälligen Generierung. Die x-Koordinate erhält jedoch einen Wert zwischen 0 und 50, während die y-Koordinate einen Wert zwischen 0 und 200 annimmt.

Zentrale Ballung:

```
For i = 1 To Standortanzahl
    Randomize()
    yvar(i) = Math.Round((Rnd() + Rnd() + Rnd() / 2) * 40, 3)
    Randomize()
    xvar(i) = Math.Round((Rnd() + Rnd() + Rnd() / 2) * 40, 3)
Next
```

Um eine zentrale Ballung zu erzielen werden hier zunächst zwei Zufallszahlen zwischen 0 und 1 und eine Zufallszahl zwischen 0 und 0,5 addiert. Die dadurch resultierende Zufallszahl zwischen 0 und 2,5 wird anschließend mit 40 multipliziert und auf drei Stellen gerundet. Die resultierenden Koordinaten liegen somit ebenfalls zwischen 0 und 100, jedoch mit erhöhter Wahrscheinlichkeit zentral.

Städtecluster:

```
Städteanzahl = 10
Zähler = 0
For Stadt = 1 To Städteanzahl
  If Zähler = 4 Then
    Zähler = 0
  End If
  Zähler = Zähler + 1
  If Zähler = 1 Then
    Stadtxvar(Stadt) = Math.Round(Rnd() * 42, 3) + 8
    Stadtyvar(Stadt) = Math.Round(Rnd() * 42, 3) + 8
  ElseIf Zähler = 2 Then
    Stadtxvar(Stadt) = Math.Round(Rnd() * 42, 3) + 50
    Stadtyvar(Stadt) = Math.Round(Rnd() * 42, 3) + 8
  ElseIf Zähler = 3 Then
    Stadtxvar(Stadt) = Math.Round(Rnd() * 42, 3) + 8
    Stadtyvar(Stadt) = Math.Round(Rnd() * 42, 3) + 50
  ElseIf Zähler = 4 Then
    Stadtxvar(Stadt) = Math.Round(Rnd() * 42, 3) + 50
    Stadtyvar(Stadt) = Math.Round(Rnd() * 42, 3) + 50
  End If
Next

For i = 1 To Standortanzahl
  Randomize()
  Stadtwahl = Math.Ceiling(Rnd() * Städteanzahl)
  Randomize()
  yvar(i) = Math.Round(Stadtyvar(Stadtwahl) + (Rnd() + Rnd() - 1) * 8, 3)
  Randomize()
  xvar(i) = Math.Round(Stadtxvar(Stadtwahl) + (Rnd() + Rnd() - 1) * 8, 3)
Next
```

Die Generierung der Koordinaten anhand von Städteclustern ist erheblich komplexer als die anderen Verfahren. Zunächst wird hier die Anzahl der Städte – also die maximale Anzahl der Cluster – als 10 festgelegt und die Variable **Zähler** wird auf 0 gesetzt. In einer ersten Schleife mit der Zählvariable **Stadt** werden die Koordinaten der 10 Städte (**Stadtxvar** und **Stadtyvar**) ermittelt. Da die Cluster nicht alle im gleichen Bereich generiert werden sollen, werden sie auf die vier Quadranten⁷⁵ der Ebene gleichmäßig aufgeteilt. Dazu wird zu Beginn der Schleife abgefragt, ob die Variable **Zähler** den Wert 4 erreicht hat. Ist dies der Fall, wird sie auf 0 gesetzt. Nach der Abfrage wird der Wert von **Zähler** um 1 erhöht. Abhängig davon, welchen Wert **Zähler** nun hat, werden die x- und y-Koordinaten innerhalb von einem der Quadranten ermittelt. Sie erhalten Zufallswerte zwischen 8 und 50 oder zwischen 50 und 92, wodurch die Koordinaten der eigentlichen Punkte im zweiten Schritt nicht außerhalb der 100x100 Einheiten großen Ebene generiert werden können.

In der zweiten Schleife wird zunächst eine Zufallszahl zwischen 1 und der **Städteanzahl** (hier als 10 festgelegt) gebildet und diese dann aufgerundet. Diese Zahl (die Variable **Stadtwahl**) gibt an, zu welcher der Städte der aktuell zu generierende Ort zugeordnet wird. Die Koordinaten des Ortes werden anschließend auf Basis der entsprechenden Koordinaten der

⁷⁵ Mit einem Quadranten ist hier je ein 50x50 Einheiten großer Bereich, also ein Viertel der Gesamtebene gemeint.

Stadt generiert. Dazu werden zunächst zwei Zufallszahlen zwischen 0 und 1 addiert, von denen dann 1 subtrahiert wird. Somit ergibt sich eine Zufallszahl zwischen -1 und 1, welche mit erhöhter Wahrscheinlichkeit einen Wert um 0 annimmt. Der Wert wird anschließend mit 8 multipliziert, gerundet und zuletzt zu der entsprechenden Koordinate der Stadt addiert. Somit werden die Punkte in einem Bereich von 8 Einheiten um die Koordinaten einer zufälligen Stadt generiert. So können Cluster mit mehr oder weniger Orten entstehen. Besonders bei einer Messreihe mit wenigen Orten, kann es sein, dass nicht allen Städten ein Punkt zugeordnet wird.

7.3.2 Generierung der Kostenmatrix

```
For i = 0 To Standortanzahl
  For j = 0 To Standortanzahl
    If (i = j) Then
      Kostenmatrix(i, j) = 9999
    Else
      Kostenmatrix(i, j) = Math.Round(Math.Sqrt((xvar(i) - xvar(j)) ^ 2
+ (yvar(i) - yvar(j)) ^ 2), 3)
    End If
  Next
Next
```

Aus den generierten Koordinaten der Punkte wird in diesem Schritt die Kostenmatrix errechnet. Die Werte werden im zweidimensionalen Array **Kostenmatrix** gespeichert. Dazu werden zwei ineinander geschachtelte Schleifen durchlaufen, beide so oft, wie es Standorte (inklusive des Startpunkts) gibt. In Abhängigkeit von den Zählvariablen der beiden Schleifen (i und j) wird bei jedem Durchlauf der inneren Schleife mithilfe des *Satz des Pythagoras* ein Wert der Kostenmatrix ermittelt. Die Zählvariable i der äußeren Schleife gibt dabei den Ausgangsort und die Zählvariable j der inneren Schleife das Ziel der aktuell berechneten Strecke vor. Bei i=0 und j=1 werden beispielsweise die Kosten für die Fahrt vom Startpunkt 0 zum Punkt 1 berechnet. Diese werden dann in Kostenmatrix unter den Indizes (0,1) gespeichert.

Für die konkrete Berechnung werden die Differenzen zwischen den x- und y-Koordinaten der beiden Punkte jeweils quadriert und anschließend addiert. Die Wurzel davon ergibt die Strecke zwischen den beiden Punkten und somit den Eintrag in der Kostenmatrix. Im Sonderfall, dass die Zählvariablen den gleichen Wert haben, muss der Wert in der Kostenmatrix gesperrt werden, weil dies die unzulässige Strecke eines Punktes zu sich selbst bedeutet. Die Sperrung geschieht, indem anstelle der üblichen Berechnung der Wert 9999 zugewiesen wird.

7.3.3 Bestimmung der Limits

```
'--Matrixgröße bestimmen-----'  
  
For i = 1 To Standortanzahl  
    For j = 0 To Standortanzahl  
        Mittelwertarray(j) = Kostenmatrix(i, j)  
    Next  
    Sortierarray(i) = (Mittelwertarray.Sum - 9999) / Standortanzahl  
Next  
Matrixgröße = Sortierarray.Sum
```

Vorbereitend für die Bestimmung der Limits wird hier die Variable **Matrixgröße** errechnet, welche als Indikator dafür dienen soll, wie nah die Punkte beieinander liegen und in welcher Größenordnung die Gesamtkosten als Konsequenz liegen werden. Da von zweidimensionalen Arrays nicht ohne weiteres Mittelwerte oder Summen innerhalb einer Zeile oder Spalte gebildet werden können, werden hier zwei eindimensionale Hilfsarrays **Mittelwertarray** und **Sortierarray** verwendet. In der inneren Schleife werden die Entfernungen eines Punktes zu allen anderen in den **Mittelwertarray** übertragen. Dessen Mittelwert wird anschließend innerhalb der äußeren Schleife im Sortierarray gespeichert. Bei der Errechnung des Mittelwerts wird die mit dem Wert 9999 gesperrte Position in der Kostenmatrix wieder herausgerechnet. Der Wert von **Matrixgröße** ergibt sich schließlich als Summe der in **Sortierarray** abgespeicherten Werte.

Anhand von Standortanzahl und Matrixgröße können die Ober- und Untergrenzen errechnet werden. Da sich die Formeln bei den Systematiken 4, 5 und 6 nur geringfügig unterscheiden, wird hier nur der Programmcode für die Generierung unter den Limitsystematiken 1 und 4 aufgeführt. Neben der Funktion zur Rundung der Ergebnisse (Math.Round) entspricht der Code des Weiteren den im Hauptteil 5.1.2 erklärten Formeln und wird daher nicht weiter erklärt.

Limitsystematik 1 (Unflexible Limits):

```
UpperBound = Math.Round(matrixgröße / 1.45)  
  
LowerBound = Math.Round(matrixgröße * (3.3 / Standortanzahl))
```

Limitsystematik 4 (Flexible Limits):

```
UpperBound = Math.Round((matrixgröße * (1.25 + 2 / Standortanzahl)) /  
(1.3 + Standortanzahl * 0.018))  
  
LowerBound = Math.Round((matrixgröße * (0.15 + 4 / Standortanzahl)) /  
(1.3 + Standortanzahl * 0.018))
```

Generierung der Limits aus Ober- und Untergrenzen:

```
'----Generierung der Limits anhand von Ober- und Untergrenze----  
For i = 1 To Standortanzahl  
    Limits(i) = Math.Round(LowerBound + ((UpperBound - LowerBound) /  
(Standortanzahl - 1)) * (i - 1))  
Next
```

Abschließend werden innerhalb dieser Schleife alle Limits in gleichmäßigen Abständen zwischen den Grenzen generiert. Dazu wird die Differenz zwischen Ober- und Untergrenze durch die Anzahl zu besuchender Orte (hier **Standortanzahl-1**) geteilt. Die Differenz beschreibt den Abstand zwischen den Limits. Sie wird mit einem von der Zählvariable abhängigen Faktor ($i - 1$) multipliziert und zur Untergrenze addiert. Dem ersten Punkt (Zählvariable $i=1$) wird somit das niedrigste Limit zugewiesen, welches der Untergrenze entspricht.⁷⁶ Bei jedem nachfolgenden Punkt erhöht sich der Faktor um 1, wodurch der Abstand einmal mehr zur Untergrenze addiert wird.

7.4 Durchführung der Methode

Die Erklärung des Programmtextes orientiert sich weitgehend an den in Teil 4.1 definierten Schritten. Auf die Unterschiede der Kriterien wird dabei ebenfalls erst an der einschlägigen Stelle eingegangen.

Schritt 0 und 1: Vorbereitung und Sortierung der Matrix: Kostenmatrix und Limits wurden hier bereits so generiert, dass sie auf die Durchführung der Methode vorbereitet sind. Demnach ist es nicht erforderlich, hier noch weitere Schritte vorzunehmen.

Schritt 2: Ermittlung der Einfügereihenfolge: Die Ermittlung der Einfügereihenfolgen wird auch hier gesondert in Teil 7.5 betrachtet. Als Ergebnis geht der Array **Einfügereihenfolge(Standortanzahl)** hervor, welcher in den nächsten Schritten verwendet wird.

Schritt 3: Vorbereitung der Iteration und erste Iteration: Abweichend von der Erläuterung in Teil 4.1 wird als Vorbereitung hier direkt der erste Iterationsschritt durchgeführt. Aufgrund dessen Besonderheiten und einfacher Durchführung wird dieser separiert.

```
'-----Erster Iterationsschritt:-----  
Teiltouren(1, 0) = 0  
Teiltouren(1, 1) = Einfügereihenfolge(1)  
Teiltouren(1, 2) = 0  
Kumkost(1, 1) = Kostenmatrix(Teiltouren(1, 0), Teiltouren(1, 1))  
Kumkost(1, 2) = Kumkost(1, 1) + Kostenmatrix(Teiltouren(1, 1), Teiltouren(1, 2))  
Restumweg(1, 1) = Limits(Einfügereihenfolge(1)) - Kumkost(1, 1)
```

Da es nur eine einzige Möglichkeit zum Einfügen gibt, werden der nullten bis zweiten Position der neuen Teiltour direkt Punkte zugeordnet. Start- (Position 0) und Endpunkt

⁷⁶ Der Abstand wird im ersten Schritt $(1-1)=0$ mal zur Untergrenze hinzu addiert.

(Position 2) sind immer Punkt 0. Der ersten Position wird der erste Punkt der Einfügereihenfolge zugewiesen. Anschließend werden die neuen kumulierten Kosten sowie der Restpuffer berechnet. Die kumulierten Kosten an der ersten Position der neuen (ersten) Teiltour entspricht der Strecke⁷⁷ vom Start zum zugeordneten Punkt. Dieser Wert wird aus der Kostenmatrix abgerufen. Zum errechneten Wert wird im nächsten Schritt die Strecke vom eingefügten Punkt zum Endpunkt hinzu addiert, wodurch die Gesamtkosten der Teiltour errechnet werden.

Vorbereitend wird im letzten Schritt der verbleibende Puffer (hier als Restumweg bezeichnet) am eingefügten Punkt errechnet. Dazu werden dessen Limit und aktuelle kumulierte Kosten abgefragt und voneinander subtrahiert.

Bei allen hierbei verwendeten zweidimensionalen Variablen spiegelt der erste Index immer den Iterationsschritt s wieder, während der zweite Index die Position in der neu gebildeten Tour beschreibt.

Schritt 4: Durchführung der Iteration in m Iterationsschritten⁷⁸: Ab dem zweiten Iterationsschritt wird der folgende, deutlich aufwendigere Vorgang durchgeführt. Die Zählvariable s gibt dabei immer den aktuellen Iterationsschritt an.

Schritt 4.1: Bestimmung der Einfügekosten:

```
'---Iterationsschritt - Bestimmung Einfügekosten
For pos = 1 To s
    Einfügekosten(s, pos) = Kostenmatrix(Teiltouren(s - 1, pos - 1),
    Einfügereihenfolge(s)) + Kostenmatrix(Einfügereihenfolge(s), Teiltouren(s - 1, pos)) -
    Kostenmatrix(Teiltouren(s - 1, pos - 1), Teiltouren(s - 1, pos))

    Arbeitseinfügekosten(pos) = Einfügekosten(s, pos)
Next
```

In diesem Schritt wird eine Schleife durchgeführt, welche die Zählvariable **pos** als Repräsentant für die Einfügeposition verwendet. Die Schleife wird s mal durchgeführt, also zweimal in Iteration 2, dreimal in Iteration 3, und so weiter. Die Einfügekosten werden unter Verwendung der in Teil 4.1 beschriebenen Formel errechnet, das heißt die beiden hinzukommenden Strecken werden addiert und die ersetzte Strecke wird subtrahiert. Die drei Bestandteile der Formel erklären sich wie folgt:

⁷⁷ Wie in Teil 3 erläutert werden auch hier Kosten und Strecken gleichbedeutend verwendet. Die Strecke entspricht somit den Kosten für die Fahrt.

⁷⁸ Da der erste Schritt separat vorgenommen wird, werden nur $m-1$ „volle“ Iterationsschritte durchgeführt

- `Kostenmatrix(Teiltouren(s - 1, pos - 1), Einfügereihenfolge(s))`
 - Entspricht der Strecke vom vor der Einfügeposition liegenden Punkt zum einzufügenden Punkt
- `Kostenmatrix(Einfügereihenfolge(s), Teiltouren(s - 1, pos))`
 - Entspricht der Strecke vom einzufügenden Punkt zum nachfolgenden Punkt (also dem Punkt, welcher vorher an dieser Position war)
- `Kostenmatrix(Teiltouren(s - 1, pos - 1), Teiltouren(s - 1, pos))`
 - Entspricht der Strecke zwischen den beiden Punkten zwischen denen die Einfügeposition liegt

Um in nachfolgenden Schritten leichter mit den Werten arbeiten zu können, werden die Werte zusätzlich in den eindimensionalen Array **Arbeitseinfügekosten** übertragen. Dieser wird im Gegensatz zum zweidimensionalen Array **Einfügekosten** im nächsten Iterationsschritt überschrieben.

Schritt 4.2: Zulässigkeitskontrolle:

```

'---Iterationsschritt - Zulässigkeitskontrolle-----
'--Puffer für einzufügenden Punkt und nachfolgende Punkte bestimmen
For pos = 1 To s
    Limitselbstpuffer(pos) = Limits(Einfügereihenfolge(s)) - Kumkost(s -
1, pos - 1) - Kostenmatrix(Teiltouren(s - 1, pos - 1), Einfügereihenfolge(s))

    '--Arbeitsrestumweg neu bestimmen
    For i = pos To s - 1
        Arbeitsrestumweg(i) = Restumweg(s - 1, i)
    Next
    Limitrestpuffer(pos) = Arbeitsrestumweg.Min - Einfügekosten(s, pos)
    '--Der geringere Puffer wird gewählt
    If Limitselbstpuffer(pos) < Limitrestpuffer(pos) Then
        Gesamtpuffer(pos) = Limitselbstpuffer(pos)
    Else
        Gesamtpuffer(pos) = Limitrestpuffer(pos)
    End If
Next

```

Wie im letzten Schritt wird auch hier eine Schleife mit **pos** als Zählvariable *s* mal durchlaufen. Auch hier werden die in Teil 4.1 angegebenen Formeln zur Errechnung der benötigten Werte verwendet. Zunächst wird mit **Limitselbstpuffer(pos)** der verbleibende Puffer des einzufügenden Punktes bei dieser Position errechnet. Er ermittelt sich aus dem Limit des einzufügenden Punktes abzüglich der kumulierten Kosten am vorherigen Punkt und der Strecke von diesem zum einzufügenden Punkt.

Für die Bestimmung der verbleibenden Puffer an den nachfolgenden Orten wird deren aktuell verbleibender Puffer (hier **Restumweg**) zunächst in die Arbeitsvariable **Arbeitsrestumweg** überschrieben. Dabei werden nur die Puffer von jenen Punkten überschrieben, welche hinter der Einfügeposition **pos** liegen.

Aus den übertragenen Puffern wird anschließend das Minimum ermittelt, von welchem die Einfügekosten der entsprechenden Position subtrahiert werden. Als Differenz ergibt sich der geringste Puffer, welcher an den nachfolgenden Orten durch den Einfügevorgang entstehen würde. Hier wird dieser unter **Limitrestpuffer(pos)** gespeichert. In der nachfolgenden Abfrage wird aus **Limitselbstpuffer(pos)** und **Limitrestpuffer(pos)** der geringere ausgewählt und dessen Wert **Gesamtpuffer(pos)** zugewiesen.

Schritt 4.3: Bestimmung des Entscheidungskriteriums:

Schritt 4.3 - Kostenkriterium:

```
'---Iterationsschritt - Auswahl der Einfügeposition - Kostenkriterium -----'  
  
'--Prüfung, ob ein zulässiger Wert vorliegt; ggf. Blockierung unzulässiger  
maxPuffer = Gesamtpuffer.Max  
If maxPuffer < 0 Then  
Else  
    For pos = 1 To s  
        If Gesamtpuffer(pos) < 0 Then  
            Arbeitseinfügekosten(pos) = 9999  
        End If  
    Next  
End If  
  
'--Minimum der zulässigen Einfügekosten bestimmen und Index herausfinden  
  
minKost = Arbeitseinfügekosten.Min  
  
Index = Array.IndexOf(Arbeitseinfügekosten, minKost)
```

Zunächst wird hier geprüft, ob überhaupt ein zulässiger Wert vorliegt. Dafür wird der Variable **maxPuffer** der Maximalwert des Arrays **Gesamtpuffer** zugewiesen. Letzterer enthält die in der Zulässigkeitskontrolle errechneten Minimalpuffer für jede Einfügeposition. In der anschließenden Abfrage wird geprüft, ob der **maxPuffer** zugewiesene Wert kleiner als 0 ist. Ist dies der Fall, so führt jede Einfügeposition zu einem unzulässigen Ergebnis. Da das Verfahren auch trotz Scheiterns weiter geführt wird, erfolgt keine Sperrung von Einfügepositionen.

Sofern die Bedingung nicht erfüllt ist, gibt es mindestens eine zulässige Einfügeposition. Demnach werden dann innerhalb einer Schleife alle Einfügepositionen durchgegangen und jene gesperrt, bei denen der **Gesamtpuffer(pos)** negativ ist. Die Sperrung erfolgt indem die Variable **Arbeitseinfügekosten(pos)** auf den Wert 9999 gesetzt wird.

Für die Wahl der Einfügeposition wird anschließend lediglich das Minimum des Arrays **Arbeitseinfügekosten** bestimmt und der Variable **minKost** zugewiesen. Von diesem Wert wird im letzten Schritt der Index abgefragt, welcher der Einfügeposition entspricht. Dieser wird der Variable **Index** zugewiesen.

Schritt 4.3 - Pufferkriterium:

```
'---Iterationsschritt - Auswahl der Einfügeposition - Pufferkriterium -----  
bietet  
'--Prüfung, welche Position den größten verbleibenden (Gesamt-) Puffer  
Index = Array.IndexOf(Gesamtpuffer, Gesamtpuffer.Max)
```

Die für die Wahl der Einfügeposition benötigten Voraussetzungen sind bereits aus der Zulässigkeitskontrolle hervor gegangen. Die Entscheidung wird basiert auf der Variable **Gesamtpuffer**. Von dieser wird der Maximalwert bestimmt und dessen Index abgerufen. Die resultierende Einfügeposition wird auch hier in der Variable **Index** abgespeichert.

Schritt 4.3 - Gewichtetes Kriterium:

```
'---Iterationsschritt - Auswahl der Einfügeposition - gew. Kriterium -----  
'--Prüfung, ob ein zulässiger Wert vorliegt; ggf. Blockierung unzulässiger  
maxPuffer = Gesamtpuffer.Max  
If maxPuffer < 0 Then  
Else  
    For pos = 1 To s  
        If Gesamtpuffer(pos) < 0 Then  
            Arbeitseinfügekosten(pos) = 9999  
        End If  
    Next  
End If  
  
'--Kriterium zur Auswahl der Einfügeposition bestimmen  
Gkriterium(0) = -9999  
minKost = Arbeitseinfügekosten.Min  
  
For pos = 1 To s  
    Gkriterium(pos) = (Gesamtpuffer(pos) - maxPuffer) * Gewichtpuffer +  
(minKost - Arbeitseinfügekosten(pos)) * Gewichtkosten  
Next  
Index = Array.IndexOf(Gkriterium, Gkriterium.Max)
```

Auch beim gewichteten Kriterium wird zunächst geprüft, ob überhaupt ein zulässiger Wert vorliegt. Dieser Schritt ist identisch zum Kostenkriterium.

Für die Wahl der Einfügeposition wird hier die Variable **Gkriterium** herangezogen. Da es keine nullte Einfügeposition gibt, wird der nullte Wert dieser Variable zunächst durch den Wert -9999 gesperrt. Anschließend wird wie beim Kostenkriterium das Minimum der Einfügekosten ermittelt und der Variable **minKost** zugewiesen. Anschließend wird für jede Einfügeposition das Entscheidungskriterium **Gkriterium(pos)** ermittelt. Es ergibt sich als Summe aus den gewichteten Differenzen zum Optimum von Kosten und Puffern:

- $(\text{Gesamtpuffer}(\text{pos}) - \text{maxPuffer}) * \text{Gewichtpuffer}$
 - Differenz aus Gesamtpuffer der Einfügeposition und dem optimalen (maximalen) Gesamtpuffer, gewichtet durch die Variable **Gewichtpuffer**
- $(\text{minKost} - \text{Arbeitseinfügekosten}(\text{pos})) * \text{Gewichtkosten}$
 - Differenz aus den minimalen Einfügekosten und denen der Position, gewichtet durch die Variable **Gewichtkosten**

Im letzten Schritt wird analog zu den anderen Kriterien anhand des Maximalen Werts von **Gkriterium** die Einfügeposition ermittelt und der Variable **Index** zugewiesen.

Schritt 4.4 und 4.5: Bildung der neuen Tour und Vorbereitung der nächsten Iteration:

```

'---Iterationsschritt - neue Teiltour bestimmen -----
Teiltouren(s, 0) = 0
Teiltouren(s, s + 1) = 0
Teiltouren(s, Index) = Einfügereihenfolge(s)
For i = 1 To Index - 1
    Teiltouren(s, i) = Teiltouren(s - 1, i)
Next
For i = Index + 1 To s
    Teiltouren(s, i) = Teiltouren(s - 1, i - 1)
Next

'--Neuen kum. Weg und Restumweg bestimmen (Vorbereitung nächster Iteration)
For i = 1 To s + 1
    Kumkost(s, i) = Kumkost(s, i - 1) + Kostenmatrix(Teiltouren(s, i - 1),
Teiltouren(s, i))
    Restumweg(s, i) = Limits(Teiltouren(s, i)) - Kumkost(s, i)
Next

```

Anhand der nach den drei Kriterien ermittelten Variable **Index** wird in diesem Schritt die neue Teiltour gebildet. Wie auch im ersten Iterationsschritt wird Start- und Endpunkt Punkt 0 zugewiesen. Dem einzufügenden Punkt wird anschließend die Position zugewiesen, welche der gewählten Einfügeposition (also **Index**) entspricht. Wurde zum Beispiel die erste Einfügeposition gewählt, so wird der Punkt vor dem vorher an erster Position stehenden Punkt eingefügt, nimmt also die neue erste Position an.

Alle restlichen Positionen werden anhand der vorherigen Teiltour (der in Iterationsschritt s-1 generierten) bestimmt. Alle Positionen vor der Einfügeposition werden innerhalb der ersten Schleife direkt aus der letzten Teiltour übernommen. Die verbleibenden Punkte, welche in der vorherigen Teiltour hinter der Einfügeposition liegen, werden in der zweiten Schleife um eine Position nach hinten verschoben der neuen Tour zugeordnet.

Nachdem die Bildung der Tour abgeschlossen ist, werden vorbereitend erneut die kumulierten Kosten errechnet. Dies geschieht erneut innerhalb einer Schleife, welche (s+1) mal durchlaufen wird. In jedem Durchlauf werden die kumulierten Kosten an der der Zählvariable entsprechenden Position ermittelt, indem dem Wert der vorherigen Position die hinzu kommenden Kosten addiert werden. Analog zum ersten Iterationsschritt wird final für

jeden Punkt der Teiltour (außer dem Start- und Endpunkt) der verbleibende Puffer (hier **Restumweg**) berechnet.

Bestimmung der Endergebnisse nach Iterationsschritt m:

```
'-----Bestimmung des Ergebnisses-----  
  
Ergebnis = Kumkost(Standortanzahl, Standortanzahl + 1)  
  
Verspätung = 0  
For i = 1 To Standortanzahl  
    If Restumweg(Standortanzahl, i) < 0 Then  
        Verspätung = Verspätung - Restumweg(Standortanzahl, i)  
    End If  
Next
```

Nach Ende der Iteration werden hier noch die Endergebnisse ermittelt. Die Gesamtkosten (hier **Ergebnis**) entsprechen den kumulierten Kosten (**Kumkost**) am Endpunkt der letzten Teiltour, welche der finalen Tour entspricht. Die Verspätung, welche ausschlaggebend dafür ist, ob das Ergebnis zulässig ist, wird errechnet indem die finalen Puffer (hier **Restumweg**) von allen Punkten überprüft werden. Dafür wird die Variable **Verspätung** zunächst gleich Null gesetzt.

Sind die Puffer negativ, so werden sie zur **Verspätung** hinzu addiert, ansonsten geschieht nichts. War das Verfahren erfolgreich, so sind alle Puffer positiv und die **Verspätung** bleibt beim Wert 0. Ansonsten nimmt die Variable **Verspätung** die Summe aller Limitüberschreitungen an.

7.5 Bestimmung der Einfügereihenfolgen

Hinsichtlich der Einfügereihenfolgen werden an dieser Stelle nur die Programmcodes für die Bestimmung der Reihenfolgen 1, 4, 6, 7 und 8 erklärt. Reihenfolgen 2, 3 und 5 entsprechen bei der Berechnung exakt ihren Gegenstücken, nur wird die Reihenfolge exakt invertiert.

Reihenfolge 1: Aufsteigend nach Höhe des Limits sortiert:

```
For i = 1 To Standortanzahl  
    Einfügereihenfolge(i) = i  
Next
```

Da die Limits der Punkte so generiert worden sind, dass der Punkt mit der niedrigsten Nummer das geringste Limit erhalten hat, sind die Punkte nach Höhe ihrer Limits sortiert. Die Position in der Einfügereihenfolge entspricht somit der numerischen Bezeichnung der Punkte.

Reihenfolge 4: Absteigend nach „Entfernung vom Startpunkt“ sortiert:

```
For i = 1 To Standortanzahl
    Sortierarray(i) = Kostenmatrix(0, i)
Next

For i = 1 To Standortanzahl
    Arbeitsindex = Array.IndexOf(Sortierarray, Sortierarray.Max)
    Einfügereihenfolge(i) = Arbeitsindex
    Sortierarray(Arbeitsindex) = 0
Next
```

Die für die Berechnung dieser und der nachfolgenden Reihenfolge gewählte Systematik ist darauf ausgelegt, Fehler bei exakt gleichen Werten zu vermeiden. So könnte es bei zwei exakt gleichen Entfernungen zweier Punkte zum Startpunkt dazu kommen, dass einer dieser Punkte doppelt in der Einfügereihenfolge erscheint und der andere nicht. Um dies zu vermeiden wird der Hilfsarray **Sortierarray** verwendet, in den innerhalb der ersten Schleife alle Entfernungen zum Startpunkt abgelegt werden.

Innerhalb der zweiten Schleife wird für die erste Einfügeposition der maximale Wert aus **Sortierarray** gewählt und dessen Index (welcher der Nummer des Punktes entspricht) dieser Position zugewiesen.⁷⁹ Damit der zugewiesene Wert nun nicht fälschlicherweise erneut abgerufen werden kann, wird er in Sortierarray mit dem Wert 0 überschrieben. Da der höchste Wert überschrieben wurde, wird im nachfolgenden Schritt der neue höchste Wert abgerufen.

Reihenfolge 6: Absteigend nach „mittlerer Entfernung zu allen anderen Punkten“ sortiert:

```
For i = 1 To Standortanzahl
    For j = 0 To Standortanzahl
        Mittelwertarray(j) = Kostenmatrix(i, j)
    Next
    Sortierarray(i) = (Mittelwertarray.Sum - 9999) / Standortanzahl
Next

For i = 1 To Standortanzahl
    Arbeitsindex = Array.IndexOf(Sortierarray, Sortierarray.Max)
    Einfügereihenfolge(i) = Arbeitsindex
    Sortierarray(Arbeitsindex) = 0
Next
```

Die Systematik von Reihenfolge 6 ist hinsichtlich der Sortierung (unterer Teil) identisch zur Reihenfolge 4, jedoch wird der **Sortierarray** hier innerhalb der ersten beiden ineinander verschachtelten Schleifen mit den mittleren Entfernungen gefüllt. Für die Berechnung dieser wird innerhalb der inneren Schleife je eine Spalte der Kostenmatrix in die Hilfsvariable **Mittelwertarray** übertragen. Von dieser wird dann der Mittelwert bestimmt (auch hier wird

⁷⁹ Hat zum Beispiel Punkt 3 die größte Entfernung zum Start, so wird diese Entfernung unter Sortierarray(3) gespeichert. Als Index dieses höchsten Wertes würde für die erste Einfügeposition somit 3 hervorgehen. Demnach wird unter Einfügereihenfolge(1) der Wert 3 hinterlegt.

der mit 9999 gesperrte Wert heraus gerechnet) und der entsprechenden Position vom **Sortierarray** zugeordnet.

Reihenfolge 7: Größte Entfernung zur bestehenden Tour:

```
For Schritt = 0 To Standortanzahl - 1
  For j = 1 To Standortanzahl
    Ausschluss = False
    For i = 0 To Schritt
      If j = Einfügereihenfolge(i) Then
        Ausschluss = True
      End If
      hilfsarray1(i) = Kostenmatrix(j, Einfügereihenfolge(i))
    Next
    If Ausschluss = False Then
      hilfsarray2(j) = hilfsarray1.Min
    Else
      hilfsarray2(j) = -9999
    End If
  Next
  Arbeitsindex = Array.IndexOf(hilfsarray2, hilfsarray2.Max)
  Einfügereihenfolge(schritt + 1) = Arbeitsindex
Next
```

Wenn auch vom Code her nicht deutlich umfangreicher, ist die Vorgehensweise bei Einfügereihenfolge 7 die mit Abstand komplexeste und rechenaufwendigste. Das liegt vor allem daran, dass hier sogar drei Schleifen ineinander geschachtelt sind. Die äußerste Schleife verwendet die Zählvariable **Schritt** und ermittelt in jedem Durchlauf eine weitere Einfügeposition. Die innerste Schleife wird immer genau so oft durchlaufen, wie bereits Punkte der Einfügereihenfolge zugeteilt worden sind. Hier werden die Entfernungen eines Punktes zu allen bereits zugeordneten Punkten in **hilfsarray1** abgespeichert. Zusätzlich wird mithilfe einer If-Abfrage geprüft, ob der Punkt bereits Teil der Einfügereihenfolge ist. Ist dies der Fall, so wird der Wahrheitswert der Variable **Ausschluss** auf **True** gesetzt. Innerhalb der mittleren Schleife werden immer alle Punkte durchgegangen. Nach Durchlauf der inneren Schleife wird anhand von **Ausschluss** geprüft, ob der aktuell geprüfte Punkt bereits Teil der Einfügereihenfolge ist. Ist dies der Fall, so wird er gesperrt. Nur wenn dies nicht der Fall ist, so wird der minimale Abstand zur Tour (kleinster Wert von **hilfsarray1**) in **hilfsarray2** abgespeichert. Nachdem die mittlere Schleife komplett durchgelaufen ist, wurde somit von jedem noch nicht der Reihenfolge zugeteiltem Punkt dessen minimale Entfernung zur Tour in **hilfsarray2** abgelegt. In der äußersten Schleife wird dann aus diesen Werten das Maximum ausgewählt und der entsprechende Punkt der Einfügereihenfolge hinzugefügt.

Reihenfolge 8: Zufällige Reihenfolge:

```
Zähler = 0
For i = 0 To Standortanzahl - 1
    Randomize()
    Zähler = Zähler + 0.001
    Folgesortiert(i) = Math.Round(Rnd() * 1000) + Zähler
    Zufallsfolge(i) = Folgesortiert(i)
Next

Array.Sort(Folgesortiert)

For i = 1 To Standortanzahl
    Arbeitsindex = Array.IndexOf(Zufallsfolge, Folgesortiert(i))
    Einfügereihenfolge(i) = Arbeitsindex + 1
Next
```

Für die Generierung einer zufälligen Reihenfolge wird innerhalb der ersten Schleife eine Reihe an ganzzahligen Zufallszahlen zwischen 0 und 1000 ermittelt. Um sicher zu gehen, dass diese nicht gleich sein können, wird ihnen in Form der Variable **Zähler** eine individuelle Dezimalzahl addiert. **Zähler** startet bei 0 und bei jedem Durchlauf erhöht sie sich um 0,001. Somit können die ersten 1000 generierten Zufallszahlen nicht exakt gleich sein. Da für jeden Ort eine Zufallszahl generiert wird und das Programm bei ca. 500 Orten an seine Grenzen stößt, ist dies ausreichend um eine Dopplung von Zufallszahlen zu vermeiden. Die Zufallszahlen werden in den zwei Arrays **Zufallsfolge** und **Folgesortiert** gespeichert. Anschließend wird nur **Folgesortiert** sortiert. Dadurch kann innerhalb der zweiten Schleife der Index eines Wertes von **Folgesortiert** innerhalb **Zufallsfolge** abgerufen werden. Dieser **(Arbeits-) Index** mit 1 addiert⁸⁰ gibt dann den einzufügenden Punkt wieder, während der Index innerhalb von **Folgesortiert** die Position in der Einfügereihenfolge widerspiegelt.

8. Schlussbetrachtung

Als Ergebnis dieser Arbeit konnte besonders die nach aufsteigenden Limits sortierte Einfügereihenfolge in Verbindung mit dem gewichteten Kriterium als die in den meisten Fällen beste Variante hervorgehoben werden. Zudem konnte in Form der Gewichtung von Kosten und Puffern eine Stellschraube definiert werden, mit welcher sich das Lösungsverfahren zumindest theoretisch an die Rahmenbedingungen anpassen lässt. Eine konkrete Empfehlung der Gewichtung in Abhängigkeit von der Höhe der Limits konnte anhand der zugrunde liegenden Daten jedoch bisher leider nicht gegeben werden. Daher ist darin ein interessanter Ansatzpunkt für eine weiterführende Analyse zu sehen.

Ein weiterer Ansatz, welcher in der Zukunft verfolgt werden könnte, ist in Ergebnissen zu sehen, welche geringe Verspätungen zugunsten günstigerer Gesamtkosten hinnehmen. Zwar wurde im Rahmen dieser Arbeit versucht, diese einzubeziehen, jedoch ergaben sich kaum aussagekräftige Ergebnisse, da die Verfahren nicht auf eine solche Zielsetzung ausgelegt

⁸⁰ Damit die Sortierung durchgeführt werden kann, muss die nullte Position im Array besetzt sein. Da Punkt 0 als Startpunkt nicht der Einfügereihenfolge hinzugefügt werden kann, wird dies durch die Addition von Eins hier wieder ausgeglichen.

sind. Ein möglicher Ansatzpunkt um dies weiter zu verfolgen, wäre ein abgewandeltes gewichtetes Kriterium, bei welchem unzulässige Positionen nicht gesperrt werden. (Schritt 4.3a entfällt) Dies sollte die Wahl von kostengünstigen Verbindungen unterstützen, während der Einfluss der Puffer es bei angemessener Gewichtung verhindern sollte, dass die Verspätungen eskalieren.

Um abseits der angepassten Gewichtung eine Verbesserung des Verfahrens zu erzielen, ergäbe sich als letzter Ansatz die Entwicklung von Systematiken, welche die Ineffizienzen und Inkompatibilitäten des Kosten- und Pufferkriteriums ausschalten oder eingrenzen. Gelingt es, die Probleme der Extremfälle auszuschalten, so kann auch das gewichtete Kriterium profitieren. So wäre die Anwendbarkeit anderer Einfügereihenfolgen (außer der limitorientierten) auch bei extremeren Gewichtungen wünschenswert. Als noch wichtiger erscheint es jedoch, die Ineffizienz der pufferbasierten Entscheidung zu reduzieren.

Auch abseits dieser Probleme, welche durch die Analyse festgestellt werden konnten, sind einige kritische Bemerkungen zur Vorgehensweise dieser Arbeit vorzunehmen. So liefen die Untersuchungen einiger Faktoren ins Leere, weil die richtigen Rahmenbedingungen für die Untersuchung nicht hergestellt werden konnten. Dies betrifft vor allem die Verteilung der Standorte⁸¹ sowie den Ansatz der „erweiterten Erfolgsquote“. Auf der anderen Seite wirkte sich die extrem große Menge an Daten sehr überwältigend aus und drohte den Rahmen der Arbeit zu sprengen. Während also der Wegfall einiger Analysemöglichkeiten bedauerlich ist, half dies auch, eine fokussiertere Herangehensweise an die Analyse zu wählen.

Letztlich ist anzumerken, dass durch die große Datenmenge eine frühzeitige Erhebung notwendig wurde. Ansätze zur Verbesserung des Verfahrens oder zur effizienteren Gestaltung von Formeln konnten zum Ende hin nicht mehr implementiert werden, da die Erhebung der Daten und Teile der Analyse bereits auf Basis der „alten“ Ansätze und Formeln durchgeführt worden waren. Insgesamt ist der Autor jedoch zufrieden, dass ein effektiver und vergleichsweise einfacher Ansatz zur Lösung des Problems auch ohne spätere Anpassungen gefunden werden konnte.

⁸¹ Die Messreihen der „Städtecluster“- und geballten Generierung hatten aufgrund von „zu guten“ Erfolgsquoten kaum eine Aussage, da keine Differenzierung möglich war.

V. Quellenverzeichnis

Arnold, Dieter; Isermann, Heinz; Kuhn, Axel; Tempelmeier, Horst; Furmans, Kai (2008): *Handbuch Logistik*, 3., neu bearbeitete Auflage, Berlin Heidelberg

Domschke, Wolfgang; Drexel, Andreas; Klein, Robert; Scholl, Armin (2015): *Einführung in Operations Research*, 9. Auflage, Berlin Heidelberg

Hinrichsen, Jens (1975): *Verkehrswissenschaftliche Studien 28; Branch- and Bound-Verfahren zur Lösung des Rundreiseproblems*, Göttingen

Neumann, Klaus; Morlock, Martin (2002): *Operations Research*, 2. Auflage, München Wien

Thonemann, Ulrich (2005): *Operations Management*, München

Vahrenkamp, Richard (2003): *Quantitative Logistik für das Supply Chain Management*, München

Wenger, Wolf (2009): *Multikriterielle Tourenplanung*, Hohenheim

Zimmermann, Werner; Stache, Ulrich (2001): *Operations Research; Quantitative Methoden zur Entscheidungsvorbereitung*, 10. Auflage, München Wien

VI. Eidesstattliche Erklärung

Ich versichere, dass ich die vorliegende Arbeit ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Quellen und Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quelle kenntlich gemacht.

Ort, Datum

Unterschrift / Stefan Fortmann

VII. Einverständniserklärung

Ich erkläre mich damit einverstanden, dass ein Exemplar meiner Master-Thesis in die Bibliothek des Fachbereichs aufgenommen wird;
Rechte Dritter werden dadurch nicht verletzt.

Ort, Datum

Unterschrift / Stefan Fortmann