

# DIPLOMARBEIT

Titel der Magisterarbeit

„Kombinierte Standort-Touren-Planung in der  
Produktneueinführung“

Verfasser

Christian Strobach

angestrebter akademischer Grad

Magister der Sozial- und Wirtschaftswissenschaften  
(Mag. rer. soc. oec.)

Wien, im Juni 2008

Studienkennzahl lt. Studienblatt:  
Studienrichtung lt. Studienblatt:  
Betreuer

A 157  
Internationale Betriebswirtschaft  
Univ.-Prof. Dr. Walter Gutjahr

## Inhaltsverzeichnis

1	EINFÜHRUNG .....	1
1.1	Aufgabenstellung.....	1
1.2	Motivation .....	1
1.3	Ziele.....	2
1.4	Gliederung der Arbeit .....	2
2	OPERATIONS RESEARCH (OR).....	4
2.2	Theoretische Grundlagen .....	4
2.1.1	Modelltypen.....	5
2.1.2	Teilgebiete des Operations Research .....	5
2.2	Lineare Optimierung .....	7
2.3	Lineare ganzzahlige und kombinatorische Optimierung.....	9
2.3.1	Lineare ganzzahlige Optimierung .....	9
2.3.2	Kombinatorische Optimierung .....	10
2.4	Spezialfälle der kombinatorischen Optimierung.....	11
2.4.1	Standortplanung (Facility Location Problem) .....	11
2.4.2	Tourenplanung (TPP).....	12
2.4.3	Location-Routing-Problem (Standort-Tourenproblem) .....	14
2.5	Lösungsverfahren der kombinatorischen Optimierung .....	16
2.5.1	Allgemein.....	16
2.5.2	Exakte Lösungsverfahren.....	18
2.5.3	Heuristische Lösungsverfahren .....	18
3	PROBLEMMODELLIERUNG.....	22
3.1	Problemformulierung .....	23
3.2	Datenbeschaffung und Aufbereitung .....	24
3.3	Verbales Modell.....	27
3.3.1	Zielfunktion.....	28
3.3.2	Nebenbedingungen .....	29
4	LÖSUNGSVERFAHREN.....	30
4.1	Iterated Local Search .....	30
4.2	Softwareeinsatz.....	31
4.3	Programmaufbau und Implementierung.....	32
4.3.1	Programmierschritt I .....	34
4.3.2	Programmierschritt II .....	34
4.3.3	Programmierschritt III.....	35
4.3.4	Programmierschritt IV .....	37

---

4.4	Zusammenfassung .....	38
5	EXPERIMENTELLER TEIL .....	39
5.1	Interpretation der Ausgabedaten .....	39
5.2	Auswertung und Analyse der Daten - Standortplanung.....	40
5.3	Auswertung und Analyse der Daten - Tourenplanung .....	41
5.4	Graphischer Tourenplan .....	42
5.5.1	Computerlaufzeit.....	43
5.5.2	Lösungsqualität .....	43
5.5.3	Sensibilitätsanalyse 1.....	47
5.3.3.1	Probleminstanz (0,05GE).....	47
5.3.3.2	Probleminstanz (0,02GE).....	50
5.3.3.3	Probleminstanz (0,005GE) .....	52
5.3.3.4	Probleminstanz (0,001GE) .....	54
5.5.4	Zusammenfassung .....	56
6.	SCHLUSSFOLGERUNGEN .....	57
6.1	Die wichtigsten Ergebnisse dieser Arbeit.....	57
6.1.1	Informationsqualität .....	57
6.1.2	Modell .....	57
6.1.3	Lösungsverfahren .....	58
6.2	Veränderung anderer Parameterwerte .....	58
7.	LITERATURVERZEICHNIS .....	60
	Kurzfassung.....	62
	Abstract .....	63
	Danksagung.....	64
	Lebenslauf.....	65

# 1 EINFÜHRUNG

Die Themenstellung wurde vorrangig aufgrund meines im dritten Studienabschnitt gelegten Schwerpunktes Operations Research am *Institut für Statistics and Decision Support Systems* (Betreuer: Prof. Dr. Walter Gutjahr) getroffen. Parallel konnte ich während meines Studiums auch diesbezüglich einen entscheidenden Bezug zur Praxis herstellen. Ein in Wien ansässiges Unternehmen im Bereich Werbung und Promotion stellte mir schließlich erfahrungsspezifische Daten zur Verfügung. Mein Bemühen beruht auf der Absicht, den in diesem Feld tätigen Unternehmen ein Standardtool zu bieten, das den Optimierungs- und Entscheidungsprozess für ähnliche Projekte effizienter und wirtschaftlicher gestaltet.

## 1.1 Aufgabenstellung

Das Management eines bekannten österreichischen Molkereiunternehmens hat sich entschieden, ein neues Produkt auf dem Markt zu bringen. Um die Zielgruppen auf die Einführung des Produkts aufmerksam zu machen, hat die Marketingabteilung als erste verkaufsfördernde Werbemaßnahme die „Sampling Strategie“.<sup>1</sup>

ausgewählt. In Absprache mit dem Management wurde eine Agentur beauftragt, diese Werbemaßnahme in Wien durchzuführen. Die Zielvorgaben des Unternehmens lauten dabei zum einen die Steigerung des Bekanntheitsgrades durch Maximierung der erzielbaren Sichtkontakte<sup>2</sup> und zum anderen die Minimierung der Projektkosten. Es sollen also jene Standorte in Wien ausgewählt werden, die den größten Nutzen in Abhängigkeit von Transport-, Personal- und Projektkosten erzielen. Meine Aufgabe besteht darin, ein mathematisches Entscheidungsmodell zu konstruieren und das entsprechend effiziente Lösungsverfahren auszuwählen, welches die vorgegebenen Projektziele unter Berücksichtigung kapazitätsbedingter Ressourcen optimal befriedigt.

## 1.2 Motivation

Warum ein mathematisches Modell?

In sehr vielen unterschiedlichen Bereichen so auch im Bereich der Wirtschaftswissenschaften ist die Konstruktion mathematischer Modelle von sehr hoher Relevanz. Diese Modelle zeigen eine zusammenfassende Darstellung des Kernproblems eines Entscheidungsprozesses mit allen wechselseitigen Abhängigkeiten und erleichtern die Analyse bzw. Auswertung erheblich.

Durch die Abbildung der Realität auf ein Modell bietet diese Anwendung die Möglichkeit, den Prozeß der Entscheidungsevaluierung, Entscheidungsfindung und den Gebrauch von geeigneten Lösungsverfahren transparent, zeitoptimal und kostengünstig zu gestalten.

---

<sup>1</sup> Das Wort „Sampling“ stammt aus dem englischen und bedeutet „Stichprobe“ oder „Auswahl“. Dabei handelt es sich um eine Maßnahme der Verkaufsförderung, bei der Warenproben durch Kooperationspartner ausgegeben bzw. verschenkt werden.  
Quelle: Internet/Wikipedia

<sup>2</sup> Sichtkontakte entsprechen jener Anzahl potentieller Konsumenten, die durch eine Werbeaktion auf das Produkt aufmerksam werden.

Warum „kombinatorische Optimierung“?

Wie in einem der folgenden Kapitel detailliert dargelegt wird, ist die Deklaration des Optimierungsproblems allgemein als ein „kombinatorisches Optimierungsproblem“<sup>3</sup> oder auch „lineares ganzzahliges Optimierungsproblem“<sup>4</sup> unabdingbar und führt zu einem zufrieden stellenden Ergebnis. Aufgrund der Nichtteilbarkeit einiger Entscheidungsvariablen, wird in dieser Fallstudie davon ausgegangen, daß es sich um ein ganzzahliges Optimierungsproblem handelt. Mit anderen Worten dürfen einige Variablen nur ganzzahlige und nicht beliebig reelle Werte annehmen.

Warum Heuristiken?

Grundsätzlich kann man bei der linearen Optimierung zwischen exakten und heuristischen Lösungsverfahren<sup>5</sup> unterscheiden. Zusätzlich zur Modellierung werden speziell entwickelte Algorithmen zur Lösung von Optimierungsproblemen angewendet. Dafür wurden in den letzten Jahren eigens spezielle Softwarepakete entwickelt. Heuristiken verkürzen die Rechenzeiten und liefern zulässige, optimale Lösungen.

### 1.3 Ziele

Das Hauptziel dieser Arbeit ist durch die Aufgabenstellung vorgegeben. Es soll also ein mathematisches Modell für den Entscheidungsprozess spezifiziert werden. Hier steht nicht nur eine gute Performance des Modells im Vordergrund, sondern es soll auch geklärt werden, ob es mit dem gewählten Ansatz und dem eingesetzten Lösungsverfahren möglich ist, optimale Ergebnisse zu erzielen.

Neben diesem Hauptziel können auch diverse Unterziele identifiziert werden:

- Kontrolle der Güte/Qualität der ermittelten Ergebnisse
- Überprüfung der Validität des Modells
- Interpretation und Analyse der ausgewerteten Daten
- Feedback und Empfehlungen an das Management

### 1.4 Gliederung der Arbeit

**Kapitel 2** beschäftigt sich mit den allgemeinen Grundlagen des Operations Research. Nach einer kurzen Einleitung folgt eine relevante Vertiefung im Bereich der linearen und ganzzahlig linearen (kombinatorischen) Optimierung. In Anbetracht der konkreten Aufgabenstellung, werden anschließend drei Spezialfälle der kombinatorischen Optimierung herausgriffen und separat behandelt. Der letzte Teil

---

<sup>3</sup> Vgl. Hiller, Lieberman [1988], Operations research, 4. Auflage, Verlag Oldenbourg;  
WISU-Zeitschrift [Juni 2003], Lange Verlag Düsseldorf  
Stephan Dempe, Heiner Schreier [2006], Operations research – Deterministische Modelle und Methoden, International Student Edition, 1. Auflage  
Wolfgang Grundmann [2002], Operations research – Formeln und Methoden, , 1. Auflage, Teubner Verlag

<sup>4</sup> Vgl.: Hiller, Lieberman [1988], Operations research, 4. Auflage, Verlag Oldenbourg

<sup>5</sup> Vgl.: Wolfgang Grundmann [2002], Operations research – Formeln und Methoden, , 1. Auflage, Teubner Verlag  
Werners Brigitte [2008], Grundlagen des Operations Research, Springer Verlag, 2. Auflage  
Klaus-Peter Kistner [1993], Optimierungsmethoden, Physica Verlag Heidelberg, 2. Auflage  
Wayne Winston [2005], Operations research – Applications and Algorithms, International Student Edition, 4th Edition  
Jukka Siedentopf [1996], Eine exemplarischer Vergleich lokaler Suchverfahren (Teil I), Universität Leipzig  
Domschke, Scholl [2006], Heuristische Verfahren - Arbeits- und Diskussionspapiere, Universität Jena  
Fink, Rothlauf [2006], Heuristische Optimierungsverfahren in der Wirtschaftsinformatik, Universität Mannheim

des Kapitels befaßt sich mit den unterschiedlichen Lösungsverfahren und deren Anwendungsgebiete.

**Kapitel 3** widmet sich ganz der Problemmodellierung. Zu Beginn wird die Aufgabenstellung konkretisiert und auf die vorgegebenen Ziele und Restriktionen detailliert eingegangen. In weiterer Folge wird das Entscheidungsmodell verbalisiert und ein mathematisches Modell dafür entworfen.

**Im Kapitel 4** wird das gewählte Lösungsverfahren bestimmt und auf die konkrete Problemstellung adaptiert. Danach folgen der Softwareeinsatz und eine Beschreibung des Programmablaufes. Zentraler Kern dieses Kapitels ist die Implementierung des Modells in das Softwarepaket SCILAB.

**Kapitel 5** beschäftigt sich mit dem experimentellen Teil der Diplomarbeit. Aufbauend auf die vorangegangenen Kapitel, wird das Entscheidungsmodell mittels eines Lösungsverfahrens getestet und bewertet. Abschließend werden die Ergebnisse eingehend analysiert und interpretiert.

Das abschließende **Kapitel 6** fasst die wichtigsten Erkenntnisse noch einmal zusammen und weist auf weitere untersuchenswerte Aspekte bezogen auf die aktuelle Problemstellung hin.

## 2 OPERATIONS RESEARCH (OR)

Operations Research bedeutet soviel wie Unternehmensforschung und befaßt sich mit der Analyse von praxisnahen, komplexen Entscheidungsproblemen. Es handelt sich dabei um quantifizierbare Probleme, bei denen es meistens mehrere Entscheidungsmöglichkeiten gibt. Ziel der OR ist es, sämtliche zulässige Alternativen zu evaluieren und die beste (optimale) auszusuchen.

Das Anwendungsspektrum dieser Methodik der Entscheidungsfindung ist überaus breit gehalten und erfreut sich überdies in den letzten fünf Jahrzehnten einer kontinuierlich steigenden Zuwachsrates. Operations Research wird unter anderem sehr intensiv im Bereich der Wirtschaft, dem Militärwesen, im Krankenhausmanagement oder im Verwaltungssektor angewendet.

Die Wurzeln des OR lassen sich geschichtlich gesehen bis in die frühen 40er Jahre des vorigen Jahrhunderts zurückverfolgen. Aufgrund der knappen Ressourcen während des zweiten Weltkrieges waren Länder wie die USA, Großbritannien oder die Sowjetunion zur effektiven Zuteilung vorhandener Kapazitäten gezwungen. Durch die stetige Weiterentwicklung auf diesem Gebiet und durch die industrielle und wirtschaftliche Nutzbarkeit, ist die OR über die Jahre zu einem eigenen Wissenszweig gewachsen.

### 2.2 Theoretische Grundlagen

*„OR im weitem Sinne beschäftigt sich mit der Modellbildung und Lösungsfindung (Verwendung von Algorithmen) sowie Methoden zur Datenermittlung.“<sup>6</sup>*

*„OR im engeren Sinne widmet sich primär der Entwicklung von Algorithmen.“<sup>6</sup>*

Zentraler Kern der OR ist die Abbildung eines realen Entscheidungsproblems durch ein Optimierungs- oder Simulationsmodell unter Anwendung exakter oder heuristischer Lösungsverfahren. Nachstehen eine schematische Darstellung des Prozesses:

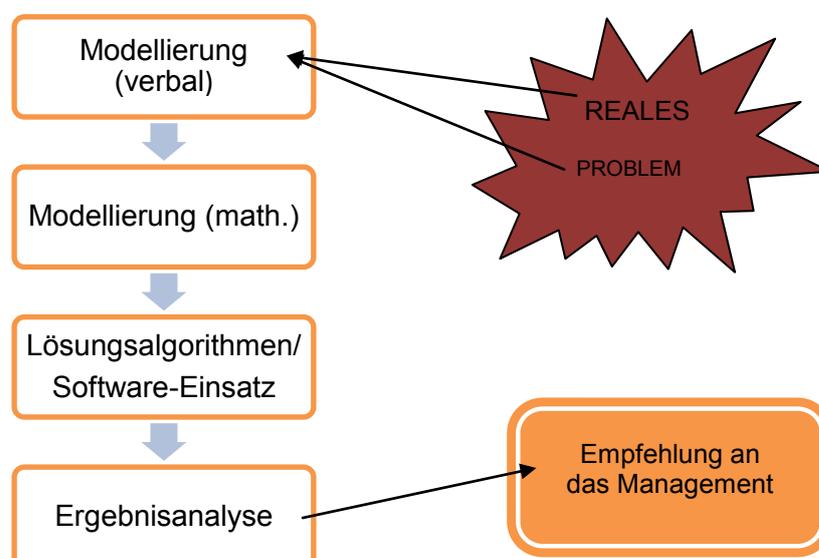


Abbildung 2.1: Schematische Darstellung des Problemlösungsprozesses

<sup>6</sup> Zitat von Domschke, Drexl [2002], Einführung in Operations Research, 5.Auflage

### 2.1.1 Modelltypen

„Ein Entscheidungsmodell dient uns zur Vorbereitung und zum Treffen von Entscheidungen.“<sup>7</sup> In der OR stehen uns verschiedene Modelltypen für den Entscheidungsprozess zu Verfügung, deren Charakterisierung ich folglich nur überblicksartig umfassen werde.

*Entscheidungs- bzw. Optimierungsmodelle* stellen ein konkretes Entscheidungs- oder Planungsproblem dar. Ziel solcher Modelle ist die Generierung optimaler oder auch suboptimaler Lösungen. Die einfachste Form eines solchen Modells beinhaltet eine zulässige Menge von Alternativen und exakt eine Zielfunktion.

*Simulationsmodelle* sind in der Regel sehr komplexe Optimierungsmodelle, für die bislang jedoch keine analytischen Lösungsverfahren existieren.

*Beschreibungs- Erklärungs- und Prognosemodelle* sind Modelle, die hauptsächlich zur Informationsgewinnung dienen. Diese Modellart beschreibt die Elemente und deren Beziehungen in realen Systemen. Erklärungs- und Prognosemodelle hingegen dienen der Vorhersage von zukünftig eintretenden Entwicklungen.

Der effizienten Modellierung wird in der OR stets eine sehr große Bedeutung zuteil. Ziel ist die Entwicklung eines Modells, das Mithilfe einer Standardsoftware im Vergleich zu anderen Modellen mit gleicher optimaler Lösung, einen geringeren Rechenaufwand verursacht.

Häufig können zum Beispiel Planungs- und Entscheidungsprobleme nur mittels kombinatorischer oder ganzzahliger Optimierung gelöst werden. Nachteil einer solchen Optimierung ist jedoch der hohe Rechenaufwand. Diesem kann aber mit geeigneten heuristischen Lösungsverfahren entgegengewirkt werden. Wie sich im Verlauf dieser Arbeit noch zeigen wird, ist auch für die konkret vorliegende Aufgabenstellung der Einsatz einer Heuristik als Lösungsvariante notwendig. Kapitel 3 befaßt sich mit diesem Modell im Detail und versucht weiters auf die allgemeinen Modellierungsgrundsätze einzugehen.

### 2.1.2 Teilgebiete des Operations Research

Die Operations Research lässt sich zunächst in folgende Teilgebiete unterteilen:

- *Lineare Optimierung*  
Die lineare Optimierung ist der wichtigste Teilbereich der Operations Research. Im Mittelpunkt stehen Modelle, die durch lineare Zielfunktion(en), lineare Nebenbedingungen und Nichtnegativitätsbedingung charakterisiert sind. Als Lösungsverfahren werden z.B. der Simplex-Algorithmus oder die Dualitätstheorie angewendet.
- *Graphentheorie und Netzplantechnik*  
„Es eine Technik, die zur Veranschaulichung von Organisationsstrukturen und Projektabläufen dient. Dies geschieht in graphischer Form. Modelle und Verfahren zur Bestimmung kürzester Wege oder maximaler und kostenminimaler Flüsse zählen auch zu dieser Rubrik“<sup>8</sup>.

<sup>7</sup> Zitat: WISU-Zeitschrift [Juni 2003], Lange Verlag Düsseldorf

<sup>8</sup> Zitat: Domschke, Drexl [2002], Einführung in Operations Research, 5. Auflage

Die Netzplantechnik ist in der Praxis eine oft eingesetzte Methode der Planung und dient zur Kontrolle und dem Monitoring betrieblicher Abläufe.

- *Ganzzahlige lineare oder kombinatorische Optimierung*  
Bei dieser Art der Optimierung dürfen die Variablen (eine oder mehrere) entweder nur ganzzahlige oder binäre Ausprägungen annehmen. Die Lösungen bei einer kombinatorischen Optimierung entstehen vereinfacht gesagt durch das Kombinieren oder Reihen von Lösungselementen. Bekannte Probleme in diesem Kontext sind Reihenfolge-, Gruppierungs-, Zuordnungs- und Auswahlprobleme. Der Großteil bei kombinatorischen Problemen lässt sich als ganzzahlige bzw. binäre Optimierungsmodelle formulieren und lösen.
- *Dynamische Optimierung*  
„Bei der dynamischen Optimierung (DO) werden Modelle betrachtet, die in einzelne „Stufen“ (z.B. Zeitabschnitte) zerlegt werden. Dadurch wird gewährleistet, dass die Gesamtoptimierung durch eine stufenweise, rekursive Optimierung ersetzbar ist.“<sup>9</sup>
- *Nichtlineare Optimierung*  
Obwohl allgemein sehr viele Entscheidungsprobleme mit Hilfe von linearen Optimierungsmodellen gelöst werden können, findet man in der Realität oft nichtlineare Entscheidungsstrukturen vor. Beispiele wären etwa nichtlineare Preis-Absatz- oder Kapitalwertfunktionen. Nachteil solcher Modelle ist der wesentlich höhere Rechenaufwand.<sup>10</sup>
- *Warteschlangentheorie*  
Sie untersucht das Abfertungsverhalten von Service- und Bedienstationen. Ein Beispiel bei dem die Warteschlangentheorie zum Einsatz kommt, sei der Bankschalter genannt. Relevante Kenngrößen hierbei sind mittlere Wartezeiten von Aufträgen oder Kunden, Bediennraten sowie Kapazitätsauslassungen.<sup>11</sup>
- *Simulation*  
Simulationen dienen zum „Durchspielen“ einzelner Alternativen bzw. Systemvarianten in komplexen Systemen. Sie werden eingesetzt, wenn eine exakte Analyse zu aufwendig, gefährlich oder kostenintensiv wäre.<sup>12</sup>

In Anbetracht meiner konkreten Aufgabenstellung werde ich zunächst auf die Grundlagen verschiedener Optimierungsprobleme näher eingehen. Diese Grundlagen dienen zum allgemeinen Verständnis und bilden die Basis zur Modellierung des Entscheidungsproblems. Zu diesen zählen die lineare und ganzzahlig lineare Programmierung. Bei der ganzzahligen Programmierung werde ich das Thema weiter vertiefen und zur kombinatorischen Optimierung, die ebenfalls in diesen Bereich fällt, überleiten. Danach folgt eine kurze Beschreibung dreier Spezialfälle, die eine ähnliche Problemstruktur wie die vorliegende Fallstudie aufweisen. Nachdem ich die Grundregeln der Modellierung behandelt habe, werde ich mich im darauffolgenden Kapitel mit der zweiten bedeuteten Thematik der Operations Research auseinandersetzen – respektive dem Lösungsprozess. Hier werde ich zunächst die unterschiedlichen Verfahren beschreiben, die bei der

<sup>9</sup> Zitat: Domschke, Drexl [2002], Einführung in Operations Research, 5. Auflage

<sup>10</sup> Vgl.: WISU-Zeitschrift [Juni 2003], Lange Verlag Düsseldorf

<sup>11</sup> Vgl.: WISU-Zeitschrift [Juni 2003], Lange Verlag Düsseldorf

<sup>12</sup> Vgl.: Domschke, Drexl [2002], Einführung in Operations Research, 5. Auflage

Lösungsfindung zur Anwendung kommen. Mit dem Einsatz von Algorithmen oder auch Heuristiken wird der Entscheidungsprozess abschließend dargestellt.

## 2.2 Lineare Optimierung

Die lineare Optimierung ist eines der gebräuchlichsten Werkzeuge bzw. eine der bedeutendsten Techniken zur Lösung des Problems der Ressourcenallokation. „Es wird also überall dort angewendet, wo das Aktivitätsniveau bestimmter Tätigkeiten festzulegen ist, die um die für ihre Durchführung notwendigen, begrenzten Ressourcen konkurrieren.“<sup>13</sup>

Zunächst muß dieses Problem in Form eines mathematischen Problems (LP-Modell) definiert und formuliert werden. Wie weiter oben im Text bereits erwähnt, liegt ein lineares Problem nur dann vor, wenn die Zielfunktion(en) und alle Nebenbedingungen linear sind und weiters die Nichtnegativitätsbedingung eingehalten wird. Eine allgemeine Formulierung eines LP-Modells bekommt man unter Verwendung der Vektorschreibweise:

Jedes beliebige LP-Modell läßt sich in der folgenden Form darstellen:

$$\text{(Minimiere) Maximiere } F(x_1 \dots x_p) = \sum_{j=1}^p c_j x_j$$

unter den Nebenbedingungen

$$\sum_{j=1}^p a_{ij} x_j \text{ (}\geq\text{)} \leq b_i \quad \text{für } i = 1, \dots, m$$

und der Nichtnegativitätsbedingung (Nnb)

$$x_j \geq 0 \quad \text{für } j = 1, \dots, p$$

Wobei  $x=(x_1, \dots, x_p)$  den Lösungsvektor,  $c=(c_1, \dots, c_p)$  den Zielfunktionskoeffizienten,  $a_{ij}$  die Koeffizientenmatrix und  $b_i=(b_1, \dots, b_m)$  die rechte Seite darstellen.

Um ein lineares Optimierungsproblem am Computer ausführen zu können, bedarf es ausschließlich algebraischen Anweisungen. Da hierfür Gleichungen anstatt Ungleichungen notwendig sind, müssen zunächst die funktionalen Ungleichheitsbedingungen in äquivalente Gleichheitsbedingungen umwandeln werden. Diese Umformung geschieht durch Einführung von sogenannten Schlupfvariablen (Slackvariablen), die mit Null bewertet werden. Dies betrifft jedoch nicht die Nichtnegativitätsbedingung.

Wir erhalten so folgende Normalform eines LP's:

$$\text{Maximiere } F(x_1, \dots, x_p, x_{p+1}, \dots, x_n) = \sum_{j=1}^p c_j x_j + \sum_{j=p+1}^n 0 \cdot x_j$$

<sup>13</sup> Zitat von: Hiller, Lieberman [1988], Operations research, 4. Auflage, Verlag Oldenbourg (Seite 25)

Unter den Nebenbedingungen

$$\sum_{j=1}^p a_{ij} x_j + x_{p+1} = b_i \quad \text{für } i = 1, \dots, m_1$$

$$\sum_{j=1}^p a_{ij} x_j - x_{p+1} = b_i \quad \text{für } i = m_1 + 1, \dots, m_2$$

$$\sum_{j=1}^p a_{ij} x_j = b_i \quad \text{für } i = 1, \dots, m$$

Und der Nichtnegativitätsbedingung (Nnb)

$$x_j \geq 0 \quad \text{für } j = 1, \dots, n$$

Die ursprünglichen Variablen  $x_1, \dots, x_p$  des Problems werden als Entscheidungsvariablen und die neuen  $x_{p+1}, \dots, x_n$  als Schlupfvariablen bezeichnet.

Oftmals wird ein LP-Modell auch in Matrixschreibweise dargestellt:

$$\begin{array}{l} \text{Maximiere } F(x) = \\ \text{Nebenbedingung:} \end{array} \quad \begin{array}{l} c^T x \\ Ax = b \\ x \geq 0 \end{array}$$

$$c = \begin{pmatrix} c_1 \\ \dots \\ c_{n-m} \\ 0 \\ \dots \\ 0 \end{pmatrix} \quad \text{und} \quad A = \begin{bmatrix} a_{11} & \dots & a_{1,n-m} & 1 & & \\ \cdot & \dots & \cdot & & 1 & \\ a_{m1} & \dots & a_{m,n-m} & & & 1 \end{bmatrix}$$

$C$  und  $x$  sind jeweils  $n$ -dimensionale Vektoren;  $b$  ist ein  $m$ -dimensionaler Vektor und  $A$  eine  $(m \times n)$  – Matrix. Diese Darstellung wird **kanonische Form** genannt.

## 2.3 Lineare ganzzahlige und kombinatorische Optimierung

### 2.3.1 Lineare ganzzahlige Optimierung

Bei der linearen Optimierung geht man davon aus, dass alle Variablen reellwertig sind. Mit anderen Worten impliziert diese Eigenschaft also, daß alle Aktivitäten beliebig teilbar sind. Bei vielen praktischen (realitätsnahen) Problemen sind jedoch nur ganzzahlige Entscheidungsvariablen sinnvoll bzw. kann der Lösungsprozess nur mit ganzzahligen Werten wirklich realisiert werden wie etwa bei der Verwendung von Menschen, Maschinen oder Fahrzeuge.

Das mathematische Modell zur ganzzahligen Programmierung ist mit dem der linearen Optimierung fast identisch, allerdings durch die zusätzliche Restriktion der Ganzzahligkeit ergänzt:

$$(\text{Minimiere}) \text{ Maximiere } F(x_1 \dots x_p) = \sum_{j=1}^p c_j x_j$$

Mit der bereits bekannten Nebenbedingung

$$\sum_{j=1}^p a_{ij} x_j (\geq) \leq b_i \quad \text{für } i = 1, \dots, m$$

Und der zusätzlichen Restriktion zur Nnb

$$x_j \in \mathbb{Z}^n$$

$$x_j \geq 0 \quad \text{für } j = 1, \dots, p$$

Zusätzlich wird man in der Praxis häufig mit Situationen konfrontiert werden, bei denen sogenannte JA- oder NEIN-Entscheidungen getroffen werden müssen. Beispiele dafür wären: Soll ein bestimmtes Projekt durchgeführt werden oder soll ein bestimmter Standort angefahren werden?

In solchen Fällen führt die ganzzahlige Programmierung zu Problemen. Stehen solche eingeschränkten Entscheidungen zur Auswahl, dann können diese Wahlmöglichkeiten (JA oder NEIN) durch Entscheidungsvariablen dargestellt werden. Durch eine entsprechende Verwendung von **binären Variablen** (0 oder 1) kann dieses Problem beseitigt und in weiterer Folge gelöst werden:

$$x_j = \begin{cases} 1, & \text{wenn } j\text{te Entscheidung JA} \\ 0, & \text{wenn } j\text{te Entscheidung NEIN} \end{cases}$$

Grundsätzlich wären solche Probleme ohne Ganzzahligkeitsrestriktion leichter zu lösen, was sich auch darin widerspiegelt, dass die Lösungsverfahren zur ganzzahligen Programmierung viel aufwendiger sind als jene der linearen Programmierung. In diesem Kontext möchte ich zur kombinatorischen Optimierung überleiten.

### 2.3.2 Kombinatorische Optimierung

Während bei linearen Programmen der zulässige Lösungsraum, definiert durch jene Gesamtheit der zulässigen Lösungen, die alle Nebenbedingungen des Optimierungsmodells erfüllt, nicht abzählbar ist, ist dieser bei ganzzahligen Problemen abzählbar. Der feine Unterschied zur kombinatorischen Optimierung liegt darin, dass der zulässige Lösungsbereich bei kombinatorischen Problemen endlich ist. Viele betriebswirtschaftliche Entscheidungsprobleme sind kombinatorischer Natur. Dabei entstehen die Lösungen durch „Kombinieren“ und „Reihen“ von Lösungselementen.

Zunächst könnte die optimale Lösung zum Beispiel durch eine **vollständige Enumeration** ermittelt werden. Diese Lösungsalternative wird in der Praxis jedoch nicht angewendet, da die Anzahl der zu überprüfenden Lösungen mit der Problemgröße exponentiell steigt. Das Ergebnis wäre ein ineffizienter und unwirtschaftlicher Rechenaufwand. So müsste beispielsweise ein Computer bei bereits 20 Binärvariablen  $2^{20}=1.048.576$  Rechenoperationen durchführen.

Eine vollständige Enumeration ist auf folgenden Gründen unökonomisch:

- Aufbau aller Alternativen führt zu exponentiellen Rechenaufwand
- Keine zwischenzeitliche Prüfung der Zulässigkeit
- Keine zwischenzeitliche Berücksichtigung des Zielfunktionsbeitrages der einzelnen Entscheidungsvariablen

Für viele praktische Probleme sind daher neuere Algorithmen (insbesondere das Branch-and-Bound.-Verfahren) oder aber auch gewisse Näherungs-Algorithmen (Heuristische Verfahren) eine gute Alternative. Diese gelangen zwar nicht mit Sicherheit zur optimalen Lösung, jedoch mit Rechenzeiten im Sekunden- oder Minutenbereich gute suboptimale Lösungen ermitteln. Näheres zu den in meiner Diplomarbeit angewendeten Lösungsverfahren wird im darauffolgenden Kapitel erörtert. Kombinatorische Optimierungsprobleme lassen sich wie folgt unterteilen in:

- *Reihenfolgeprobleme*  
Hier wird die Besuchs- oder Bearbeitungsreihenfolge von z.B. Maschinen festgelegt; Bekanntestes Beispiel ist das *Traveling Salesman Problem (TSP)*;
- *Gruppierungsprobleme*  
Beispiele dafür waren Losgrößen- oder Tourenprobleme;
- *Zuordnungsprobleme*  
Thematik dieses Problems sind die Zuordnungen zwischen Objekten (Arbeiter, Maschinen etc.);
- *Auswahlprobleme*  
Ermittlung einer oder mehrerer Teilmengen auszuwählender Objekte;

Im Anschluss werde ich aufgrund des gewählten Modellierungsansatzes für diese konkrete Aufgabenstellung auf zwei dieser Spezialprobleme näher eingehen.

## 2.4 Spezialfälle der kombinatorischen Optimierung

In der Operations Research gibt eine Vielzahl an Spezialproblemen der linearen oder linear ganzzahligen Optimierung. Unter Rücksichtnahme meiner konkreten Aufgabenstellung, lässt sich das Entscheidungsmodell der Fallstudie aufgrund zweier verschiedener Optimierungsansätze aufstellen. Die Grundlagen dazu möchte ich im Folgenden grob umreißen.

### 2.4.1 Standortplanung (Facility Location Problem)

Vorweg sei zu erwähnen, dass bei der Ermittlung möglichst günstiger Standorte für z.B. Produktionsstätten, Lager oder sonstige Einrichtungen vielfältige quantitative und qualitative Kriterien zu berücksichtigen sind. Folgende Parameter charakterisieren sowohl in der Literatur als auch in der Praxis ein solches Problem und dienen uns zur Modellierung in Standardform.

Ein Unternehmen/Firma beliefert  $n$  Kunden mit einem jeweiligen Bedarf  $b_j$  an einem Gut oder Produkt. Zwecks Minimierung der anfallenden Kosten plant das Unternehmen die Errichtung von Zweigstellen. Hierfür stehen  $m$  potentielle Standorte zur Verfügung. An jedem dieser Standorte  $i$  kann eine Zweigstelle mit einer Kapazität von  $a_i$  errichtet werden, welche natürlich auch Fixkosten in der Höhe von  $f_i$  Geldeinheiten verursachen. Bekannt sind außerdem noch die Transportkosten  $c_{ij}$  für die Belieferung des Kunden  $j$  von der Zweigstelle  $i$  mit einer Mengeneinheit des Gutes.

Gesucht werden jene Standorte, die die geringsten Gesamtkosten hinsichtlich des Transportes und der Errichtung aufweisen. Zur Lösung dieses Problems werden zwei verschiedene Variablentypen benötigt:

$x_{ij} \rightarrow$  Menge an Waren von Standort  $i$  an den Kunden  $j$

$y_i = \begin{cases} 1, & \text{Zweigstelle wird am Standort } i \text{ errichtet} \\ 0, & \text{sonst} \end{cases}$

Beim Aufstellen des Modells in Standardform unterscheidet man zwischen dem **kapazitierten Standardmodell** in seiner Form mit

$$\begin{aligned} \text{Minimiere} & \rightarrow \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} + \sum_{i=1}^m f_i y_i \\ \sum_{i=1}^m x_{ij} & = b_j, \quad \forall j = 1, \dots, n \\ \sum_{j=1}^n x_{ij} & \leq a_i y_i, \quad \forall i = 1, \dots, m \\ x_{ij} & \leq b_j y_i, \quad i = 1, \dots, m, j = 1, \dots, n \\ x_{ij} & \geq 0, y_i \in \{0,1\}, i = 1, \dots, m, j = 1, \dots, n \end{aligned}$$

Abbildung 2.2: Mathematisches Modell eines kapazitierten Standortproblems<sup>14</sup>

<sup>14</sup> Vgl.: Dempe, Schreier [2006], Operations research – Deterministische Modelle und Methoden, Int. Student Edition  
Wolfgang Grundmann [2002], Operations research – Formeln und Methoden, , 1. Auflage, Teubner Verlag

und dem unkapazitierten Standardmodell, bei dem die Kapazitäten der Zweigstellen beliebig groß gewählt werden können, in seiner Form mit

$$\begin{aligned}
 \text{Minimiere} & \rightarrow \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} + \sum_{i=1}^m f_i y_i \\
 \hline
 \sum_{i=1}^m x_{ij} & = 1, \quad \forall j = 1, \dots, n \\
 x_{ij} & \leq y_i, \quad i = 1, \dots, m, j = 1, \dots, n \\
 x_{ij} \geq 0, y_i & \in \{0,1\}, \quad i = 1, \dots, m, j = 1, \dots, n
 \end{aligned}$$

Abbildung 2.3: Mathematisches Modell eines unkapazitierten Standortproblems<sup>14</sup>

Zur exakten Lösungsfindung bieten sich zum Beispiel Enumeration-, Näherungs-, Verbesserungs- oder Austauschalgorithmen an, auf die ich im nächsten Kapitel eingehen werde.

#### 2.4.2 Tourenplanung (TPP)

Die Tourenplanung (engl. Vehicle Routing oder Vehicle Scheduling) befasst sich mit der Thematik, gegebene Liefer- oder Sammelaufträge mit dem vorhandenen Fuhrpark so auszuführen, dass geforderte Ziele erfüllt werden. Jedes eingesetzte Fahrzeug wird dabei über die Menge der auszuführenden Aufträge und deren Reihenfolge definiert. Allgemein unterscheidet man in der Tourenplanung zwischen knotenorientierten und kantenorientierten Tourenplanungsproblemen (TPP). Bei ersteren werden Aufträge betrachtet, deren Mittelpunkt auf die Belieferung einzelner Kunden fokussiert. Beispiele dafür wären Möbel- oder Haushaltsgeräteauslieferungen (Die Lieferadresse entspricht dabei einem Knoten). Solche knotenorientierten Tourenprobleme sind allgemein in der Literatur als *Traveling Salesman Problem (TSP)* bekannt. Bei kantenorientierten Tourenproblemen hingegen werden ganze Straßen, sogenannte Kanten, in Betracht gezogen. In der Praxis betrifft das etwa das Briefträger-, Müllentsorgung- oder Winterdienstproblem. Diese Art von Tourenproblemen wird in der Literatur allgemein als *Briefträgerproblem* abgehandelt. Da es sich, auf meine explizite Aufgabenstellung bezogen, um ein knotenorientiertes Tourenproblem handelt, werde ich nachstehend Grundlagen zu diesem Optimierungsproblem anführen. Wichtigster Teil bei der Tourenplanung ist die Modellierung des Problems. Dabei wird versucht, ein mathematisches Modell in seiner Standardform aufzustellen. Zunächst will ich das Problem allgemein beschreiben.<sup>15</sup>

<sup>15</sup> Vgl.: Dempe, Schreier [2006], Operations research – Deterministische Modelle und Methoden, Int. Student Edition  
Wolfgang Grundmann [2002], Operations research – Formeln und Methoden, , 1. Auflage, Teubner Verlag

Das Travelling Salesman Problem entspricht dem Problem eines Handlungsreisenden. Dabei sind  $n$  Orte mit  $(i=1, \dots, n)$  und die Entfernungen  $d_{ij}$  ( $i=1, \dots, n; j=1, \dots, n$ ) zwischen den einzelnen Standorten sowie zu dem Depot selbst gegeben.

- Welche Fahrrouten sollen die Fahrzeuge wählen?

Diese Situation spiegelt ein Reihenfolgeproblem wieder, da die zurückgelegte Strecke abhängig von der Reihenfolge der angefahrenen Standorte minimiert werden soll. Es wird die kürzeste Rundreise gesucht, bei der jeder Ort genau einmal berührt wird.

Die Entscheidungsvariable wird wie folgt definiert:

$$x_{ij} = \begin{cases} 1, & \text{falls Tour von } i \text{ nach } j \\ 0, & \text{sonst} \end{cases}$$

Das mathematische Modell des TSP in seiner Standardform und Matrixschreibweise lautet:

$$\begin{aligned} \text{Minimiere } Z &= \sum_{i=1}^n \sum_{j=1}^n d_{ij} x_{ij} \\ \sum_{j=1}^n x_{ij} &= 1, \quad \forall i = 1, \dots, n \\ \sum_{i=1}^n x_{ij} &= 1 \quad \forall i = 1, \dots, n \\ u_i - u_j + nx_{ij} &\leq n - 1 \quad \forall 2 \leq i \neq j \leq n \\ x_{ij}, u_i, u_j &\geq 0 \end{aligned}$$

Abbildung 2.4: Mathematisches Modell eines Tourenproblems<sup>15</sup>

Würde die letzte Restriktion fehlen, dann wäre es ein simples Zuordnungsproblem. In diesem Fall würden sogenannte Kurzzyklen entstehen. Um solche Kurzzyklen von vornherein auszuschließen wird eine neue Variable  $u$  mit  $(i=1, \dots, n)$  eingeführt. Dieses kombinatorische Optimierungsproblem ist **NP-schwer**. Für die Lösung solcher diskreten schwierigen Optimierungsaufgaben existieren drei verschiedene Näherungsalgorithmen: die vollständige Enumeration, die Branch-and-Bound-Methode und das heuristische Lösungsverfahren.

### 2.4.3 Location-Routing-Problem (Standort-Tourenproblem)

Die Verteilung bzw. der Vertrieb von Produkten an Endkonsumenten zählt zu einer der bedeutendsten Tätigkeiten eines Produktionsunternehmens. In Anlehnung an eine Veröffentlichung von Srivastava und Benton im Jahr 1990 belaufen sich die Gesamtkosten für Transport und Lagerung auf ungefähr 20% des Bruttosozialprodukts. Folglich können durch ein verbessertes Vertriebssystem wesentliche Einsparungen erzielt werden. Die Säulen eines funktionierenden Vertriebssystems bilden die Standortoptimierung und die Verteilung (Tourenoptimierung) von den Produkten an Konsumenten. Auf die vorliegende Aufgabenstellung konkretisiert, soll zum einen die optimale Standortwahl der temporären Einrichtungen (Werbbestände) evaluiert werden, und zum anderen die optimale Tourenplanung zur Kostenminimierung beitragen.

Da in vielen realen Situationen diese beiden Optimierungsprobleme von einander abhängig sind, ist es von essentieller Bedeutung, die Entscheidungsprobleme gleichzeitig in Betracht zu ziehen. Diese Thematik bildet die Grundlage eines kombinierten Standort-Tourenproblems. Der Ausdruck „Standort-Touren-Problem“ kann häufig zu Mißverständnissen bzw. Fehlinterpretationen führen. Es handelt sich hierbei nicht etwa um ein einziges, eindeutiges Problem, wie beispielsweise bei einem Weber oder Travelling Salesman Problem, sondern um eine Gruppe von Optimierungsproblemen innerhalb der Standortplanung. So gesehen kann die folgende Definition von Bruns im Jahr 1998

*„Unter einem Standort-Touren-Problem versteht man die Standortplanung unter Berücksichtigung von Tourenplanungsaspekten.“<sup>16</sup>*

als Referenz genommen werden. Diese Definition folgt aus einer hierarchischen Betrachtungsweise: Das primäre Ziel ist es, das Standortproblem, auch Master Problem genannt, optimal zu lösen. Um dies wiederum zu erreichen, muss gleichzeitig das Tourenproblem, genannt Subproblem, gelöst werden. Dieser Umstand impliziert einen integrierten Lösungsansatz, der im Verlauf dieser Arbeit in Form eines gewichteten Mittels (Kriteriums) auch zum Einsatz kommt. Dieses Kriterium führt die beiden Optimierungsprobleme zusammen und charakterisiert deren Wechselbeziehung. Eine weitere wichtige Eigenschaft bei der Tourenplanung ist die Forderung nach der Existenz von Mehrfachstops. Wäre diese nicht gegeben, dann wäre auch die Wahl der Reihenfolge der anzufahrenden Standorte überflüssig.

In der Literatur wird Maranzana oft als einer der Pioniere auf dem Gebiet der Standort-Tourenoptimierung zitiert. Bereits im Jahr 1964 hat er eine Arbeit publiziert, die den wesentlichen Einfluß der Transportkosten auf die Standortwahl bewies. Es ist relativ einfach zu veranschaulichen, warum das Standort- und Tourenproblem Spezialfälle des kombinierten Problems sind. Würden beispielsweise alle Standorte vom Lager direkt angefahren werden, dann liegt ein einfaches Standortproblem vor. Wären die Standorte schon vorgegeben, dann würde man in diesem Fall ein simples Tourenproblem vorfinden.

Vom wirtschaftlichen Standpunkt aus, ist das Standort-Tourenproblem für das Vertriebsmanagement ein wesentlicher Bestandteil. Aus mathematischer Sicht hingegen, wird das Modell als Kombinatorisches Optimierungsproblem behandelt.

Wie die zuvor angeführten Spezialprobleme, ist das kombinierte Standort-Tourenproblem ein NP-hartes Problem.

---

<sup>16</sup> Zitat: Gabor Nagy, Said Salhi [2006], Location-routing – Issues, models and methods, European Journal of Operational Research, 177 (2007) 649-672

Um einerseits akzeptable Rechenzeiten und andererseits optimale Ergebnisse für das vorliegende Optimierungsproblem erzielen zu können, werde ich im Verlauf der Arbeit geeignete Lösungsverfahren behandeln. In Anlehnung an die Arbeit von Min, Jayaraman und Srivastava kann das Standort-Tourenproblem nach folgenden Aspekten strukturiert und klassifiziert werden:<sup>17</sup>

Kriterium	Ausprägung
<b>Klassifikation des LPR's hinsichtlich problembezogenen Aspekte</b>	
Hierarchische Ebene	Einfach
	Mehrfach (zwei Ebenen)
Nachfrage/Angebotscharakteristika	Deterministisch
	Stochastisch
Zahl der Einrichtungen	Eine
	Mehrere
Größe des Fuhrparks	Ein LKW/PKW
	Mehrere LKW/PKW
Kapazität des Fuhrparks	Unbeschränkt
	Beschränkt
Kapazität der Einrichtungen	Unbeschränkt
	Beschränkt
Ebene der Einrichtungen	Primär
	Sekundär
Planungszeitraum	Statisch (eine Periode)
	Dynamisch (mehrere Perioden)
Zeitfenster	Keine deadline
	Bewegliche deadline
	Strikte deadline
Zielfunktion	Eine Zielfunktion
	Mehrere Zielfunktionen
Modelldaten	Hypothetisch
	Real
<b>Klassifikation des LPR's hinsichtlich lösungsbezogenen Aspekte</b>	
Exakte Lösungsverfahren	Branch and Bound
	Dynamisch
	Ganzzahlig
	Nichtlinear
Heuristische Lösungsverfahren	Location zuerst, dann Routing
	Routing zuerst, dann Location
	Savings
	Improvment

Abbildung 2.5: Klassifikation eines LPR's<sup>17</sup>

<sup>17</sup> Vgl.: Min, Jayaraman, Srivastava [1997], Combined location-routing problems – A synthesis and future research directions, European Journal of Operational Research, 108(1998) 1-15  
 Dilek Tuzun, Laura I. Burke [1998], A two-phase tabu search approach to the location-routing problem, European Journal of Operational Research, 116(1999) 87-99  
 Gabor Nagy, Said Salhi [2006], Location-routing – Issues, models and methods, European Journal of Operational Research, 177 (2007) 649-672

## 2.5 Lösungsverfahren der kombinatorischen Optimierung

### 2.5.1 Allgemein

Die Kernaufgaben der Operations Research liegen insbesondere im Entwurf eines Entscheidungsmodells und in der Ermittlung einer optimalen Lösung.

*„Allgemein versteht man unter dem Begriff Lösung eine bestimmte Handlungsalternative oder einen Handlungsplan, der im Stande ist, ein Entscheidungsproblem zu lösen. Respektive bedeutet der Ausdruck „optimale Lösung“ jene Lösung, die zu einer bestmöglichen Eliminierung des Problemzustandes führt. Betrachten wir ein Optimierungsproblem, welches eine zu maximierende Zielfunktion  $F$  fordert, dann würde die optimale Lösung den höchstmöglichen Zielfunktionswert  $F(x^*)$  aufweisen. Sinngemäß kann dies auch auf ein zu minimierendes Problem umgelegt werden.“<sup>18</sup>*

Je nach Anforderungsprofil werden darüber hinaus auch quantitative Erklärungsmodelle zur Entscheidungsunterstützung eingesetzt, die zu einer Ergebnisermittlung und Bewertung der jeweils einzelnen Alternativen beitragen. Die Operations Research versucht also jene zulässige Handlungsalternative zu ermitteln, die einem bestimmten vorgegebenen Ziel am besten von allen Alternativen entspricht (→ Optimierung). Die Voraussetzungen dafür sind:



Abbildung 2.6: Voraussetzungen zum Erhalt optimaler Lösungen<sup>19</sup>

Wenn nur wenige Handlungsalternativen zur Verfügung stehen, dann kann beispielsweise eine **vollständige Enumeration** durchgeführt werden. Dies bedeutet, dass alle Alternativen explizit aufgestellt und bewertet werden. Im Anschluß kann die optimale Alternative ausgewählt werden. Bei einer großen Anzahl an Handlungsalternativen wäre dieses Vorgehen verständlicherweise zu ineffizient.

<sup>18</sup> Zitat: WISU-Zeitschrift [Juni 2003], Lange Verlag Düsseldorf

<sup>19</sup> Vgl.: Werners Brigitte [2008], Grundlagen des Operations Research, Springer Verlag

In diesem Fall werden mathematische Optimierungsmodelle entworfen und verwendet, die unter Einsatz von **Optimierungsalgorithmen** die optimale Lösung berechnen. Ein Algorithmus (Lösungsverfahren) ist eine allgemeine Verarbeitungsvorschrift, die meistens mittels eines ablauffähigen Programms in einer adäquaten Programmiersprache zur Lösung eines Entscheidungsproblems führt. Die Effizienz eines Algorithmus ist dabei abhängig von dessen Laufzeit und Speicherbedarf und ist zentrales Thema der Komplexitätstheorie.

*„Lässt sich der Rechenaufwand eines Algorithmus durch ein von der Problemgröße anhängiges Polynom nach oben beschränken, so spricht man von einem effizienten Verfahren“.<sup>20</sup>*

Lineare Optimierungsprobleme oder Kürzeste-Wege-Probleme entsprechen in der Regel solch einer Struktur und lassen sich daher auch polynomial lösen. Für alle anderen Verfahren – nichteffizienten Verfahren – würden die Rechenzeit und die Lösungsmenge mit zunehmender Problemgröße exponentiell wachsen. Diese Probleme zählen zur Klasse der **NP-schweren** Probleme und werden mittels der kombinatorischen Optimierung gelöst. Für jede dieser Gruppen gibt es unterschiedliche Lösungsverfahren:

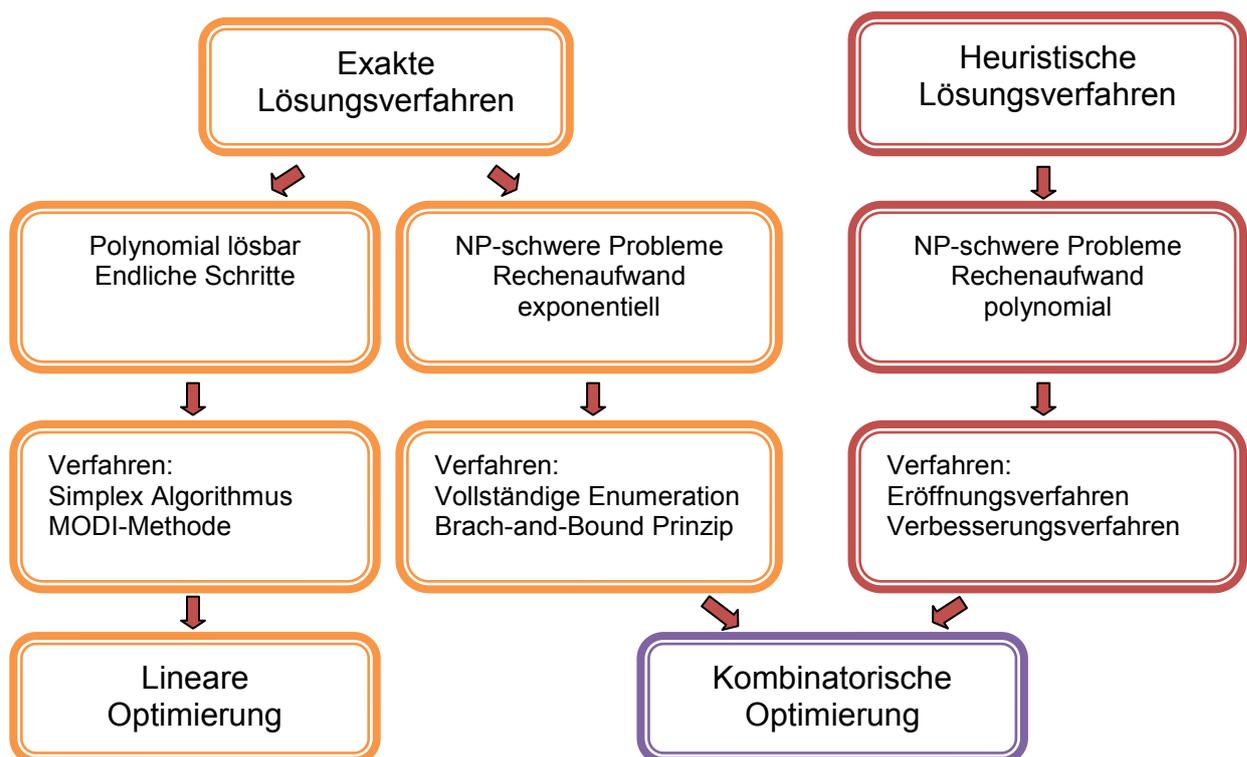


Abbildung 2.7: Systematische Darstellung der Lösungsverfahren

Unter heuristischen Lösungsverfahren versteht man eine systematische Vorgangsweise oder die Verwendung eines Algorithmus, die zu einem möglichst guten Ergebnis führen. Wie sich daraus gleich erkennen lässt, kann nicht garantiert werden, dass die gefundene Lösung auch der optimalen Lösung entspricht. Allerdings zeigt der Einsatz solcher Heuristiken einen polynomialen Rechenaufwand.

<sup>20</sup> Zitat: WISU-Zeitschrift [Juni 2003], Lange Verlag Düsseldorf

### 2.5.2 Exakte Lösungsverfahren

Zu den exakten Lösungsverfahren zählen die vollständige Enumeration (erschöpfende Aufzählung) und die Branch-and-Bound-Methode (B&B). Grundgedanke bei der vollständigen Enumeration ist die komplette Aufzählung aller zulässigen Lösungen. Diese Methode ist nur für relativ kleine  $n$  (=Entscheidungsvariablen) geeignet. Anders als bei diesem Lösungsverfahren zielt die Branch-and-Bound-Methode auf eine sukzessive Zerlegung des zulässigen Bereiches in Teilmengen ab. Dies wird mit der Hilfe einer sogenannten Baumstruktur (Suchbaum, Entscheidungsbaum) unterstützt. Das Verfahren besteht aus zwei Komponenten, der Verzweigung (Branching) sowie der Schrankenberechnung und dem Ausloten (Bounding). Bei der Verzweigung werden solche Teilmengen geschaffen, in denen die optimale Lösung liegt und sonstige Teilmengen, die keine Lösung enthalten. Als ersten Schritt formuliert man eine Verzweigungsregel zur Auftrennung von Mengen zulässiger Lösungen – Branching. Als weiteren Schritt wird eine nächste Vorschrift zur Berechnung der unteren und oberen Schranke der Zielfunktion in den Teilmengen formuliert – Bounding. Jeder Variable der Optimierungsaufgabe werden drei verschiedene Zustände zugeordnet: gesetzt – gesperrt – nicht entschieden. Mit diesen Regeln wird eine effiziente Abarbeitung des Entscheidungsbaumes angestrebt. Für die Auswahl der Zweige im Entscheidungsbaum gibt es drei unterschiedliche Suchstrategien:

- FIFO – first in first out
- LIFO – last in first out
- LLB – least lower bound

Ziel ist es, alle im B&B-Baum entstehenden Teilprobleme vollständig zu verzweigen und auszuloten, um so eine optimale Lösung zu erhalten. Da die exakten Lösungsverfahren auf die vorliegende Fallstudie nicht anwendbar sind, werde ich auf diese Verfahren auch nicht mehr näher eingehen und gleich zu den Heuristiken übergehen.<sup>21</sup>

### 2.5.3 Heuristische Lösungsverfahren

Im vorhergegangenen Abschnitt wurde dargelegt, dass die Anwendung klassischer exakter Lösungsverfahren stark beschränkt ist und folglich bei komplexen schweren Optimierungsproblemen zu keinem Lösungsalgorithmus mit polynomialer Zeitkomplexität führt. Bei solchen NP-schweren Problemen würde daher die Anwendung exakter Verfahren oft jedes wirtschaftlich noch vertretbare Maß an Rechenaufwand übersteigen. Um dennoch hochwertige Lösungen mit realitätsverträglichem Rechenaufwand ermitteln zu können, ist der Einsatz von heuristischen Optimierungsverfahren unabdingbar.

Probleme in der Operations Research bei denen es zum Einsatz solcher Verfahren kommt, sind beispielsweise Reihenfolge-, Zuordnungs-, und Gruppierungsprobleme der kombinatorischen Optimierung.

---

<sup>21</sup> Vgl.: Stephan Dempe, Heiner Schreier [2006], Operations research – Deterministische Modelle und Methoden, International Student Edition, 1. Auflage  
Wolfgang Grundmann [2002], Operations research – Formeln und Methoden, , 1. Auflage, Teubner Verlag

Ein sehr bekanntes Anwendungsbeispiel in diesem Zusammenhang ist das **Traveling Salesman-Problem (TSP)**, bei dem ein Handlungsreisender seine Kunden in streckenminimaler bzw. kostenminimaler Reihenfolge besuchen möchte. Mit Hilfe geeigneter Regeln und Vorschriften wird bei einem heuristischen Verfahren nur eine Teilmenge und nicht der vollständige zulässige Bereich abgesucht. Deshalb können solche Verfahren auch nicht garantieren, dass die „beste“ gefundene Lösung auch die optimale Lösung ist. Sie liefern also im Allgemeinen **suboptimale Lösungen**. Es ist auch möglich, dass Heuristiken, selbst wenn eine zulässige Lösung vorhanden ist, abbrechen, noch bevor sie überhaupt eine Lösung gefunden haben. Heuristische Verfahren sind in der Regel aber auf das spezielle Problem zugeschnitten und bieten schnelle, qualitative Lösungen. Sie können in die folgenden Gruppen klassifiziert und auf ihre Leistungsfähigkeit beurteilt werden:

### (1) Eröffnungsverfahren

- Häufig in Form eines Greedy-Algorithmus

### (2) Lokale Such- bzw. Verbesserungsverfahren

- z.B. Iterated Local search, Simulated-Annealing-Methode

### (3) Relaxionsbasierte Verfahren

- z.B. LP-Relaxion, Lagrange-Relaxation

### (4) Unvollständig ausgeführte Optimierungsverfahren

- Anwendung bei der Branch-and-Bound-Methode

## Eröffnungsverfahren

Eröffnungsverfahren dienen der Bestimmung einer zulässigen (Start-)Lösung eines Optimierungsmodells bzw. der zugehörigen Modellinstanz. Häufig wird die zulässige Lösung durch sukzessive Aufnahme von Lösungselementen konstruiert. Man kann diese Ansätze grob in Uniformierte Verfahren, bei denen durch ihren starren Ablauf auf das Ausnutzen konkreter Problemdaten verzichtet wird, in Greedy- und in Look Ahead-Heuristiken (Vorausschauende Verfahren) unterscheiden. Ein Beispiel für ein uniformiertes Verfahren ist die bekannte Nordweststreckenregel.

Greedy Verfahren trachten in jedem Konstruktions- oder Verfahrensschritt nach dem bestmöglichen Zielfunktionswert (der damit erreichbaren Teillösung) und bestmöglicher Erfüllung von Nebenbedingungen (Ausschöpfung aller Kapazitäten). Bei diesem Verfahren werden zur Ermittlung einer zulässigen Anfangslösung unterschiedliche Methoden wie etwa die Nearest-Neighbour-, Zeilenminimum-, Spaltenminimum-, oder Matrixminimum-Regel angewendet. Im Falle eines Travelling Salesman Problems beginnt das Nearest Neighbour Verfahren mit dem ersten Knoten und nimmt in jedem weiteren Schritt einen neuen Knoten, welcher die geringste Entfernung zum zuletzt eingefügten Knoten aufweist, in die Tour mit auf. Der Nachteil dabei ist, dass die letzten aufzunehmenden Knoten oftmals sehr weit voneinander entfernt liegen, sodaß sich eine unnötig lange Tour ergibt.

Vorausschauende Verfahren dagegen, versuchen diese „Kurzsichtigkeit“ dadurch zu umgehen, daß sie in jedem Schritt abschätzen, welche Auswirkungen die aktuelle Zuordnung eines Lösungselements auf die in nachfolgenden Schritten noch erzielbare Lösungsgüte besitzt.<sup>21</sup>

## Lokale Such- oder Verbesserungsverfahren

Lokale Suchverfahren basieren auf dem Prinzip der sukzessiven Verbesserung durch wiederholte (iterative) Modifikationen vorhandener Problemlösungskandidaten. Dabei wird im Verfahrensablauf jeweils genau eine Lösung als die aktuelle Lösung betrachtet. Die in einer Iteration durchzuführende Veränderung an der aktuellen Lösung wird durch die Definition einer Nachbarschaftsrelation zwischen Lösungen im Lösungsraum bestimmt. Es kommt zur unterschiedlichen Anwendung spezifizierter Transformationsvorschriften, die sich grob in drei Kategorien unterteilen lassen:

- Veränderung einer Lösung an genau einer Stelle wie z.B. Veränderung des binären Lösungsvektor von 0 auf 1 oder umgekehrt
- Vertauschen von Elementen
- Verschieben von Elementen

So wird in einem Suchschritt durch eine geringe Veränderung eine ähnliche Lösung aus der jeweiligen Nachbarschaft im Lösungsraum als die neue aktuelle Lösung bestimmt.

Neben der Nachbarschaftsdefinition lassen sich Verfahren hinsichtlich einer bestimmten Strategie zur Untersuchung von Nebenbedingungen und der **Zugauswahl** unterscheiden. Je nach Strategie kann dies zu einem hohen Maß an Intensivierung im Suchprozess führen. Es stellen sich also folgende Fragen:

In welcher Reihenfolge sollen die Nachbarschaftslösungen untersucht werden?

- Zufällig oder systematisch?

Zu welcher Nachbarlösung soll übergegangen werden, um von ihr aus in der nächsten Iteration die Suche fortzusetzen?

- First fit - erste gefundene bessere Nachbarlösung wird genommen oder Best fit - Nachbarschaft wird vollständig untersucht und die beste Möglichkeit realisiert z.B. Greedy Strategie

Sobald in der aktuellen Nachbarschaft keine Lösung mehr mit besserem Zielfunktionswert erreicht werden kann, wird die Suche abgebrochen und die aktuelle Lösung stellt ein lokales Optimum dar. Man spricht von sogenannten reinen Verbesserungsverfahren und in manchen Ausnahmefällen kann in diesem Zusammenhang ein lokales Optimum auch ein globales darstellen.

Zusätzlich gibt es gewisse Techniken, die zur teilweisen Überwindung lokaler Optima führen. Dabei ist auch die in der Diplomarbeit verwendete **Iterated Local Search** eine sehr hilfreiche und effiziente Technik. Hier werden viele Neustarts mit unterschiedlichen Startlösungen vorgenommen und die beste Lösung herausgefiltert. Andere Strategien wiederum versuchen ein solches lokales Optimum zu verlassen, indem sie Züge erlauben, die zwischenzeitlich schlechtere Zielfunktionswerte zulassen. Hierzu zählen Simulated Annealing, Tabu Search und Genetische Algorithmen.

Der Unterschied zwischen reinen Verbesserungsverfahren und solchen, die auch verschlechternde Nachbarlösungen zulassen, wird in der nächsten Abbildung verdeutlicht.

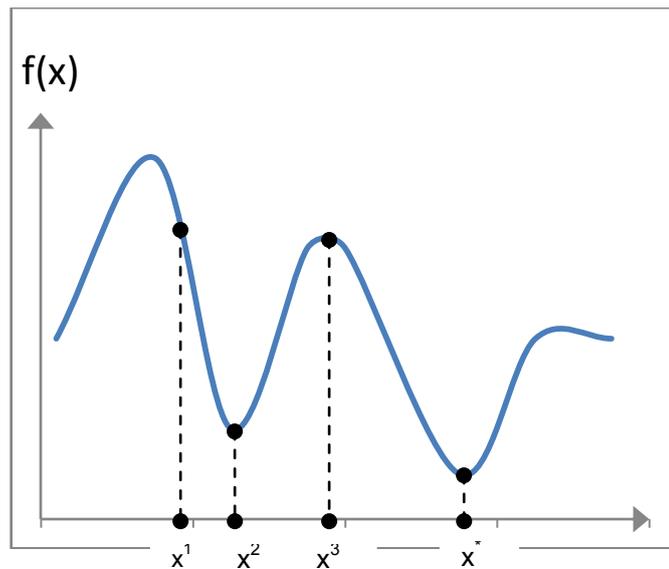


Abbildung 2.8: Lokale und globale Optima

- Anfangslösung:  $x^1$  (Minimierende Funktion)
- Lokales Minimum:  $x^2$  ( $y^*$  wird bei Verbesserungsverfahren nicht gefunden)
- Lokale Suchverfahren: Von  $x^2$  zum schlechteren Nachbar  $x^3$  und schließlich zum globalem Minimum  $x^*$

### Relaxationsbasierte Verfahren

Ein komplexes Optimierungsmodell kann durch eine Relaxation vereinfacht werden. Dabei werden komplizierte Nebenbedingungen entweder abgeschwächt oder überhaupt weggelassen. Durch die Lösung einer solchen Relaxation erhält man bei z.B. einer maximierenden Zielfunktion eine obere Schranke für den optimalen Zielfunktionswert.

Vernachlässigt man die Ganzzahligkeitsforderung von Variablen in einem linearen Modell, dann entsteht die LP-Relaxation. Und durch Weglassen von Nebenbedingungen ohne Rücksicht darauf, dass die Zielfunktion verletzt wird, kommt man zur Lagrange-Relaxation.

### 3 PROBLEMMODELLIERUNG

„Ein Modell ist ein zweckorientiertes, ggf. vereinfachtes Abbild eines Ausschnitts der Realität, welches hinsichtlich der interessierenden Zusammenhänge strukturähnlich oder strukturgleich ist. Erfolgt die Abbildung derart, dass jedem Element bzw. jeder Beziehung zwischen Elementen des Urbildes ein Element bzw. eine Beziehung im Modell gegenübersteht und umgekehrt, dann spricht man von einem isomorphen Modell. Da in der Realität die realen Systeme jedoch sehr komplex sind, werden in der Operations Research in der Regel homomorphe (strukturähnliche) Modelle verwendet.“<sup>22</sup>

Wie bereits im Kapitel 2 erläutert, können Modelle in drei verschiedene Gruppen eingeteilt werden. Ich befrage mich ausschließlich mit dem Entscheidungsmodell, welches zur Vorbereitung und zum Treffen von Entscheidungen dient. Zum besseren strukturellen und systematischen Verständnis folgend die graphische Darstellung eines Ablaufschemas des Entscheidungsprozesses an:

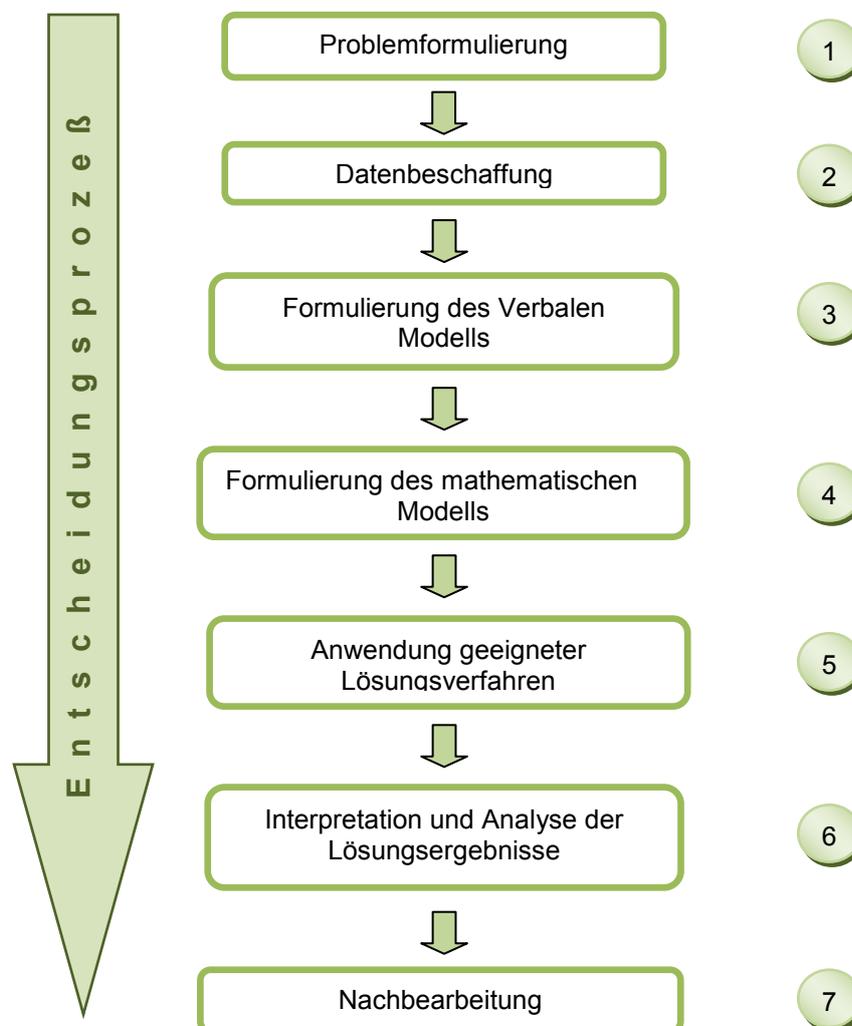


Abbildung 3.1: Struktur eines Entscheidungsprozesses

<sup>22</sup> Zitat: WISU-Zeitschrift [Juni 2003], Lange Verlag Düsseldorf

### 3.1 Problemformulierung

Die Contacts GmbH, eine renommierte Werbe- und Promotionagentur mit Sitz in Wien, wurde von einem Milchherstellungsunternehmen mit folgendem Projekt beauftragt.

Als eine der innovativsten Molkereien Europas setzt die Niederösterreichische Molkerei AG (kurz NÖM) besonders auf Neu- und Weiterentwicklung ihrer Produkte. Um den hohen Standards des Unternehmens gerecht zu werden, versucht das Management Konsumententrends aufzuspüren und ständig neue, hochwertige Produkte zu entwickeln. Nachdem sich die Marketingabteilung des Unternehmens für die Entwicklung eines neuen „trendigen“ Produktes ausgesprochen hat, wurden die notwendigen Produktionsschritte in die Wege geleitet. In der Regel arbeiten solche Unternehmen mit unterschiedlichen Produktionszyklen. Der erste Zyklus gilt als sehr entscheidend für den restlichen Verlauf der Produkteinführung in den Markt. Es werden bestimmte Mengen auf spezielle Vertriebskanäle in Umlauf gebracht, um einerseits ein qualitatives und quantitatives Feedback von den Kunden zu erhalten und andererseits auf die Präsenz des neuen Produktes aufmerksam zu machen. Außerdem können nach der ersten Produktionslinie gegebenenfalls Veränderungen an der Rezeptur, der Verpackung oder dem Design vorgenommen werden. Diese Methode wird im Fachjargon *Sampling* genannt.

Zusammen mit dem Management hat sich die Marketingabteilung für eine spezielle Maßnahme der Verkaufsförderung entschieden, bei der Warenproben gratis durch Promotoren an Kunden übergeben werden. Der Planungszeitraum für dieses Projekt ist mit einer Woche angesetzt. In der ersten Produktionslinie wurden ca. 150.000 Einheiten produziert. Die erste Zielvorgabe die die NÖM Molkerei aufgestellt hat, befaßt sich mit der Maximierung vorhandener Ressourcen im Bereich Werbung und Infiltration in den Markt. Konkret sollen die möglichen Sichtkontakte potentieller Kunden maximiert werden. Es sollen mindestens 400.000 Sichtkontakte auf diese Art der Verkaufsförderung erzielt werden. In der ersten Phase sollen alle Einheiten ausschließlich in Wien verteilt werden. Dabei ist es dem Unternehmen von großer Bedeutung, dass jene Standorte bevorzugt angefahren werden, die die größte Kundenfrequenz aufweisen. Um eine Diversität unter den Standorten zu erreichen, soll einerseits kein Standort mehr als drei Mal pro Woche und andererseits vier verschiedene Standorte pro Tag angefahren werden. Eine weitere Zielvorgabe des Unternehmens ist es, die Kosten die bei der Durchführung der Touren entstehen, so niedrig wie möglich zu halten.

Zusammenfassend können nun folgende Fragen zur Erreichung der Zielsetzung und unter den gegebenen Restriktionen formuliert werden:

(1) Welche Standorte sollen an welchen Wochentag angefahren werden?  
**Standortplanung**

(2) In welcher Reihenfolge sollen die Standorte angefahren werden?  
**Tourenplanung** - Kostenminimierung

## 3.2 Datenbeschaffung und Aufbereitung

Mit dieser Entscheidungsstruktur muss nun die Agentur die optimalen Lösungen finden und dem Management präsentieren. Aufgrund der mehrjährigen Erfahrung in diesem Geschäftsfeld, sind dem Unternehmen bestimmte Werte bekannt. Die folgenden Daten wurden mir von der Firma Contacts GmbH für diese Arbeit zur Verfügung gestellt und bilden den Ausgangspunkt für meine weitere Modellierung.

Standort							
Nr.:	WIEN	Mo	Di	Mi	Do	Fr	Sa
<b>1</b>	<b>1010 Wien, Schwedenplatz</b>						
	Mittlere Absatzmenge	5000	4500	4500	5000	5500	5000
	Sichtkontakte	17500	15750	15750	17500	19250	17500
<b>2</b>	<b>1020 Wien, Stadioncenter</b>						
	Mittlere Absatzmenge	3500	3500	3500	4000	5500	5500
	Sichtkontakte	12250	12250	12250	14000	19250	19250
<b>3</b>	<b>1020 Wien, Praterstern</b>						
	Mittlere Absatzmenge	3500	3500	3500	3500	4000	4000
	Sichtkontakte	12250	12250	12250	12250	14000	14000
<b>4</b>	<b>1030 Wien, Landstrasse / Rochusm.</b>						
	Mittlere Absatzmenge	3000	3000	3000	3000	3500	3500
	Sichtkontakte	10500	10500	10500	10500	12250	12250
<b>5</b>	<b>1040 Wien, Karlsplatz</b>						
	Mittlere Absatzmenge	3500	3500	3500	4500	4000	4000
	Sichtkontakte	12250	12250	12250	15750	14000	14000
<b>6</b>	<b>1070 Wien, Naschmarkt</b>						
	Mittlere Absatzmenge	1000	1000	1000	2000	4500	8000
	Sichtkontakte	3500	3500	3500	7000	15750	28000
<b>7</b>	<b>1090 Wien, Hauptuniversität</b>						
	Mittlere Absatzmenge	5000	4500	4500	4000	3000	2500
	Sichtkontakte	17500	15750	15750	14000	10500	8750
<b>8</b>	<b>1100 Wien, Reumannplatz</b>						
	Mittlere Absatzmenge	5000	5000	5000	5500	6000	6500
	Sichtkontakte	17500	17500	17500	19250	21000	22750
<b>9</b>	<b>1110 Wien, Gasometer</b>						
	Mittlere Absatzmenge	3000	3000	3000	3000	4000	4500
	Sichtkontakte	10500	10500	10500	10500	14000	15750
<b>10</b>	<b>1150 Wien, Mariahilferstrasse</b>						
	Mittlere Absatzmenge	4500	4000	4000	4000	6500	7000
	Sichtkontakte	15750	14000	14000	14000	22750	24500
<b>11</b>	<b>1160 Wien, Lugnercity</b>						
	Mittlere Absatzmenge	4500	4500	4500	4500	6000	5500
	Sichtkontakte	15750	15750	15750	15750	21000	19250
<b>12</b>	<b>1190 Wien, Q19 Shopping Center</b>						
	Mittlere Absatzmenge	3500	3500	3500	3500	3000	3000
	Sichtkontakte	12250	12250	12250	12250	10500	10500
<b>13</b>	<b>1200 Wien, Milleniumstower</b>						
	Mittlere Absatzmenge	4000	4000	4000	4000	5000	6000
	Sichtkontakte	14000	14000	14000	14000	17500	21000
<b>14</b>	<b>1210 Wien, Shopping City Nord</b>						
	Mittlere Absatzmenge	3500	3500	3500	3500	4000	4000
	Sichtkontakte	12250	12250	12250	12250	14000	14000
<b>15</b>	<b>1210 Wien, Floridsdorf U6</b>						
	Mittlere Absatzmenge	5000	4500	4500	4500	5500	5500
	Sichtkontakte	17500	15750	15750	15750	19250	19250
<b>16</b>	<b>1220 Wien, Dounauzentrum</b>						
	Mittlere Absatzmenge	5000	5000	5000	6000	7000	7000
	Sichtkontakte	17500	17500	17500	21000	24500	24500
<b>17</b>	<b>1220 Wien, Gewerbepark Stadlau</b>						
	Mittlere Absatzmenge	3500	3500	3500	4000	6000	7000
	Sichtkontakte	12250	12250	12250	14000	21000	24500

Tabelle 3.1: Sichtkontakte und mittlere Absatzmenge

**Sichtkontakte** können meistens nur geschätzt werden. Sichtkontakte entsprechen jener Kundenfrequenz, die an einem bestimmten Standort physisch gesichtet worden sind.

Die *mittlere Ansatzmenge* hingegen ist die Menge der potentiellen Käufer. Diese zwei Kennziffern sind für viele Unternehmen sehr bedeutend und können darüber entscheiden, ob ein Produkt in den Markt eindringt oder nicht.

In diesem Zusammenhang soll erwähnt werden, dass für Wien jene 17 Standorte ausgewählt worden sind, die die höchste Besucherfrequenz aufweisen. Nachdem die zweite Frage in Verbindung mit einem Tourenproblem steht, habe ich die Distanzen zwischen den Standorten und die Distanz zum Lager erhoben (siehe Tabelle 9). Jedem Standort wurde zum Zwecke der Erleichterung für die weiteren Berechnungsschritte eine Ziffer zugeordnet wie Abbildung 8 zeigt:

Legende	Standort
1	1010 Wien, Schwedenplatz
2	1020 Wien, Stadioncenter
3	1020 Wien, Praterstern
4	1030 Wien, Landstrasse / Rochusmarkt
5	1040 Wien, Karlsplatz
6	1070 Wien, Naschmarkt
7	1090 Wien, Hauptuniversität
8	1100 Wien, Reumannplatz
9	1110 Wien, Gasometer
10	1150 Wien, Mariahilferstrasse - Wbf
11	1160 Wien, Lugnercity
12	1190 Wien, Q19 Shopping Center
13	1200 Wien, Milleniumstower
14	1210 Wien, Shopping City Nord
15	1210 Wien, Floridsdorf U6 Endstation
16	1220 Wien, Donauzentrum
17	1220 Wien, Gewerbepark Stadlau

Tabelle 3.2: Standortzuweisungen

Distanz / km	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	L
von 1 nach	0	6	2	3	4	5	3	10	11	5	8	9	8	10	8	8	12	40
von 2 nach	6	0	3	5	6	7	6	11	5	7	9	9	11	14	10	8	12	38
von 3 nach	2	3	0	2	4	5	6	14	8	5	7	7	9	11	8	7	9	37
von 4 nach	3	5	2	0	2	3	4	11	8	4	6	8	10	12	11	9	13	36
von 5 nach	4	6	4	2	0	0,5	2	8	8	1	4	8	11	13	12	11	15	31
von 6 nach	5	7	5	3	0,5	0	2,5	8,5	8,5	1	2,5	8,5	11,5	13,5	12,5	11,5	15	32
von 7 nach	3	6	6	4	2	2,5	0	11	11	2	3,5	6	8	9	11	14	17	33
von 8 nach	10	11	14	12	8	8,5	11	0	7	8	9	11	13	14	12	13	17	30
von 9 nach	11	5	8	8	8	8,5	11	7	0	9	11	14	16	18	17	16	18	33
von 10 nach	5	7	5	4	1	1	2	8	9	0	1	6	12	15	14	17	19	31
von 11 nach	8	9	7	6	4	2,5	3,5	9	11	1	0	5	11	14	15	17	19	32
von 12 nach	9	9	7	8	8	8,5	6	11	14	6	5	0	4	10	11	18	20	46
von 13 nach	8	11	9	10	11	11,5	8	13	16	12	11	4	0	5	4	15	16	43
von 14 nach	10	14	11	12	13	13,5	9	14	18	15	14	10	5	0	2	14	17	45
von 15 nach	8	10	8	11	12	12,5	11	12	17	14	15	11	4	2	0	16	14	42
von 16 nach	8	8	7	9	11	11,5	14	13	16	17	17	18	15	14	16	0	6	40
von 17 nach	12	12	9	13	15	15	17	17	18	19	19	20	16	17	14	6	0	50

Tabelle 3.3: Entfernungen zwischen den Standorten

Folgende Diagramme geben einen groben Überblick über die Sichtkontaktverteilung pro Standort und die relativen Distanzen zum Lager:

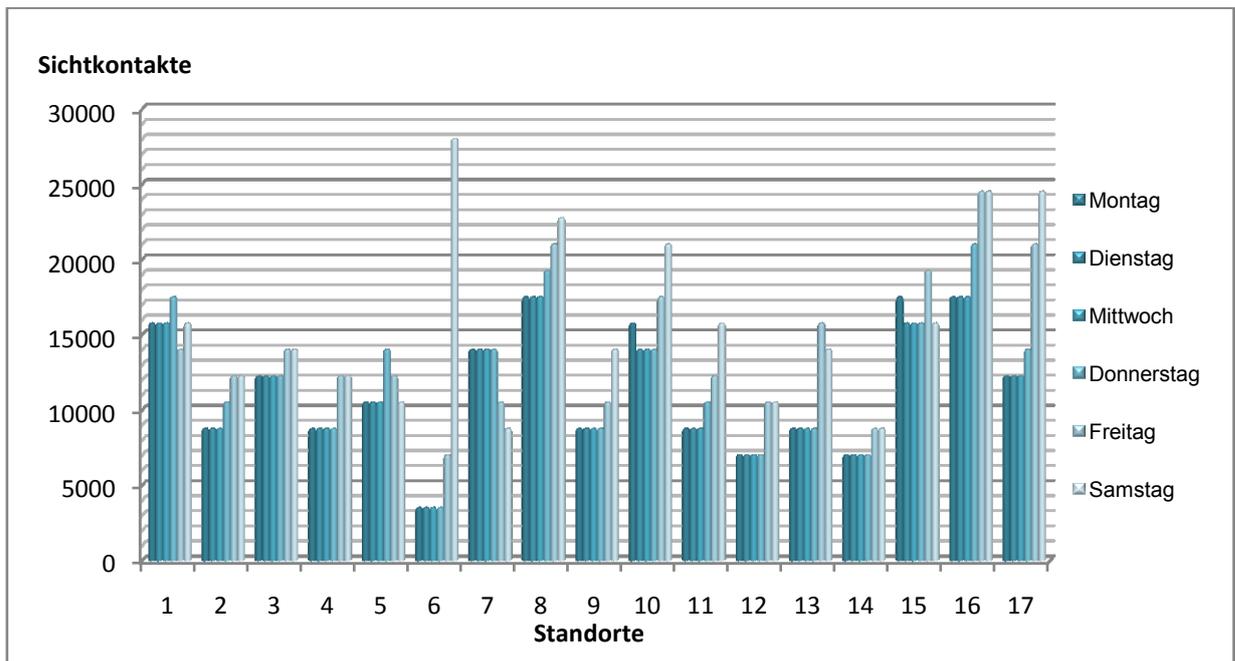


Diagramm 3.1: Sichtkontaktverteilung

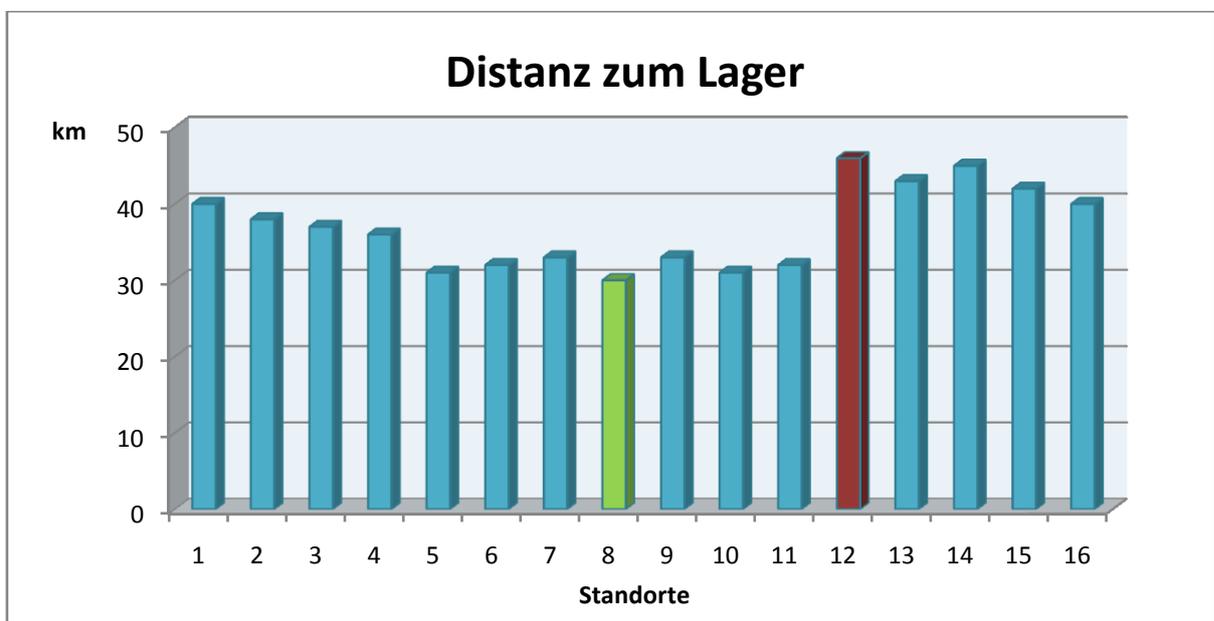


Diagramm 3.2: Lagerdistanzen

### 3.3 Verbales Modell

In Anbetracht der Problemformulierung kann man annehmen, dass es sich in diesem konkreten Fall nicht nur um ein einziges Optimierungsproblem handelt sondern um einen Mix aus einem Standort, Zuordnungs- und Tourenproblem. Diese Problematik kann mittels eines speziellen kombinierten Standort- und Tourenmodell sehr gut gelöst werden. Das Ziel ist, aus einer effizienten Standortaufteilung und Tourenoptimierung Kapital zu schlagen. Im Unterschied zu einem herkömmlichen, klassischen Standortproblem wird bei diesem kombinierten Modell die Reihenfolge der anzufahrenden Standorte berücksichtigt und mit dem übergeordneten Ziel der Sichtkontaktmaximierung in Beziehung gesetzt. Dabei ergibt sich folgendes Abhängigkeitsschema:

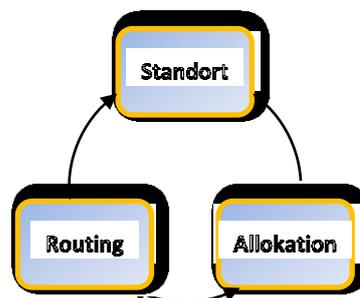


Abbildung 3.2: Abhängigkeiten in einem Location-Touring Problem<sup>23</sup>

Zu Beginn eines Modellierungsprozesses wird das Location-Routing Problem in problem- und lösungsbezogene Klassifikationen unterteilt und deren spezifische Ausprägungen definiert. In diesem konkreten Fall können nun folgende Informationen über das Optimierungsproblem und dem verwendeten Lösungsverfahren formuliert und konkretisiert werden:

Klasse	Klassifizierungsmerkmal	Ausprägung
<b>Problembezogen</b>		
I.	Hierarchischer Level	Einstufig
II.	Nachfragestruktur	(deterministisch) Keine Anforderung
III.	Zahl der Standorte	Multipel – 17 Standorte & Lager
IV.	Fuhrparkgröße	Ein LKW wird eingesetzt
V.	Fahrzeugkapazität	Unbeschränkt
VI.	Standortkapazität	Unbeschränkt
VII.	Zwischenlager	Nicht notwendig
VIII.	Planungshorizont	Einphasig – 1 Woche
IX.	Zeitfenster	Nicht spezifiziert – keine Deadline
X.	Zielfunktion	Mehrfach
XI.	Datentyp	Realistisch
<b>Lösungsbezogen</b>		
I.	Exakte Verfahren	Nicht effizient
II.	Heuristische Verfahren	Iterated Local Search

Tabelle 3.4: Klassifizierung des Location-Routing Problems

<sup>23</sup> Hokey Min, Vaidyanathan Jayaraman, Rajesh Srivastava [1997], Combined location-routing problems

Das Molkereiunternehmen hat zwei separate Projektziele vorgegeben. Zum einen wird die Maximierung der möglichen Sichtkontakte verfolgt und zum anderen will man die entstehenden Projektkosten so gering wie möglich halten. Für das Management weist keines der beiden Ziele eine höhere Priorität auf, das heißt, dass parallel für beide Zielvorgaben optimale Lösungen gefunden werden sollten. Wie im vorangegangenen Kapitel dargelegt, wird ein Optimierungsmodell in seiner Standardform von einer Zielfunktion, den Nebenbedingungen und der Nichtnegativitätseigenschaft beschrieben.

### 3.3.1 Zielfunktion

Nachdem das Management des Unternehmens in diesem konkreten Fall zwei unterschiedliche Zielvorgaben deklariert hat, muss allen voran versucht werden, diese beiden Zielfunktionen zu einer einzigen zusammen zu führen.

Aus diesem Grund wird von einem verschachtelten bzw. kombinierten Standort-Tourenproblem gesprochen. Würde beispielsweise die zweite Zielsetzung nicht existieren, dann wäre das Problem mittels einer klassischen Standortoptimierung einfach zu lösen. In diesem Fall jedoch sind die Kosten indirekt von der Wahl der Standorte abhängig, da die Standorte zueinander und zum Lager unterschiedliche Entfernungen aufweisen (siehe Tabelle 3.3). Eine weitere bedeutende Eigenschaft bei der Tourenplanung weist die Wahl der Reihenfolge der anzufahrenden Standorte auf. Um beide Zielfunktionen gleichermaßen optimal befriedigen zu können, wird ein gewichtetes Mittel (weiter unten im Text auch als gewichtetes Kriterium bezeichnet) eingeführt.

#### Projekt(fahrt)kosten

Die variablen Kosten pro zurückgelegten Kilometer werden von der Agentur mit  $c$  Geldeinheiten kalkuliert. Je weniger Kilometer gefahren werden, desto weniger Projektkosten fallen an. Hier handelt es sich um ein einfaches Tourenproblem, dessen Ziel es ist, jene Tour zu eruieren, welche die geringsten Gesamtkilometer aufweist. Die verschiedenen Standorte werden mit der Variable  $i$  bezeichnet. Pro Wochentag  $t$  sollen  $M$  Standorte angefahren werden. Folglich resultieren aus  $M$  anzufahrenden Standorten ( $M!$ ) mögliche Permutationen. Unter Permutation versteht man im Allgemeinen die Zahl der Reihenfolgen  $R$ . Da in der Aufgabenstellung das Lager zugleich den Ausgangs- und Endpunkt der Tour darstellt, wiederholen sich bestimmte Reihenfolgen. Diese sich wiederholenden Reihenfolgen werden gestrichen bzw. nicht miteinbezogen und wir erhalten  $R=(M!)/2$  zulässige Reihenfolgen. Wurde die optimale Route gefunden, wurden gleichzeitig auch die Projektkosten minimiert. In jeder Iteration des Local Search wird dieser Vorgang wiederholt und die Projektkosten neu ermittelt.

#### Maximierung der Sichtkontakte

Die Maximierung der Sichtkontakte ohne Einbindung bzw. Rücksichtnahme der zurückgelegten Kilometer würde das Ergebnis verfälschen und die Zielvorgaben nicht erreichen. Hier muss zuerst der Benefit, welcher sich aus den erzielten Gesamtsichtkontakten ergibt, ausgerechnet werden. Ein Sichtkontakt wird von dem Unternehmen mit  $b$  Geldeinheiten beziffert. Durch eine einfache Subtraktion der Projektkosten  $c$  vom Benefit  $b$  bekommt man das gewichtete Mittel. Dadurch wird eine einzige Zielfunktion ermittelt, bei welcher das Kriterium maximiert wird.

Die wichtigsten Variablen- und Indexbeschreibungen in übersichtlicher tabellarischer Form zusammengefaßt:

Erklärung	Variable	Ausprägung
Standort	$i$	$i=1,\dots,N$
Reihenfolge	$j$	$j=1,\dots,R$
Wochentag	$t$	$t=1,\dots,T$
Anzahl der Standorte, die pro Tag angefahren werden müssen	$M$	
Max. Anzahl der Besuche an einem Standort pro Woche	$B$	
Kumulierte Projektkosten pro Kilometer	$c$	
Benefit (Gewinn) pro Sichtkontakt	$b$	

Tabelle 3.5: Variablenbeschreibung

Für die Standardform des Modells werden verschiedene Entscheidungsvariablen benötigt. Diese definieren sich einerseits über die Wahl des Standorts und andererseits über die zu wählende Reihenfolge der Standorte. Folgende Entscheidungsvariablen sind notwendig;

$$\text{Variable } x_{it} = \begin{cases} 1, & \text{wenn der Standort } i \text{ am Tag } t \text{ angefahren wird} \\ 0, & \text{sonst} \end{cases}$$

$$\text{Variable } y_{jt} = \begin{cases} 1, & \text{wenn die Reihenfolge } j \text{ am Tag } t \text{ gewählt wird} \\ 0, & \text{sonst} \end{cases}$$

### 3.3.2 Nebenbedingungen

Nach der Formulierung der Zielfunktion und Entscheidungsvariablen werden noch die Restriktionen als Nebenbedingungen definiert:

1. Es müssen genau  $M$  Standorte pro Tag angefahren werden:

$$\sum_{t=1}^T \sum_{i=1}^N x_{it} = M$$

2. Ein Standort darf maximal  $B$  Mal pro Woche aufgesucht werden:

$$\sum_{i=1}^N \sum_{t=1}^T x_{it} \leq B$$

3. Nichtnegativitätsbedingung:

$$x_{it}, y_{jt} \in \{0,1\}$$

Für die konkret vorliegende Problem Instanz wird das allgemeine Modell durch folgende Werte ersetzt:

- Anzahl der verschiedenen Standorte wird mit  $N=17$  beziffert
- Die Woche besteht aus  $T=6$  Tagen
- Pro Tag sollen genau  $M=4$  Standorte angefahren werden
- Jeder Standort darf maximal  $B=3$  Mal besucht werden

## 4 LÖSUNGSVERFAHREN

### 4.1 Iterated Local Search

Mit einer zulässigen und zufälligen Startlösung beginnend, versucht dieses Verfahren iterativ in einer geeignet definierten „Nachbarschaft“ bessere Lösungen zu finden, solange bis keine Verbesserungen mehr möglich sind.

**Nachbarschaft** → Für jede zulässige Lösung  $s$  im Suchraum  $S$  bezeichnet  $N(s) \subseteq S$  die Nachbarschaft von  $s$ .

**Lokales Optimum** → Eine Lösung  $s^*$  ist genau dann lokal optimal bezüglich einer Nachbarschaftsstruktur  $N$ , wenn  $f(s^*) \leq f(s)$  für alle  $s \in N(s^*)$

Es existieren unterschiedliche Arten auf welchen Weg die Nachbarschaftsstruktur verändert werden kann. Diese werden durch die Nachbarschaftsoperatoren spezifiziert. In diesem speziellen Fall ist die Anzahl der anzufahrenden Standorte pro Tag fix vorgeben. Mit dieser Forderung war es nur möglich, einen „schon bestehenden“ Standort gegen einen „neuen“ Standort auszuwechseln, ohne diese Restriktion zu verletzen.

Zudem kann man bei einer lokalen Suche zwischen der First-Accept, erste beste Nachbarschaft wird genommen, und der **Best-Accept** Variante, der Lösungsraum wird vollständig abgesucht und die beste Lösung wird ausgewählt, entscheiden. Letztere Variante kommt in dieser Diplomarbeit zur Anwendung. Um die Theorie ein wenig verständlicher darzustellen möchte ich den Vorgang des Iterated Local Search anhand eines simplen Beispiels zeigen:

1. Finden einer Startlösung z.B.

$$\begin{array}{ccc} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{array}$$

2. Geringfügige Änderung der Lösung z.B.

$$\begin{array}{ccc} 0 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{array}$$

3. Lösungen vergleichen → falls besser dann merken, sonst zur nächsten Nachbarschaftslösung solange bis der Lösungsraum vollständig abgesucht wurde:

$$\begin{array}{ccc} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{array}$$

Nach diesem Durchgang wechselt man zur nächsten Spalte und beginnt nach dem gleichen Prinzip wieder von vorne. Da in unserem Fall einerseits anstatt einer 3x3 Matrix eine 17x6 Matrix vorliegt und andererseits genau vier Standorte angefahren werden müssen, gibt es pro Spalte  $4 \times 13 = 52$  Möglichkeiten, insgesamt also  $52 \times 6 (\text{Tage}) = 324$  mögliche Nachbarschaftslösungen. Aufgrund der Restriktion, dass ein Standort nicht mehr als drei Mal pro Woche angefahren werden darf, werden nicht alle dieser 324 Möglichkeiten auch zulässig sein. Mithilfe einer einfachen Bedingungsanweisung können solche unzulässigen Lösungen a priori ausgeschlossen werden. Wie im vorangegangenen Punkt beschreiben, handelt es sich

hier um ein kombiniertes Standort-Tourenproblem. Hierbei sei zu erwähnen, dass der Iterated Local Search ausschließlich bei der Wahl des optimalen Standortes angewendet wird, nicht hingegen bei dem Tourenproblem. Der Grund dafür liegt in der relativ kleinen Anzahl anzufahrender Standorte. Bei vier geforderten Standorten würden sich nur 12 ( $4!/2$ ) mögliche Reihenfolgen ergeben. Diese können für jede einzelne Nachbarschaftslösung vollständig aufgezählt, errechnet und miteinander verglichen werden (vollständige Enumeration). Ein gewichtetes Kriterium, welches die Kostenminimierung (Tourenoptimierung) mit der Sichtkontaktmaximierung (Standortoptimierung) verbindet, wird als Vergleichskennzahl herangezogen und bildet die Basis für den Iterated Local Search.

## 4.2 Softwareeinsatz

**Scilab** ist eine frei verfügbare Software von I.N.R.I.A. (Institut National de Recherche en Informatique et Automatique) die auf Basis von **Matlab** entwickelt worden ist. **Matlab** selbst ist ein Softwarepaket für numerische Berechnungen und für die Visualisierung von Daten im technisch-wissenschaftlichen Bereich. Es wurde in den 70er Jahren an der University of New Mexico und der Stanford University entwickelt, um Kurse aus linearer Algebra und Numerische Analysis zu unterstützen. Der Name ("Matrix Laboratory") erinnert noch heute daran. **Matlab** ist eine universelle Software, welche in weiten Bereichen der angewandten Mathematik eingesetzt wird.

Dabei stehen numerische Rechnungen und die Darstellung von Zahlen im Vordergrund. Der Grundbaustein ist eine Matrix, deren Dimensionen nicht explizit definiert werden müssen. Dadurch können numerische Probleme innerhalb kürzester Zeit optimal gelöst werden.

Die Programmiersprache (Syntax) unterscheidet sich zu **Scilab** nur unwesentlich, einzige Ausnahme sind die Grafikbefehle. Das Anwendungsgebiet dieser Software ist mathematischer und wirtschaftlicher Natur und umfasst folgende Teilbereiche:

- **Lösungen linearer Gleichungssystem**
- Berechnung von Eigenwerten und Eigenvektoren
- Fourier-Transformation
- Lösung von Differentialgleichungen
- **Optimierungsverfahren**
- Lösung nichtlinearer Gleichungssysteme
- **Erzeugung von Zufallszahlen**
- Methoden der linearen Algebra für optimale Steuerungen

**Scilab** arbeitet mit einer Matlab-ähnlichen Programmiersprache, die Zuweisungen, Operatoren, Funktionen, logische Operationen und Schleifen enthält. Die in **Scilab** benutzten Datentypen gestatten deutlich kürzeres und übersichtlicheres Programmieren als z.B. Fortran oder C++.

**Scilab** arbeitet mit unterschiedlichen Datentypen, die im Wesentlichen als Matrizen dargestellt werden. Die in **Scilab** integrierte Grafik ermöglicht es, Ergebnisse darzustellen und in unterschiedliche Formate zu exportieren. Auf der anderen Seite stellt **Scilab** immens viele Graphikbefehle zur Verfügung.

Scilab kann man am Anfang einfach wie einen Taschenrechner benutzen, der Operationen mit Vektoren — reell oder komplex — beherrscht, und der Kurven und Flächen graphisch darstellen kann.<sup>24</sup>,

### 4.3 Programmaufbau und Implementierung

Zur besseren Übersicht und strukturellen Veranschaulichung habe ich bestimmte Programmabläufe, die in jeder Iteration wiederholt aufgerufen werden, als Funktionen geschrieben. Wie zuvor erwähnt, habe ich mich für das Verfahren des Iterated Local Search entschieden. Um ein qualitativ aussagekräftiges Datenspektrum zu erhalten, habe ich eine Datenreihe mit unterschiedlich zufälligen Startlösungen im Ausmaß von  $n=100$  (Iterationen) generiert. Zu Beginn des Programms werden die Daten der Sichtkontakte (SK\_matrix) und die der Distanzen (dm) in Matrizenschreibweise implementiert.

In der ersten und zweiten Funktion werden aufgrund allgemein gültiger Rechenvorschriften die oben genannten Matrizen in Vektorschreibweise umgewandelt:

```
function[ZV_initial]=Matrix2Vektor(MATRIX_initial);

    m_trans = MATRIX_initial';
    sv_bin = m_trans(:);
    ZV_initial = sv_bin';
endfunction;

function[SKV]=SK_Matrix2Vektor(SK_matrix);
    SK_trans = SK_matrix';
    SKV = SK_trans(:);
endfunction;
```

Abbildung 4.1: Programmteil 1

---

<sup>24</sup> Quelle Internet: wikipedia; www.scilab.org

Die dritte und letzte Funktion kann man einem Travelling Salesman Problem gleichsetzen und widerspiegelt die Lösungsstruktur meines Tourenproblems:

```
function[Gesamtkilometer_initial,Streckennummer]=TSP(MATRIX_initial)
for z=1:6
    hm=MATRIX_initial(:,z); //Hilfsmatrix
    w=find(hm==1); //finde bitte alle Positionen mit einer 1!
    A=w(1); //Erste Position ordne A zu
    B=w(2); //Zweite Position ordne B zu usw.
    C=w(3);
    D=w(4);
    L01=dm(A,18)+dm(A,B)+dm(B,C)+dm(C,D)+dm(D,18);
    L02=dm(A,18)+dm(A,B)+dm(B,D)+dm(D,C)+dm(C,18);
    L03=dm(A,18)+dm(A,C)+dm(C,B)+dm(B,D)+dm(D,18);
    L04=dm(A,18)+dm(A,C)+dm(C,D)+dm(D,B)+dm(B,18);
    L05=dm(A,18)+dm(A,D)+dm(D,B)+dm(B,C)+dm(C,18);
    L06=dm(A,18)+dm(A,D)+dm(D,C)+dm(C,B)+dm(B,18);
    L07=dm(B,18)+dm(B,A)+dm(A,C)+dm(C,D)+dm(D,18);
    L08=dm(B,18)+dm(B,A)+dm(A,D)+dm(D,C)+dm(C,18);
    L09=dm(B,18)+dm(B,C)+dm(C,A)+dm(A,D)+dm(D,18);
    L10=dm(B,18)+dm(B,C)+dm(C,D)+dm(D,A)+dm(A,18);
    L11=dm(B,18)+dm(B,D)+dm(D,A)+dm(A,C)+dm(C,18);
    L12=dm(B,18)+dm(B,D)+dm(D,C)+dm(C,A)+dm(A,18);
    Alle_Strecken_gelistet=[L01;L02;L03;L04;.....;L17]
    [Minimum_Strecke Minimum_position]=min(Alle_Strecken_gelistet)
    var1(z)=Minimum_Strecke; //Hilfsvariable
    var2(z)=Minimum_position;
end
Gesamtkilometer_initial=sum(var1) //Gesamtkilometer der Tour
Streckennummer=[var2];
endfunction;
```

Abbildung 4.2: Programmteil 2

Zunächst wird in der Initialmatrix bzw. in der durch den Local Search veränderten Matrix jede Spalte separat (die Spalten entsprechen den Wochentagen wobei Spalte Eins den Montag und Spalte Sechs den Freitag repräsentiert) betrachtet. In jeder dieser Spalten befinden sich an unterschiedlichen Positionen insgesamt vier Einser. Bislang könnte man nur ablesen, welche Standorte angefahren werden sollen, nicht aber in welcher Reihenfolge. Diese ist jedoch für die Kostenberechnung entscheidend. Es werden also die Positionen der Einser eruiert und diesen anschließend die Buchstaben A,B,C und D zugeordnet.

Wie zuvor beschrieben, gibt es bei vier besuchten Standorten zwölf Möglichkeiten in welcher Reihenfolge sie angefahren werden können. Jede dieser Alternativen wird berechnet und die Strecke mit den geringsten Kilometern (*=Minimum\_Strecke*) wird ausgewählt. Dieser Ablauf wird auf jeden Wochentag übertragen und schließlich die Gesamtkilometer der ganzen Woche berechnet. Laut Problemformulierung lautet die zweite Zielsetzung, die Projektkosten zu minimieren. Diese ist mit der Minimierung der Gesamtkilometer äquivalent, da die Projektkosten pro Kilometer abgerechnet werden. In jeder Iteration des Local Search werden die Gesamtkilometer neu errechnet und so in den Kosten mit berücksichtigt. Nun kommen wir zum Hauptprogramm und zur Befriedigung der ersten Zielsetzung, nämlich der Maximierung der Sichtkontakte.

### 4.3.1 Programmierschritt I

Zu Beginn wird zunächst eine 17x6 Nullmatrix erstellt, die zu einer zulässigen Anfangs- bzw. Initialmatrix (=Matrix\_initial) umfunktioniert wird. Es werden pro Wochentag (Spalte) vier Einser zufällig vergeben. Hierbei ist darauf zu achten, dass es erstens genau vier Einser pro Spalte sein müssen und zweitens darf kein Standort mehr als dreimal angefahren werden. Die erste Bedingung ist in der While-Schleife ( $k < 5$ ) wieder zu erkennen. Um die zweite Restriktion zu berücksichtigen, wurde zunächst eine unzulässige Lösung geschaffen, deren Eigenschaft mit der While-Schleife ( $r_{01} > 3, \dots$ ) überprüft wird.

Da die erste Lösung unzulässig ist, wird die Schleife ausgeführt und bricht erst wieder ab, wenn die Summe der Einträge jeder Zeile  $\leq 3$  ist.

```

r01=4;r02=4;r03=4;r04=4;r05=4;r06=4;r07=4;r08=4;r09=4;r10=4;.....;r17=4;
while r01 > 3 | r02 > 3 | r03 > 3 | r04 > 3 | r05 > 3 |....| r17 > 3 then
    MATRIX_zeros = zeros(17,6);
    for j=1:6;
        k=1;
        while k<5 then
            x = rand();
            y = 16*x+1;
            z = round(y);
            if MATRIX_zeros(z,j)==0 then
                MATRIX_zeros(z,j)=1;
                k=k+1;
                MATRIX_initial = MATRIX_zeros;
            end;
        end;
    end;
    r_1=MATRIX_initial(1,:); r01=sum(r_1);
    r_2=MATRIX_initial(2,:); r02=sum(r_2);
    r_3=MATRIX_initial(3,:); r03=sum(r_3);
    r_4=MATRIX_initial(4,:); r04=sum(r_4);
    r_5=MATRIX_initial(5,:); r05=sum(r_5);
    r_6=MATRIX_initial(6,:); r06=sum(r_6);
    r_7=MATRIX_initial(7,:); r07=sum(r_7);
    r_8=MATRIX_initial(8,:); r08=sum(r_8);
    r_9=MATRIX_initial(9,:); r09=sum(r_9);
    r_10=MATRIX_initial(10,:); r10=sum(r_10);
    r_11=MATRIX_initial(11,:); r11=sum(r_11);
    r_12=MATRIX_initial(12,:); r12=sum(r_12);
    r_13=MATRIX_initial(13,:); r13=sum(r_13);
    r_14=MATRIX_initial(14,:); r14=sum(r_14);
    r_15=MATRIX_initial(15,:); r15=sum(r_15);
    r_16=MATRIX_initial(16,:); r16=sum(r_16);
    r_17=MATRIX_initial(17,:); r17=sum(r_17);
end;

```

Abbildung 4.3: Programmteil 3

### 4.3.2 Programmierschritt II

Dieser Programmabschnitt dient der Kostenermittlung und in weiterer Folge der Errechnung des gewichteten Kriteriums, welches als Vergleichskennzahl zwischen den Iterationen des Local Search herangezogen wird. Für die Startlösung (Initialmatrix) wurde für das Kriterium als *Gewinn\_initial* gewählt. Die Gesamtkosten für das Projekt entsprechen jenen Kosten, die durch den Material- und Personeneinsatz einerseits, und den Planungs- und Organisationsaufwand andererseits entstehen. Die Agentur hat die Kosten auf pro gefahrenen Kilometer umgewälzt und kalkulierte dafür einen Betrag in der Höhe von 3 Euro. Der absolute Benefit hingegen wird in Geldeinheiten pro Sichtkontakt definiert und beträgt laut

Marketingabteilung des Konzerns 0,01€/Sichtkontakt. Aus der Subtraktion der beiden Beträge lässt sich der Gewinn für das Unternehmen ermitteln. Der Gewinn ist resultiert aus der zufälligen Startlösung und der für diese Lösung entsprechenden optimalen Reihenfolge der Tour. Dieser Wert entspricht dem ersten gewichteten Kriterium.

```

disp('')
MATRIX_initial;
Startmatrix=MATRIX_initial;
ZV_initial = Matrix2Vektor(MATRIX_initial);
SKV = SK_Matrix2Vektor(SK_matrix);
Initialsichtkontakte = ZV_initial * SKV;
Liste_SK_initial(1)=Initialsichtkontakte;
disp('')
Gesamtkilometer_initial=TSP(MATRIX_initial);
disp('Ein Sichtkontakt bringt wirtschaftlich umgerechnet in Geldwerten 1CENT.')
Benefit_initial = Initialsichtkontakte / 100;
disp(Benefit_initial,'Der daraus resultierende Gesamtbefiz (Umsatz) ergibt in EURO:')
disp('Nun werden die akkumulierten Kosten abgezogen')
Gesamtkosten = Gesamtkilometer_initial*3;
disp(Gesamtkosten,'Die Gesamtkosten werden mit 3€ pro gefahrenen Km berechnet und ergeben:')
Gewinn_initial = Benefiz_initial - Gesamtkosten;
disp(Gewinn_initial,'Der Gewinn nach Abzug der Kosten beträgt:')
Gewinn_ls = 1 + Gewinn_initial;
disp('Der Gewinn entspricht dem Nutzen des Projektes. Dieser Nutzen wird für das Kriterium
(gewichtet) verwendet.')
disp('Die lokale Suche wird erst dann abgebrochen, wenn kein besseres Kriterium mehr gefunden
werden kann.')
disp('Der Algorithmus hat dann sein lokales Optimum erreicht!')

```

Abbildung 4.4: Programmteil 4

### 4.3.3 Programmierschritt III

In diesem Abschnitt konzentriere ich mich auf den bedeutendsten Baustein des Optimierungsproblems – den Local Search.

Zu Beginn wird mit einer While-Schleife abgefragt, ob das neu errechnete Kriterium größer ist als das vorhergehende. Im erstmaligen Vergleich wird die Schleife stets ausgeführt, da nur das Kriterium der Startlösung in Betracht gezogen werden kann. Im nächsten Schritt wird die Initialmatrix sukzessive verändert. Dabei scheidet systematisch ein „alter“ Standort aus und ein „neuer“ Standort wird statt dessen aufgenommen. Nachdem die Entscheidungsvariablen von binärer Ausprägung sind, kann dieser Vorgang sehr einfach, durch das Setzen von einem Element mit der Ausprägung Eins auf die Ausprägung Null und umgekehrt, umgesetzt werden. Es werden also durch minimale Veränderungen der Nachbarschaft iterativ neue bessere Lösungen gesucht. Nachdem die Nachbarschaft vollständig untersucht worden ist, verlässt das Programm die beiden While-Schleifen und wiederholt den Vorgang für den nächsten Tag. Dies geschieht mit der For-Schleife, die von Eins bis Sechs läuft. Die neue durch den Local Search veränderte Matrix wird als *Matrix\_ls* bzw. *Matrix\_max* bezeichnet.

Weiters fehlt noch ein letzter wichtiger Programmierschritt, welcher zur Einhaltung der Restriktionen notwendig ist. Wurde ein „besseres“ Kriterium in der Nachbarschaft gefunden, ist dieses nur dann zulässig, wenn kein Standort mehr als drei Mal angefahren wurde. Um dieser Forderung nachzukommen, werden die Einsen pro Zeile summiert und mit Hilfe einer If-Anweisung abgefragt, ob die Restriktion eingehalten wurde. Abhängig von den neu vergebenen Standorten, werden die

Sichtkontakte pro Standort für alle Wochentage summiert und als *Sichtkontakte\_neu* gespeichert.

Abschließend wird aufgrund der unterschiedlichen Distanzen der Standorte zueinander parallel dazu das Ergebnis der Tourenplanung neu berechnet und aktualisiert. Ist das neu ermittelte Kriterium größer als das vorhergehende, wird die Initialmatrix durch die veränderte Matrix ersetzt und der ganze Ablauf solange wiederholt (Iterative Lokale Suche) bis ein lokales Optimum erreicht wurde.

```

Iteration=0;
is=Initialsichtkontakte;
m=1;
while Gewinn_initial < Gewinn_ls do
    Gewinn_initial = Gewinn_ls ;
    for b=1:6;
        mx=MATRIX_initial;
        d=1;
        while d<5 then
            for c=1:17;
                if mx(c,b)==1 then
                    mx(c,b)=0;
                    d=d+1;
                    mx;
                    e=1;
                    while e<14 then
                        for f=1:17;
                            MATRIX_ls=mx;
                            if MATRIX_ls(f,b)==0 then
                                MATRIX_ls(f,b)=1;
                                e=e+1;
                                MATRIX_ls;
                                ZV_ls = Matrix2Vektor(MATRIX_ls);
                                Sichtkontakte_neu = ZV_ls * SKV;
                                r_1=MATRIX_ls(1,:); r01=sum(r_1);
                                r_2=MATRIX_ls(2,:); r02=sum(r_2);
                                r_3=MATRIX_ls(3,:); r03=sum(r_3);
                                r_4=MATRIX_ls(4,:); r04=sum(r_4);
                                r_5=MATRIX_ls(5,:); r05=sum(r_5);
                                r_6=MATRIX_ls(6,:); r06=sum(r_6);
                                r_7=MATRIX_ls(7,:); r07=sum(r_7);
                                r_8=MATRIX_ls(8,:); r08=sum(r_8);
                                r_9=MATRIX_ls(9,:); r09=sum(r_9);
                                r_10=MATRIX_ls(10,:); r10=sum(r_10);
                                r_11=MATRIX_ls(11,:); r11=sum(r_11);
                                r_12=MATRIX_ls(12,:); r12=sum(r_12);
                                r_13=MATRIX_ls(13,:); r13=sum(r_13);
                                r_14=MATRIX_ls(14,:); r14=sum(r_14);
                                r_15=MATRIX_ls(15,:); r15=sum(r_15);
                                r_16=MATRIX_ls(16,:); r16=sum(r_16);
                                r_17=MATRIX_ls(17,:); r17=sum(r_17);
                                if Sichtkontakte_neu > Initialsichtkontakte & r01 < 4 & r17 < 4 then
                                    Initialsichtkontakte = Sichtkontakte_neu;
                                    MATRIX_max=MATRIX_ls;
                                else end;
                            else end;
                        end;
                    end;
                else end;
            end;
        end;
        if Initialsichtkontakte>is then
            mn=MATRIX_max;
        end
        is=Initialsichtkontakte;
    end;
    Sichtkontakte_mit_LS=Initialsichtkontakte;
    Hilfsliste(m,1)=Initialsichtkontakte;
    Benefit_ls = Sichtkontakte_mit_LS / 100;
    //disp('')
    Gesamtkilometer=TSP(MATRIX_max);
    Gesamtkosten_ls = Gesamtkilometer*3;
    //disp('')
    Gewinn_ls = Benefit_ls - Gesamtkosten;
    Iteration=Iteration+1;
end;

```

```

        m=m+1;
        //Liste_ls(:,:)=Sichtkontakte_mit_LS; //Liste der mittels LS veränderten Nachbarn.
        //plot(Iteration,Sichtkontakte_mit_LS,type="s")
        //xgrid(7)
        //xtitle("LOCAL SEARCH OPTIMIZATION","Iteration","Sichtkontakte")
        MATRIX_initial=mn;
end;

```

Abbildung 4.5: Programmteil 5

#### 4.3.4 Programmierschritt IV

Hier werden im Wesentlichen Listen und Ausgabeneigenschaften definiert und programmiert. Die einzige vorhandene IF-Schleife bewirkt, daß die ermittelten optimalen Ergebnisse verschiedener Startlösungen miteinander verglichen werden und die wirtschaftlich effizienteste Standortzuordnung ausgesucht wird. Der Umfang der gesamten Datenreihe wurde mit  $n=100$  beziffert. Der größtmöglich zu erzielende Wert bei den Sichtkontakten muss nicht unbedingt bedeuten, daß er auch der wirtschaftlichste ist. Da es sich um ein kombiniertes Standort-Tourenproblem handelt, fließen die Gesamtkilometer der Tour und als Folge auch die Projektkosten wesentlich mit ein.

```

MATRIX_optimal=MATRIX_ls;
Lokales_Optimum_SK=Sichtkontakte_mit_LS;
//disp(Iteration,'Anzahl der Iterationen:')
Liste(1)=Lokales_Optimum_SK;
Iterationen(1)=Iteration;
Liste_km(1)=TSP(MATRIX_max);
[Gesamtkilometer,Streckennummer]=TSP(MATRIX_max);
Liste_Streckennummer(1,:)=Streckennummer';
///end;
//disp('')
//Kernliste=[Liste_SK_initial Liste Iterationen Liste_km]
//disp('')
//
//Liste_Streckennummer
//disp('')
//Hilfsliste
if Lokales_Optimum_SK>=Local_Optima then
    if Gesamtkilometer<=Distanz then
        Local_Optima=Lokales_Optimum_SK;
        Distanz=Gesamtkilometer;
        Order=Streckennummer;
        lm=MATRIX_optimal;
    end;
end;
end;
Local_Optima
disp('')
Distanz
disp('')
Reihenfolge=Order'
disp('')
Loesungsmatrix=lm
disp('')
Liste_Streckennummer
Kernliste=[Liste_SK_initial Liste Iterationen Liste_km]

```

Abbildung 4.6: Programmteil 6

## 4.4 Zusammenfassung

Grundsätzlich bietet das Softwarepaket von Scilab eine gute und solide Basis im Bereich der Optimierungsprobleme. Der Markt bietet selbstverständlich noch weitaus umfangreichere Software zu diesem Thema an. Der Vorteil von Scilab ist vordergründig die einfache Programmiersprache. Für die in dieser Arbeit behandelte Aufgabenstellung, konnten sehr zufriedenstellende Ergebnisse erzielt werden. Bei der Beschreibung meines Programmablaufes bin ich bewußt nicht näher auf die einzelnen Befehle eingegangen, da einerseits dafür eine sehr repräsentative HELP-Funktion im Programm selbst, und andererseits auch umfangreich ergänzende Literatur vorhanden ist.

## 5 EXPERIMENTELLER TEIL

### 5.1 Interpretation der Ausgabedaten

Folgende Lösungsparameter werden von dem Programm ermittelt. Allen voran wird der größte Wert der Datenreihe im Ausgabefenster ausgegeben. Dieser wird als **Local\_Optima** bezeichnet und entspricht dem lokalen Optimum eines Iterated Local Search. Eine Datenreihe mit n=100 zufälligen Startlösungen liefert genau hundert Endlösungen.

**Local\_Optima** → **Maximal zu erzielende Sichtkontakte bei 100 Iterationen!**

Die nächsten beiden Parameter die ausgegeben werden, beziehen sich auf das Tourenproblem, wobei die Angabe der Gesamtkilometer der Tour nur sekundär von Bedeutung ist. Die Ausgabe **Reihenfolge** definiert, welche Tourenreihenfolge an welchem Wochentag durchgeführt werden soll. Eine Reduzierung der zurückgelegten Distanzen führt automatisch zu einer Kostenminimierung, die von dem Unternehmen als zweite Zielsetzung vorgegeben wurde.

**Reihenfolge** → **Kürzeste Wegstrecken mit dem Ziel der Kostenminimierung**

Die **Lösungsmatrix** gibt Auskunft darüber, welcher Standort an welchem Wochentag angefahren werden sollte. Es werden nur jene Standorte angefahren, deren Matrixeinträge den Wert Eins angenommen haben. Alle anderen Standorte scheidet aus. Hier kann eine Prüfung vorgenommen werden ob die Restriktionen auch eingehalten worden sind (maximal drei Mal darf ein Standort besucht werden und pro Tag sollen genau vier angefahren werden).

**Lösungsmatrix** =  $\begin{cases} 1, & \text{Standort wird angefahren} \\ 0, & \text{Standort scheidet aus} \end{cases}$

Die nachstehenden ausgegebenen Listen (Liste\_Streckenummer, Kernliste und Hilfsliste) dienen zum einen der Informationsgewinnung und zum anderen zur Weiterberechnung verschiedener Kennziffern. Die Kernliste beispielsweise gibt Aufschluß darüber, wie oft wiederholt worden ist bis das Lokale Optimum erreicht wurde.

## 5.2 Auswertung und Analyse der Daten - Standortplanung

Der Iterated Local Search liefert den maximal zu erzielenden Wert von **432.250** Sichtkontakten. Dies entspricht der ersten Zielforderung des Unternehmens, die mindestens 400.000 Sichtkontakte forderte. Die Gesamtkilometer der zurückgelegten Wegstrecke (Tour) betragen **613 km**. Der Umstand der vollständigen Enumeration aller möglichen Permutationen der Fahrstrecke bewirkt, dass die Kosten in Abhängigkeit der Sichtkontakte minimiert worden sind. Wir sehen also ein eindeutiges Erreichen beider Zielvorgaben unter der Einhaltung der geforderten Restriktionen.

Die so ermittelte Lösungsmatrix hat folgende Struktur:

Standorte	Wochentage					
1	0	1	1	0	0	
0	0	0	0	0	0	
0	0	0	0	0	0	
0	0	0	0	0	0	
0	0	0	0	0	0	
1	0	1	0	0	1	
0	0	0	0	0	0	
0	1	1	1	0	0	
1	0	0	0	0	1	
0	0	1	0	1	0	
1	1	0	0	1	0	
0	0	0	0	0	1	
0	0	0	1	1	0	
0	0	0	0	0	0	
0	0	0	0	0	1	
0	1	0	1	1	0	
0	1	0	0	0	0	

Abbildung 5.1: Endlösungsmatrix

Aus dieser Matrix ist nun herauszulesen, welcher Standort an welchem Wochentag angefahren werden sollte, um die maximalen Sichtkontakte erzielen zu können. Die nachstehende Tabelle fasst diese Ergebnisse zusammen:

Wochentag	Standort 1	Standort 2	Standort 3	Standort 4
<b>Montag</b>	1	6	9	11
<b>Dienstag</b>	8	11	16	17
<b>Mittwoch</b>	1	6	8	10
<b>Donnerstag</b>	1	8	13	16
<b>Freitag</b>	10	11	13	16
<b>Samstag</b>	6	9	12	15

Tabelle 5.1: Standortwahl ohne Reihenfolge

Mit Hilfe dieser Tabelle lässt sich jedoch nicht erkennen, in welcher Reihenfolge die Standorte anzufahren sind. Dies spielt bei der Kostenminimierung jedoch eine

entscheidende Rolle. Welche Informationen dafür notwendig und wie diese zu interpretieren sind, wird im nachfolgenden Punkt erklärt.

### 5.3 Auswertung und Analyse der Daten - Tourenplanung

Unter Bezugnahme der obigen Tabelle werden den Standorten Buchstaben in alphabetischer Reihenfolge zugeordnet:

Standort 1→A; Standort 2→B; Standort 3→C; Standort 4→D

Wie schon bei der Ergebnisinterpretation dargelegt, liefert uns das Programm einen Parameter namens Reihenfolge, der lautet:

6.	4.	11.	6.	8.	1.
----	----	-----	----	----	----

Mittels der nachstehenden Tabelle kann nun abgelesen werden, welche der möglichen Routen das Programm evaluiert hat (orange markiert):

Strecken Nr.	Anfangspunkt	Standort 1	Standort 2	Standort 3	Standort 4	Endpunkt
1	Lager	A	B	C	D	Lager
2	Lager	A	B	D	C	Lager
3	Lager	A	C	B	D	Lager
4	Lager	A	C	D	B	Lager
5	Lager	A	D	B	C	Lager
6	Lager	A	D	C	B	Lager
7	Lager	B	A	C	D	Lager
8	Lager	B	A	D	C	Lager
9	Lager	B	C	A	D	Lager
10	Lager	B	C	D	A	Lager
11	Lager	B	D	A	C	Lager
12	Lager	B	D	C	A	Lager

Tabelle 5.2: Streckeneigenschaften

Verknüpft man Tabelle 5.1 mit 5.2 so erhält man automatisch den Tourenplan mit entsprechend effizienter Reihenfolge:

Wochentag	Standort 1 (A)	Standort 2 (B)	Standort 3 (C)	Standort 4 (D)
Montag	1	11	9	6
Dienstag	8	16	11	17
Mittwoch	6	10	1	8
Donnerstag	1	16	13	8
Freitag	11	10	16	13
Samstag	6	9	12	15

Tabelle 5.3: Tourenplan mit Reihenfolge

## 5.4 Graphischer Tourenplan

Unter zur Hilfenahme der folgenden Tabelle der Standortadressen

Legende	Standort
1	1010 Wien, Schwedenplatz
2	1020 Wien, Stadioncenter
3	1020 Wien, Praterstern
4	1030 Wien, Landstrasse / Rochusmarkt
5	1040 Wien, Karlsplatz
6	1070 Wien, Naschmarkt
7	1090 Wien, Hauptuniversität
8	1100 Wien, Reumannplatz
9	1110 Wien, Gasometer
10	1150 Wien, Mariahilferstrasse - Wbf
11	1160 Wien, Lugnercity
12	1190 Wien, Q19 Shopping Center
13	1200 Wien, Milleniumstower
14	1210 Wien, Shopping City Nord
15	1210 Wien, Floridsdorf U6 Endstation
16	1220 Wien, Donauzentrum
17	1220 Wien, Gewerbepark Stadlau

Tabelle 5.4: Standortadressen

und der zuvor erstellten Tabelle 5.3 kann der grafische Tourenplan problemlos erstellt werden. In dem unten angeführten Tourenplan ist beispielsweise der Montag dargestellt worden. Die eingezeichneten Kreise signalisieren die Reihenfolge der anzufahrenden Standorte. Start- und Endpunkt bildet jeweils das Lager L:

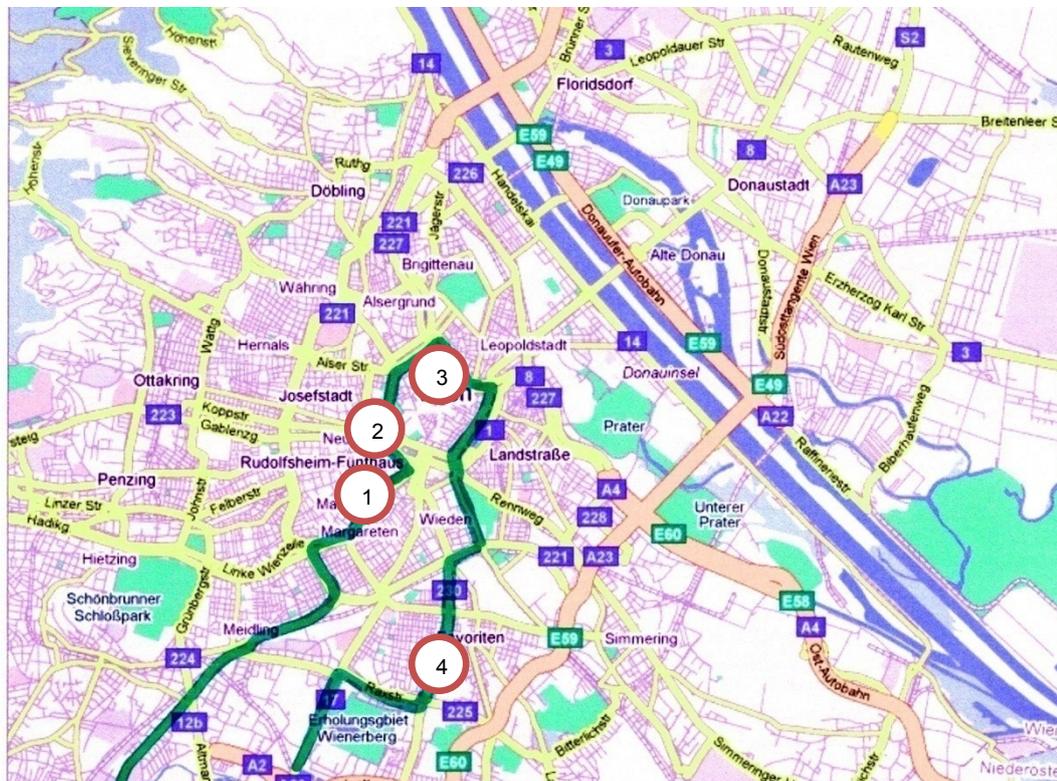


Abbildung 5.2: Grafischer Tourenplan für z.B. den Wochentag Montag<sup>25</sup>

<sup>25</sup> Quelle: Stadtplan – Maps.Google.at/maps

## 5.5 Weitere experimentelle Auswertungen

### 5.5.1 Computerlaufzeit

Umfang der Datenreihen (DR)	Rechenlaufzeit	Durchschnittliche Iteration	Durchschnittlicher Zielfunktionswert	Maxima der lokalen Optima
1	~4sek.	8,00	-	425.250
10	~45sek.	8,30	412.550,00	428.750
50	~ 6min	8,76	409.745,00	434.000
100	~17 min.	8,76	409.412,50	432.250

Tabelle 5.5: Rechenzeiten

Die entsprechenden Lösungsmatrizen lauten:

Lösungsmatrix für 50 DR	Lösungsmatrix für 10 DR	Lösungsmatrix für 1 DR
1. 1. 0. 1. 0. 0.	1. 1. 1. 0. 0. 0.	1. 0. 1. 1. 0. 0.
0. 0. 0. 0. 0. 0.	0. 0. 0. 0. 0. 0.	0. 0. 0. 0. 0. 0.
0. 0. 0. 0. 0. 0.	1. 0. 0. 0. 0. 0.	0. 0. 0. 0. 0. 0.
0. 0. 0. 0. 0. 0.	1. 0. 0. 0. 0. 0.	0. 0. 0. 0. 0. 0.
0. 0. 0. 0. 0. 0.	0. 0. 0. 0. 0. 0.	0. 0. 0. 0. 0. 0.
0. 0. 0. 0. 0. 1.	0. 0. 1. 0. 0. 1.	0. 0. 0. 0. 0. 1.
0. 0. 1. 0. 0. 1.	0. 1. 1. 0. 0. 0.	0. 1. 1. 0. 0. 0.
0. 1. 1. 1. 0. 0.	1. 0. 1. 1. 0. 0.	0. 1. 1. 0. 1. 0.
0. 1. 1. 0. 0. 1.	0. 1. 0. 0. 0. 0.	0. 0. 0. 0. 0. 1.
0. 1. 0. 0. 1. 0.	0. 0. 0. 1. 1. 1.	1. 0. 0. 0. 1. 0.
0. 0. 0. 0. 1. 0.	0. 0. 0. 0. 0. 0.	0. 1. 0. 0. 0. 1.
1. 0. 0. 1. 0. 0.	0. 0. 0. 0. 0. 0.	1. 0. 0. 1. 0. 0.
1. 0. 0. 0. 0. 0.	0. 1. 0. 0. 1. 0.	0. 0. 1. 0. 1. 0.
0. 0. 0. 0. 1. 0.	0. 0. 0. 0. 0. 0.	0. 1. 0. 0. 0. 0.
0. 0. 1. 0. 0. 0.	0. 0. 0. 1. 1. 0.	0. 0. 0. 1. 0. 0.
1. 0. 0. 1. 1. 0.	0. 0. 0. 1. 1. 1.	1. 0. 0. 1. 1. 0.
0. 0. 0. 0. 0. 1.	0. 0. 0. 0. 0. 1.	0. 0. 0. 0. 0. 1.

Anhand dieser aufgeschlüsselten Daten lässt sich ein linearer Zusammenhang zwischen der Laufzeit und dem Datenreihenumfang erkennen. Dies ist darauf zurückzuführen, dass ein heuristisches Lösungsverfahren mit polynominalen Rechenaufwand verwendet wurde. Außerdem kann man in der rechten Spalte die durchschnittliche Anzahl der ausgeführten Iterationen des Local Search während des Berechnungsprozesses ablesen. Hier kann festgestellt werden, dass die lokalen Optima im Mittel nach 9 Iterationen erreicht wurden.

### 5.5.2 Lösungsqualität

Das folgende Plott (Ausdruck) zeigt wie sich die Lösungsqualität während eines Laufs der Lokalen Suche entwickelt. Diese Grafik wurde mit dem SCILAB Grafikeditor entworfen:

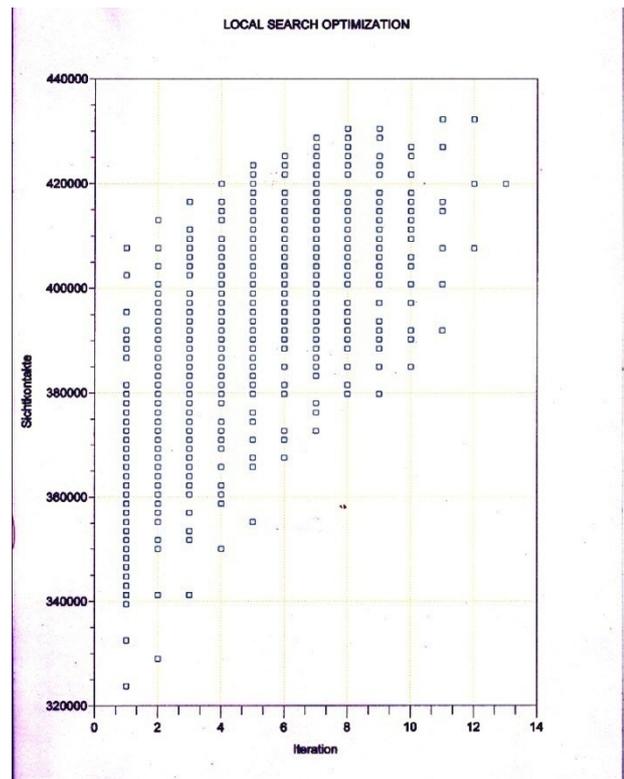


Diagramm 5.1: Local Search Optimization

Um es übersichtlicher zu gestalten, habe ich die ersten sechszwanzig Datenreihen in einem Excel Diagramm dargestellt.

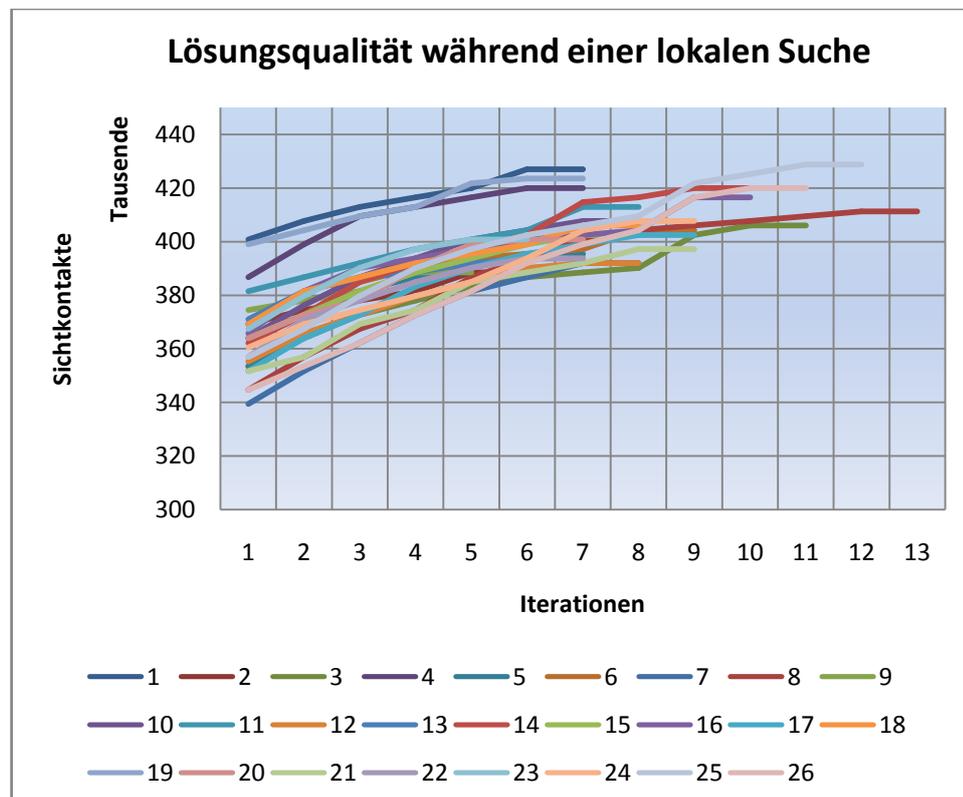


Diagramm 5.2: Verlauf der Lösungswerte

Eine weitere Tabelle zeigt die gesammelten lokalen Optima bei einer Datenreihen von  $n=100$ . Mittels dieser Werte können wichtige Maßzahlen errechnet werden.

	Datenreihe				
	01-20	21-40	41-60	61-80	81-100
L	392000	414750	406000	414750	414750
o	418250	416500	413000	414750	393750
k	400750	385000	390250	421750	402500
a	427000	392000	418250	409500	402500
l	409500	407750	406000	425250	397250
e	421750	421750	406000	418250	<b>432250</b>
	411250	392000	406000	407750	421750
O	393750	413000	385000	406000	379750
p	411250	409500	416500	423500	425250
i	418250	423500	388500	397250	414750
m	416500	413000	413000	400750	404250
a	428750	427000	416500	418250	404250
	420000	402500	409500	416500	400750
	414750	411250	400750	425250	400750
	399000	409500	388500	397250	421750
	425250	400750	409500	407750	397250
	409500	409500	400750	400750	414750
	399000	406000	414750	416500	416500
	421750	418250	400750	409500	390250
	425250	430500	409500	406000	402500

Tabelle 5.6: Lokale Optima aller Datenreihen

Daraus ergibt sich ein arithmetisches Mittel von **409.412,5** Sichtkontakten. Mit dem Maxima von **432.250** Sichtkontakten liegt das gewählte Optimum eindeutig über den Erwartungswert. Die **Standardabweichung ( $\sigma$ )** wurde mit 11244,69 berechnet. Über das nachstehende Histogramm wird die Häufigkeitsverteilung dargestellt.

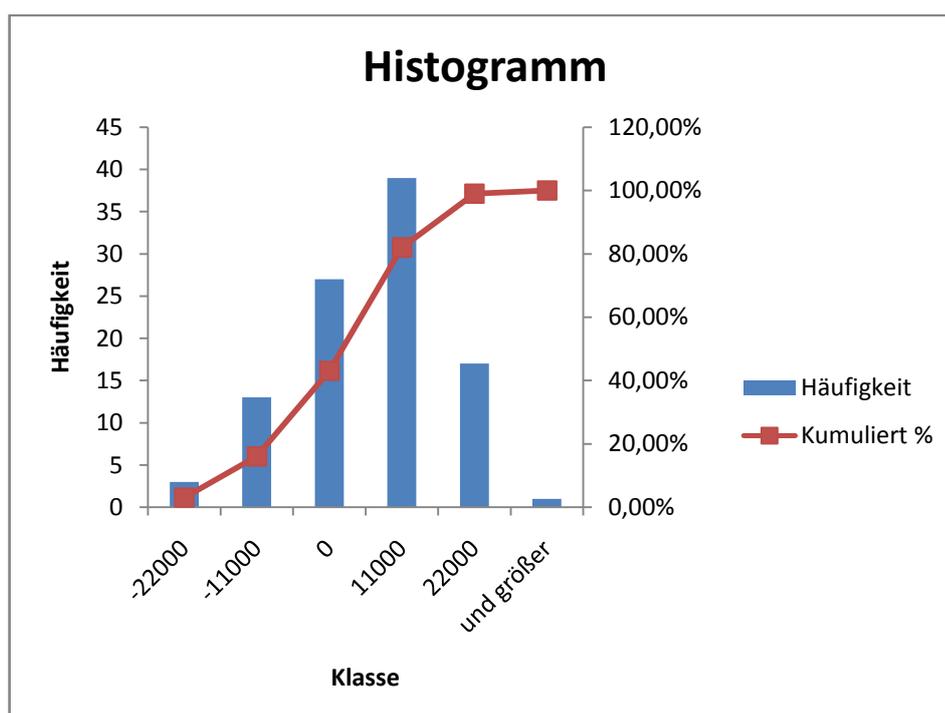


Diagramm 5.3: Verteilfunktion

Die Meßwerte wurden in den folgenden Klassen eingeteilt:

<b>Klasse</b>	<b>von</b>	<b>bis</b>
<b>I</b>	-22000	-11000
<b>II</b>	-11000	0
<b>III</b>	0	+11000
<b>IV</b>	+11000	+22000

Tabelle 5.7: Klasseneinteilung

Aus dem Diagramm hat man erkennen, dass es sich um eine Normalverteilung handelt. Diese Verteilung war aufgrund der Modellierung und des gewählten Lösungsansatzes zu erwarten.

### 5.5.3 Sensibilitätsanalyse 1

Bei der Sensibilitätsanalyse werden primär die Auswirkungen von Veränderungen verschiedener Parameterwerte im Modell auf die optimale Lösung untersucht.<sup>26</sup>

In dem vorliegenden Optimierungsproblem wählte das Unternehmen eine Probleminstanz mit dem Wert von 0,01 Geldeinheiten pro Sichtkontakt. Welche Auswirkungen auf die Ergebnisse und auf die Rechenzeiten haben eine Veränderung dieser Probleminstanz zur Folge? In dieser Arbeit wurden vier weitere Probleminstanzen untersucht und die daraus resultierenden neuen Werte in den nachfolgenden Tabellen bzw. Diagrammen dargestellt:

Die unterschiedlichen Probleminstanzen in Werten:

Instanz	Erklärung
0,05	Ein Sichtkontakt bringt dem Unternehmen 0,05 GE pro Sichtkontakt
0,02	Ein Sichtkontakt bringt dem Unternehmen 0,02 GE Pro Sichtkontakt
<b>0,01</b>	<b>Basis (Analyse vorangehend!)</b>
0,005	Ein Sichtkontakt bringt dem Unternehmen 0,005 GE pro Sichtkontakt
0,001	Ein Sichtkontakt bringt dem Unternehmen 0,001 GE pro Sichtkontakt.

Tabelle 5.8: Probleminstanzen

#### 5.3.3.1 Probleminstanz (0,05GE)

Lösungsmatrizen:

Lösungsmatrix für 100 DR						
1.	1.	1.	0.	0.	0.	
0.	0.	0.	0.	0.	0.	
0.	0.	0.	0.	0.	0.	
0.	1.	1.	0.	0.	0.	
0.	1.	0.	0.	0.	0.	
1.	0.	0.	0.	0.	1.	
0.	0.	0.	0.	0.	0.	
1.	0.	0.	1.	0.	1.	
0.	0.	0.	0.	0.	0.	
0.	0.	0.	1.	1.	0.	
0.	0.	0.	0.	1.	0.	
0.	0.	1.	0.	1.	0.	
0.	0.	1.	0.	0.	1.	
1.	1.	0.	0.	0.	0.	
0.	0.	0.	1.	1.	0.	
0.	0.	0.	0.	0.	1.	

Lösungsmatrix für 50 DR						
1.	1.	1.	0.	0.	0.	
0.	0.	0.	0.	0.	0.	
0.	0.	0.	0.	0.	0.	
0.	0.	0.	0.	0.	0.	
1.	0.	0.	1.	0.	0.	
0.	0.	0.	1.	0.	1.	
0.	0.	0.	0.	0.	0.	
1.	0.	1.	0.	1.	0.	
0.	0.	0.	0.	0.	0.	
0.	1.	0.	0.	1.	1.	
1.	1.	0.	1.	0.	0.	
0.	0.	0.	0.	0.	0.	
0.	0.	1.	0.	0.	0.	
0.	0.	1.	0.	1.	0.	
0.	0.	0.	1.	0.	1.	
0.	1.	0.	0.	1.	1.	
0.	0.	0.	0.	0.	0.	

<sup>26</sup> Zitat: Hillier, Lieberman [1988], Operations research, Oldenbourg Verlag, 4. Auflage

Lösungsmatrix für 10 DR

1.	1.	0.	0.	1.	0.
0.	1.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	1.
0.	0.	1.	0.	0.	0.
0.	1.	1.	1.	0.	0.
0.	1.	0.	0.	0.	0.
1.	0.	0.	1.	0.	1.
1.	0.	1.	1.	0.	0.
1.	0.	1.	0.	0.	0.
0.	0.	0.	0.	1.	1.
0.	0.	0.	0.	0.	1.
0.	0.	0.	0.	0.	0.
0.	0.	0.	1.	1.	0.
0.	0.	0.	0.	1.	0.

Lösungsmatrix für 1 DR

1.	1.	0.	1.	0.	0.
0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.
0.	0.	0.	1.	0.	0.
0.	0.	0.	0.	0.	1.
0.	1.	1.	0.	0.	0.
1.	0.	1.	0.	0.	1.
1.	0.	1.	0.	0.	0.
1.	0.	0.	0.	1.	1.
0.	1.	1.	0.	0.	0.
0.	0.	0.	0.	1.	0.
0.	0.	0.	1.	0.	0.
0.	0.	0.	0.	1.	0.
0.	0.	0.	0.	0.	1.
0.	1.	0.	1.	1.	0.
0.	0.	0.	0.	0.	0.

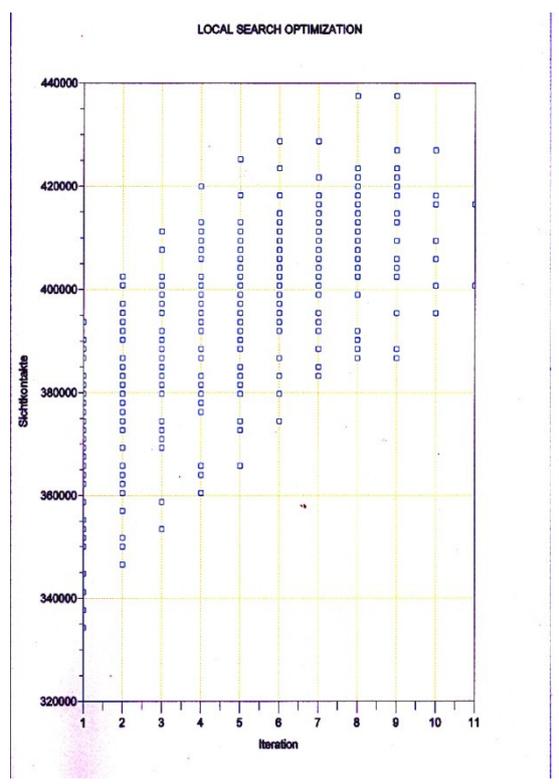
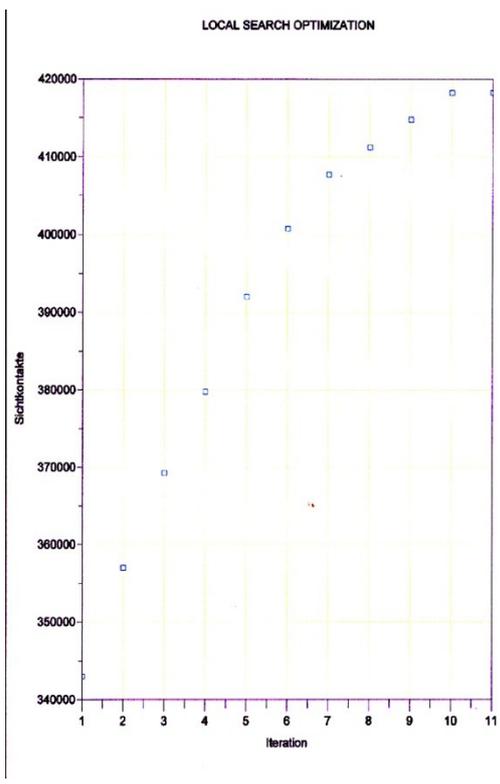


Diagramm 5.4: Local Search Optimization für 1 und 50 Datenreihen

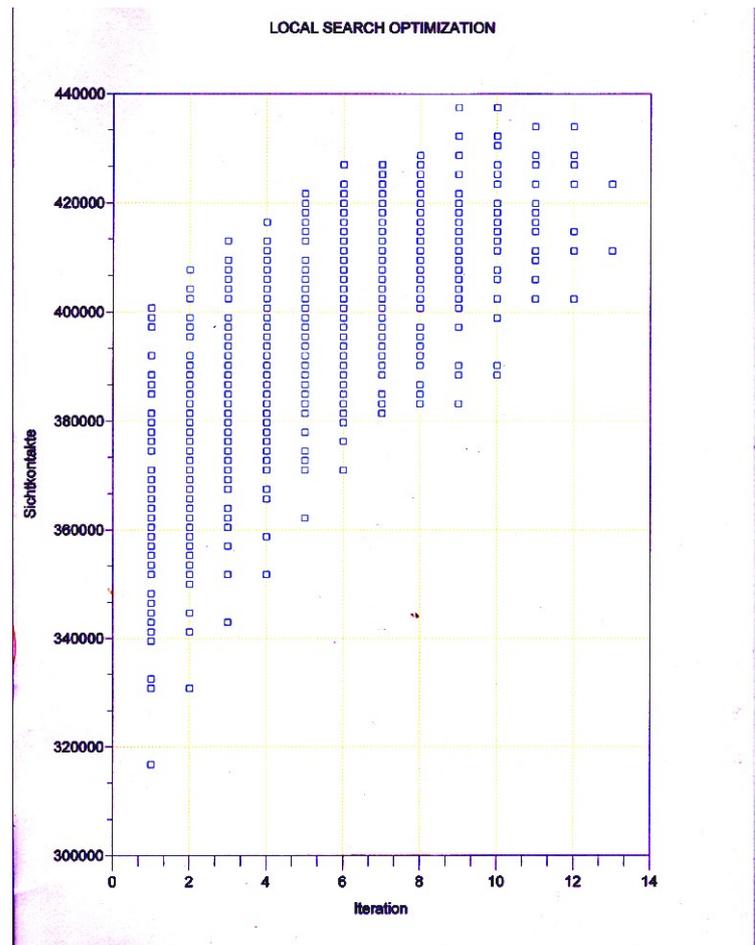


Diagramm 5.5: Local Search Optimization für 100 Datenreihen

Umfang der Datenreihe	Rechenzeit	Iterationen (Durchschnitt)	Lokal Optima (Durchschnitt)	Maxima
1	~5sek.	11	-	418.250
10	~50sek.	8,5	415.275,00	432.250
50	~6min	8,44	409.710,00	437.500
100	18min.	8,73	410.182,50	437.500

Tabelle 5.11: Daten der Rechenzeiten und lokalen Optima

Umfang der Datenreihe	Reihenfolge (pro Wochentag)						Gesamtkilometer
1	5.	8.	5.	11.	2.	4.	593,5
10	9.	4.	11.	5.	7.	2.	643,5
50	5.	8.	7.	8.	8.	4.	630
100	8.	6.	4.	11.	4.	1.	611

Tabelle 5.12: Daten der Reihenfolgen und Gesamtkilometer

## 5.3.3.2 Problem Instanz (0,02GE)

Lösungsmatrix für 100 DR

1.	0.	1.	0.	0.	0.
0.	0.	0.	0.	0.	0.
0.	0.	1.	0.	0.	0.
1.	0.	1.	0.	0.	0.
0.	0.	1.	0.	0.	0.
0.	0.	0.	0.	0.	1.
0.	0.	0.	0.	0.	0.
1.	1.	0.	1.	0.	0.
0.	0.	0.	0.	0.	0.
1.	0.	0.	1.	1.	0.
0.	0.	0.	1.	1.	0.
0.	0.	0.	0.	1.	0.
0.	1.	0.	0.	0.	1.
0.	1.	0.	0.	0.	0.
0.	1.	0.	0.	0.	1.
0.	0.	0.	1.	1.	1.
0.	0.	0.	0.	0.	0.

Lösungsmatrix für 50 DR

1.	1.	0.	1.	0.	0.
0.	0.	0.	0.	0.	0.
0.	1.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	1.
1.	0.	0.	0.	0.	0.
1.	1.	1.	0.	0.	0.
0.	0.	1.	0.	0.	0.
0.	0.	0.	1.	1.	0.
0.	0.	0.	0.	1.	1.
0.	1.	0.	0.	0.	0.
0.	0.	0.	1.	0.	0.
0.	0.	1.	0.	0.	1.
0.	0.	1.	0.	1.	0.
1.	0.	0.	1.	1.	0.
0.	0.	0.	0.	0.	1.

Lösungsmatrix für 10 DR

1.	0.	0.	1.	0.	0.
0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.
1.	0.	0.	0.	0.	0.
1.	0.	0.	0.	0.	1.
0.	1.	0.	0.	0.	0.
0.	1.	1.	1.	0.	0.
0.	0.	0.	0.	0.	0.
0.	0.	1.	0.	1.	1.
0.	1.	0.	0.	0.	0.
0.	0.	0.	1.	0.	0.
0.	0.	1.	0.	1.	0.
1.	0.	1.	0.	1.	0.
0.	0.	0.	0.	0.	1.
0.	1.	0.	1.	1.	0.
0.	0.	0.	0.	0.	1.

Lösungsmatrix für 1 DR

1.	0.	1.	0.	0.	0.
0.	0.	1.	0.	0.	0.
0.	0.	1.	0.	0.	0.
0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.
1.	0.	0.	0.	0.	1.
0.	0.	0.	0.	0.	1.
0.	1.	0.	1.	1.	0.
0.	0.	1.	0.	0.	0.
0.	0.	0.	0.	1.	1.
1.	1.	0.	0.	0.	1.
1.	0.	0.	1.	1.	0.
0.	0.	0.	0.	0.	0.
0.	0.	0.	1.	0.	0.
0.	1.	0.	0.	0.	0.
0.	0.	0.	1.	1.	0.
0.	1.	0.	0.	0.	0.

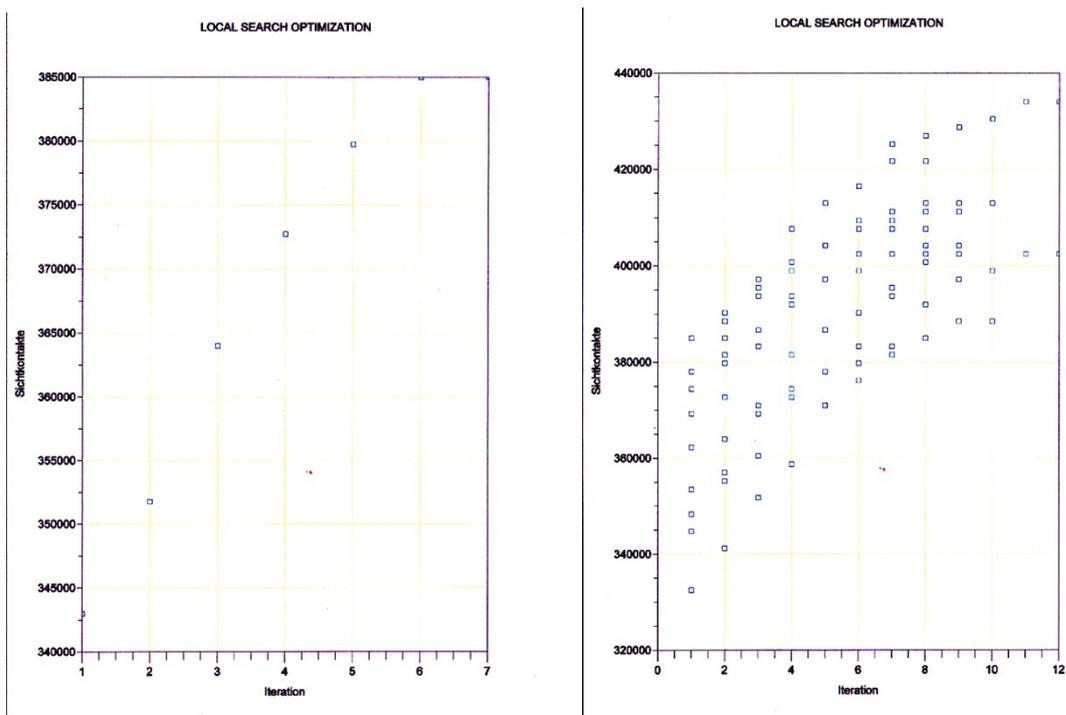


Diagramm 5.6: Local Search Optimization für 1 und 10 Datenreihen

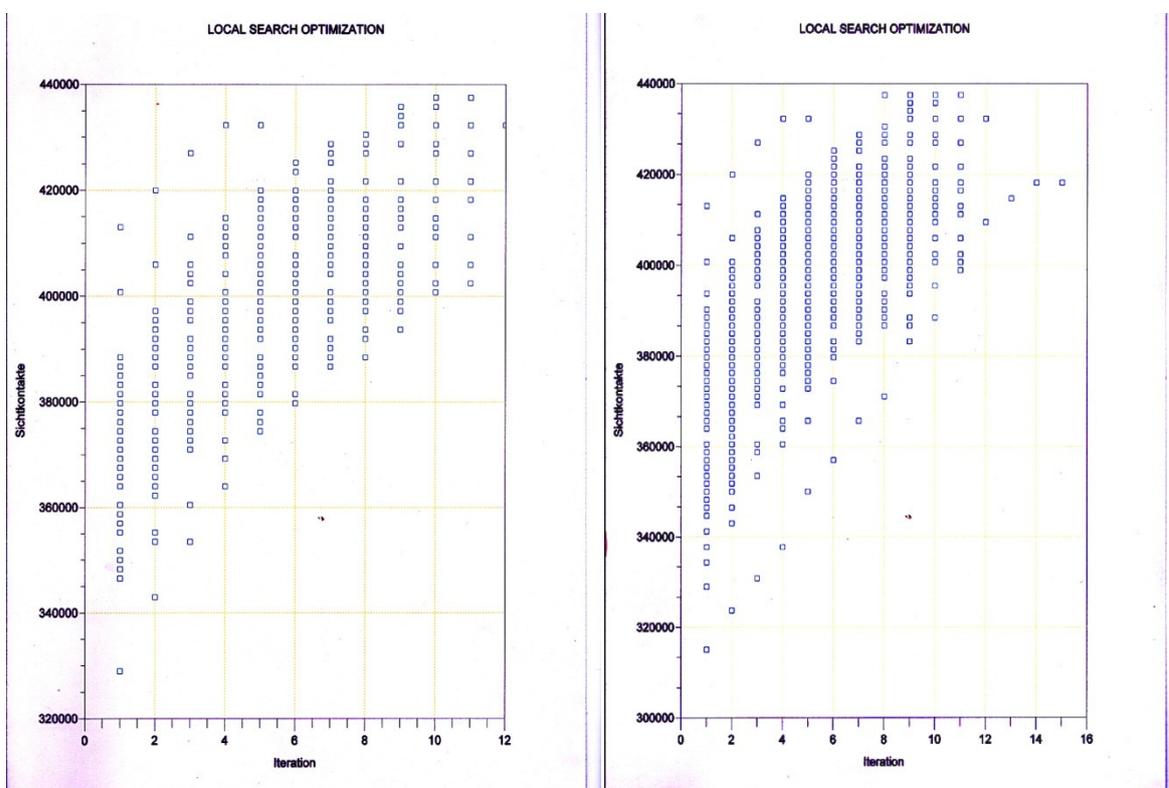


Diagramm 5.7: Local Search Optimization für 50 und 100 Datenreihen

Umfang der Datenreihe	Rechenzeit	Iterationen (Durchschnitt)	Lokal Optima (Durchschnitt)	Maxima
1	~5sek.	7,00	-	385.000
10	~45sek.	9,60	409.850,00	434.000
50	~6min	8,60	412.055,00	437.500
100	14min	8,64	411.092,50	437.500

Tabelle 5.13: Daten der Rechenzeiten und lokalen Optima

Umfang der Datenreihe	Reihenfolge (pro Wochentag)						Gesamtkilometer
1	8.	4.	3.	5.	6.	1.	586,5
10	7.	6.	11.	5.	4.	4.	621,5
50	6.	4.	6.	6.	7.	4.	648
100	5.	8.	7.	8.	8.	4.	660

Tabelle 5.14: Daten der Reihenfolgen und Gesamtkilometer

### 5.3.3.3 Probleminstanz (0.005GE)

Lösungsmatrix 01						
1.	1.	1.	0.	0.	0.	
0.	0.	0.	0.	0.	0.	
0.	0.	0.	0.	0.	0.	
0.	0.	0.	0.	0.	0.	
0.	0.	0.	0.	0.	0.	
0.	0.	0.	0.	0.	1.	
0.	0.	0.	0.	0.	0.	
0.	1.	1.	1.	0.	0.	
1.	0.	0.	0.	0.	0.	
0.	1.	0.	0.	1.	0.	
0.	0.	0.	0.	0.	0.	
0.	0.	1.	1.	0.	1.	
0.	0.	1.	1.	0.	0.	
1.	0.	0.	0.	1.	1.	
0.	1.	0.	0.	1.	1.	
0.	0.	0.	1.	1.	0.	
1.	0.	0.	0.	0.	0.	

Lösungsmatrix 02						
1.	1.	0.	1.	0.	0.	
0.	0.	0.	0.	0.	0.	
0.	0.	0.	0.	0.	0.	
0.	0.	0.	0.	0.	0.	
0.	0.	0.	0.	0.	0.	
0.	1.	0.	0.	0.	1.	
0.	0.	1.	0.	0.	1.	
1.	0.	0.	1.	1.	0.	
0.	0.	1.	0.	0.	0.	
0.	0.	0.	1.	1.	1.	
0.	1.	1.	0.	1.	0.	
0.	0.	0.	0.	0.	0.	
0.	0.	0.	0.	0.	0.	
0.	0.	1.	0.	1.	0.	
0.	1.	0.	0.	0.	0.	
1.	0.	0.	1.	0.	1.	
1.	0.	0.	0.	0.	0.	

Lösungsmatrix 03						
1.	1.	1.	0.	0.	0.	
0.	0.	0.	0.	0.	0.	
0.	0.	0.	0.	0.	0.	
0.	0.	0.	0.	0.	0.	
0.	0.	0.	1.	0.	0.	
0.	1.	0.	1.	0.	1.	
0.	0.	0.	0.	0.	1.	
0.	1.	0.	0.	1.	1.	
0.	1.	0.	0.	0.	0.	
0.	0.	0.	0.	1.	1.	
0.	0.	1.	0.	1.	0.	
0.	0.	0.	0.	0.	0.	
1.	0.	0.	1.	0.	0.	
0.	0.	1.	0.	0.	0.	
1.	0.	1.	1.	0.	0.	
1.	0.	0.	0.	0.	0.	

Lösungsmatrix 04						
0.	1.	1.	1.	0.	0.	
0.	0.	0.	0.	0.	0.	
0.	0.	0.	0.	0.	0.	
0.	0.	0.	0.	0.	0.	
0.	1.	0.	0.	0.	0.	
0.	0.	0.	0.	0.	1.	
1.	1.	1.	0.	0.	0.	
0.	1.	1.	0.	0.	1.	
0.	0.	0.	0.	0.	0.	
1.	0.	0.	0.	1.	0.	
0.	0.	0.	0.	1.	1.	
0.	0.	0.	0.	1.	1.	
0.	0.	0.	0.	0.	0.	
0.	0.	0.	1.	0.	0.	
1.	0.	1.	1.	0.	0.	
1.	0.	0.	1.	1.	0.	
0.	0.	0.	0.	0.	0.	

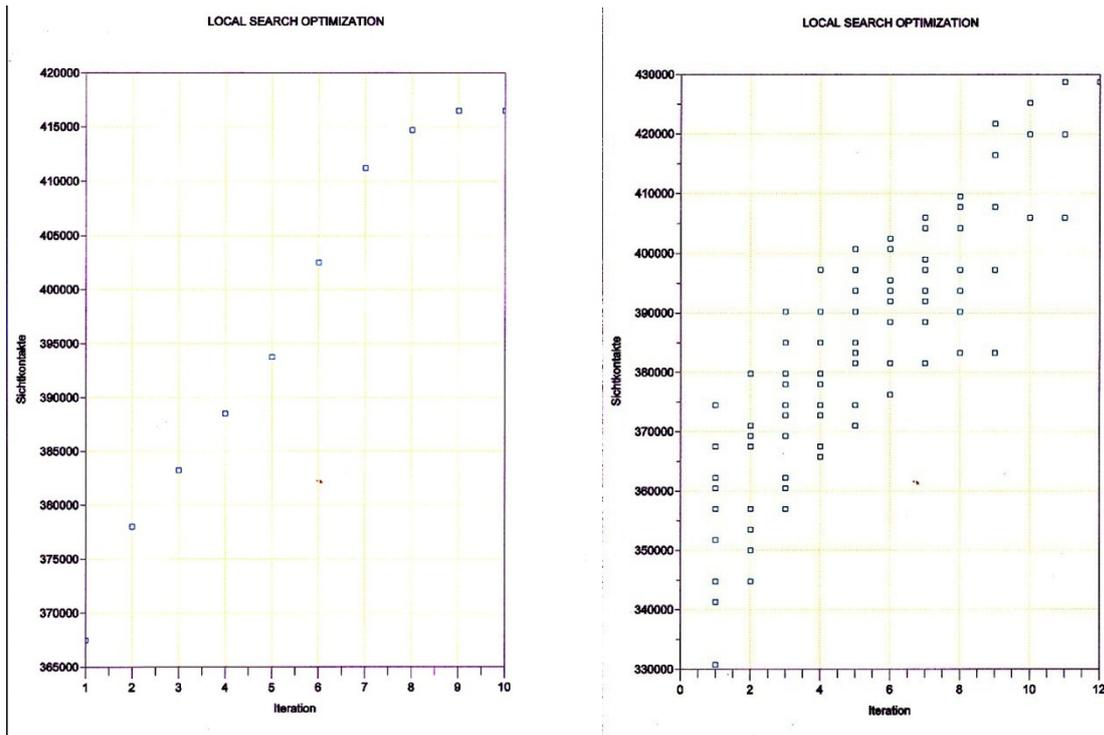


Diagramm 5.8: Local Search Optimization für 1 und 10 Datenreihen

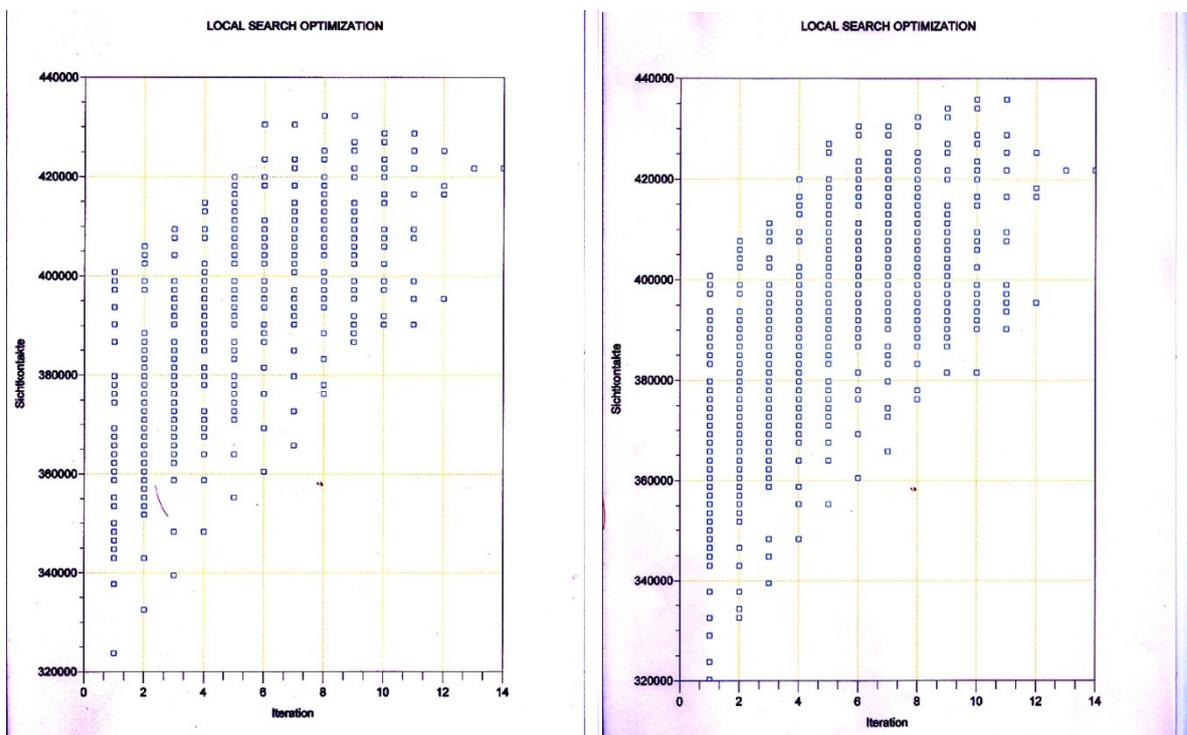


Diagramm 5.9: Local Search Optimization für 50 und 100 Datenreihen

Datenreihe		(Durchschnitt)	(Durchschnitt)	
1	~4sek.	10,00	-	416.500
10	~40sek.	9,00	402.850	428.750
50	~6min.	9,20	410.550	432.250
100	17,5min.	8,84	409.395	435.750

Tabelle 5.15: Daten der Rechenzeiten und lokalen Optima

Umfang der Datenreihe	Reihenfolge (pro Wochentag)						Gesamtkilometer
1	2.	9.	8.	6.	6.	4.	606,5
10	7.	7.	5.	5.	6.	2.	619,5
50	7.	1.	11.	11.	4.	2.	621,5
100	1.	6.	8.	6.	5.	4.	611

Tabelle 5.16: Daten der Reihenfolgen und Gesamtkilometer

### 5.3.3.4 Problem Instanz (0,001GE)

Lösungsmatrix für 100 DR						
1.	0.	1.	1.	0.	0.	
0.	0.	0.	0.	0.	0.	
0.	0.	0.	0.	0.	0.	
0.	0.	0.	0.	0.	0.	
0.	0.	0.	0.	0.	0.	
0.	0.	0.	0.	0.	1.	
0.	0.	0.	0.	0.	0.	
0.	1.	1.	0.	1.	0.	
0.	1.	1.	0.	0.	0.	
0.	0.	1.	1.	1.	0.	
1.	0.	0.	0.	1.	0.	
0.	0.	0.	0.	0.	1.	
1.	1.	0.	0.	0.	0.	
0.	0.	0.	1.	0.	0.	
1.	0.	0.	0.	0.	0.	
0.	1.	0.	1.	0.	1.	
0.	0.	0.	0.	1.	1.	

Lösungsmatrix für 50 DR						
0.	1.	1.	0.	0.	0.	
0.	0.	0.	0.	0.	0.	
0.	0.	0.	0.	0.	0.	
0.	0.	0.	0.	0.	0.	
0.	0.	1.	0.	0.	0.	
0.	0.	0.	0.	0.	1.	
1.	0.	0.	0.	0.	0.	
0.	1.	1.	1.	0.	0.	
0.	0.	0.	0.	0.	0.	
0.	0.	0.	0.	1.	1.	
1.	0.	0.	1.	0.	0.	
0.	0.	0.	0.	1.	1.	
1.	0.	1.	0.	0.	0.	
1.	0.	0.	0.	1.	0.	
0.	0.	0.	1.	0.	0.	
0.	1.	0.	1.	1.	0.	
0.	1.	0.	0.	0.	1.	

Lösungsmatrix für 10 DR						
1.	1.	1.	0.	0.	0.	
0.	0.	0.	0.	0.	0.	
1.	0.	0.	0.	0.	0.	
0.	0.	0.	0.	0.	0.	
0.	0.	0.	1.	0.	0.	
1.	1.	0.	0.	0.	1.	
0.	0.	0.	0.	0.	0.	
0.	1.	1.	1.	0.	0.	
0.	0.	0.	0.	0.	0.	
0.	0.	1.	0.	1.	0.	
0.	1.	0.	0.	0.	0.	
0.	0.	0.	0.	0.	1.	
1.	0.	0.	0.	0.	1.	
0.	0.	0.	0.	1.	0.	
0.	0.	1.	1.	0.	0.	
0.	0.	0.	1.	1.	1.	
0.	0.	0.	0.	1.	0.	

Lösungsmatrix für 1 DR						
1.	1.	0.	1.	0.	0.	
0.	0.	0.	0.	0.	0.	
0.	0.	0.	0.	0.	0.	
0.	0.	0.	0.	0.	0.	
0.	0.	0.	1.	0.	0.	
0.	0.	0.	0.	0.	1.	
0.	0.	1.	0.	0.	0.	
0.	1.	0.	1.	1.	0.	
0.	0.	0.	0.	1.	0.	
0.	1.	0.	0.	1.	0.	
0.	0.	0.	0.	0.	1.	
1.	1.	0.	0.	0.	0.	
1.	0.	1.	0.	0.	0.	
1.	0.	1.	0.	0.	1.	
0.	0.	1.	0.	0.	0.	
0.	0.	0.	1.	1.	0.	
0.	0.	0.	0.	0.	1.	

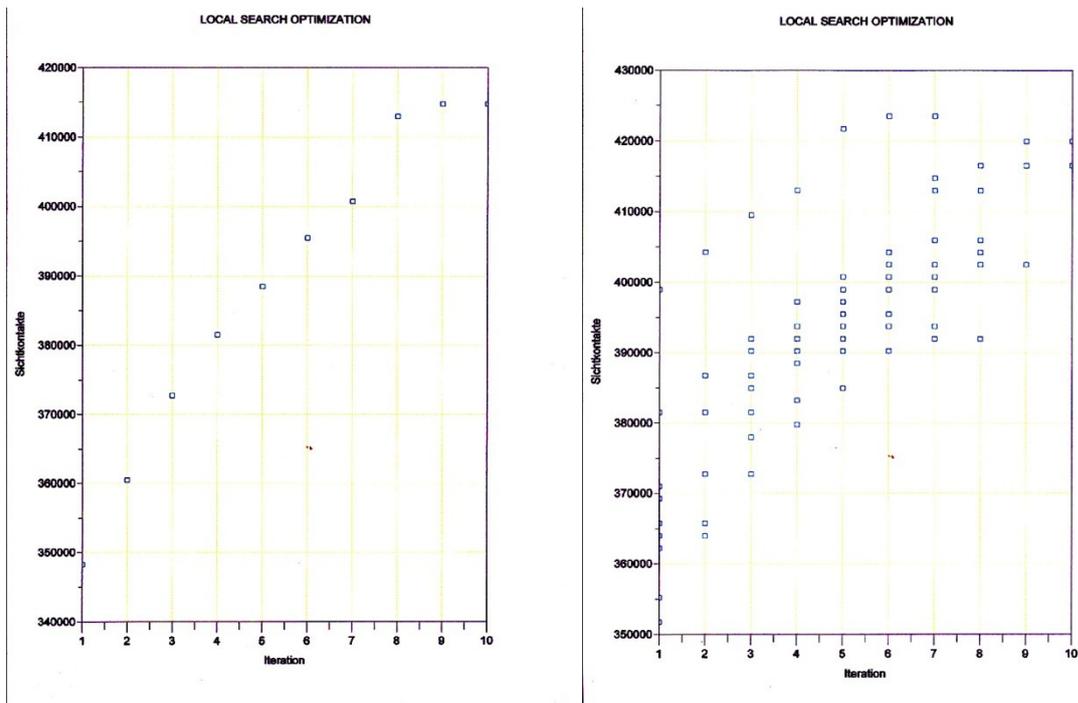


Diagramm 5.10: Local Search Optimization für 1 und 10 Datenreihen

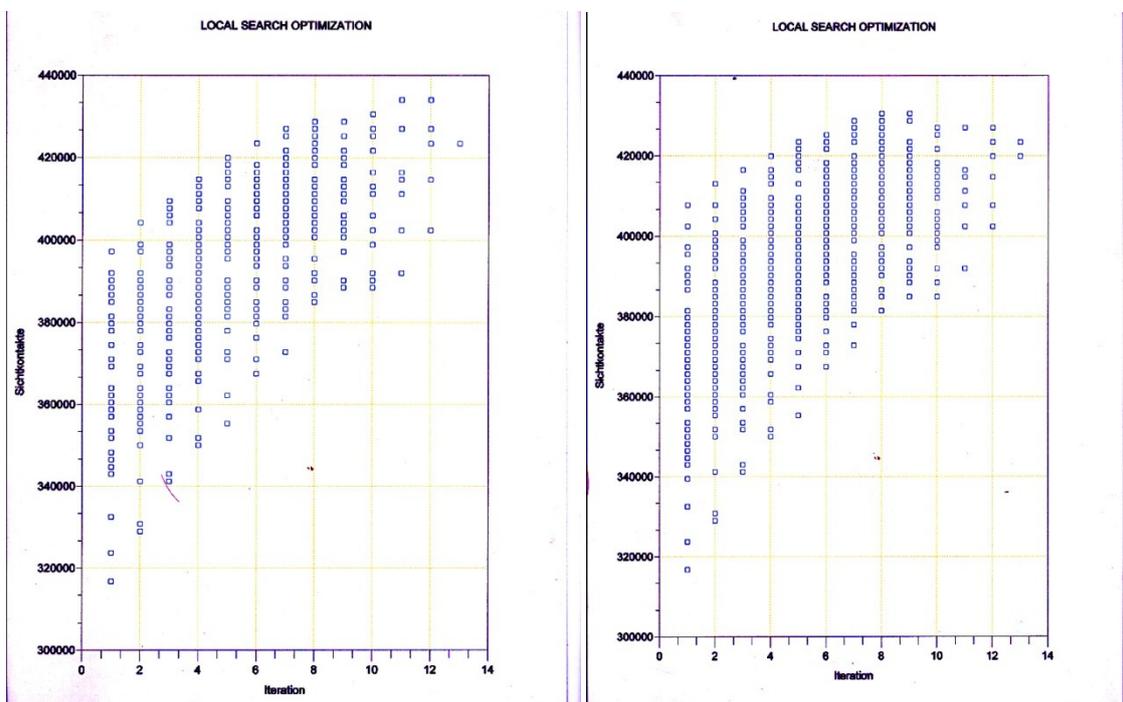


Diagramm 5.11: Local Search Optimization für 50 und 100 Datenreihen

Umfang der Datenreihe	Rechenzeit	Iterationen (Durchschnitt)	Lokal Optima (Durchschnitt)	Maxima
1	~4sek.	10,00	-	414.750
10	~45sek.	8,20	407.050	423.500
50	6,5min.	8,98	412.090	434.000
100	18,5min.	8,89	410.340	430.500

Tabelle 5.17: Daten der Rechenzeiten und lokalen Optima

Umfang der Datenreihe	Reihenfolge (pro Wochentag)						Gesamttour in km
1	1.	4.	4.	8.	10.	6.	605
10	8.	11.	1.	8.	7.	2.	621
50	7.	6.	11.	5.	4.	4.	621,5
100	6.	8.	8.	6.	7.	5.	625

Tabelle 5.18: Daten der Reihenfolgen und Gesamtkilometer

#### 5.5.4 Zusammenfassung

Je nachdem wie die Marketingabteilung des Unternehmens den Nutzen eines Sichtkontaktes bewertet, werden die Ergebnisse im Allgemeinen unterschiedlich beeinflusst. Eine Veränderung der Zielfunktionskoeffizienten definiert daher eine neue Zielfunktion und damit auch eine neue Probleminstance. Anhand dieser neuen Zielfunktion wird dann die Ergebnisqualität gemessen. Die Zielfunktionswerte der verschiedenen Probleminstance untereinander werden nicht verglichen. Anhand der obigen Auswertungen ist vorweg sehr deutlich erkennbar, dass eine der Grundvoraussetzungen für eine gute Performance, eine geeignete Wahl des Umfanges der Datenreihen darstellt. Erst wenn die Datenreihe entsprechend groß gewählt wurde, lassen sich überhaupt Rückschlüsse auf die Ergebnisstruktur ziehen. Demnach erscheint eine Datenreihe ab etwa 50 Neustarts als sinnvoll. Diese Problematik wird bei nur einem Neustart deutlich.

Bewertet nun das Unternehmen den Nutzen eines Sichtkontaktes höher als im konkreten Fall angenommen, genauer gesagt mit 0,02 GE oder 0,05 GE, dann verschiebt das Kriterium zu Gunsten der Sichtkontakte. Das Modell wird also mit dem neuen gewichteten Werten versuchen eine höhere Anzahl an Sichtkontakten zu erreichen, da die Gesamtkilometer der Touren diesen weniger beeinflussen. Hier sollte aber auch festgehalten werden, dass es nicht unwesentlich auf die Stärke (Delta) der Veränderung des Wertes ankommt.

Dem gegenüber steht eine Verschiebung des Kriteriums zu Gunsten der Fahrtkosten. Das Unternehmen bewertet also einen Sichtkontakt mit einem geringeren Nutzen als 0,01 GE. Hier wurden Werte von 0,005 GE und 0,001 GE gewählt. Nachdem jetzt die Kilometer und die damit verbundenen Kosten stärker gewichtet worden sind, werden leicht geringere Sichtkontaktwerte erzielt.

Die Rechenzeiten der fünf Instanzen sind in etwa gleich. Grundsätzlich werden in allen getesteten Instanzen gute Ergebnisse erzielt.

Würde das Projekt landesweit geplant sein, dann wären die Auswirkungen auf aufgrund der größeren Distanzen auf die Ergebnisstruktur wesentlich stärker.

## 6. SCHLUSSFOLGERUNGEN

Da schon im vorangegangenen Kapitel die Ergebnisse reichlich analysiert wurden, werde ich in diesem Teil zunächst die wichtigsten Erkenntnisse darüber in knapper Form wiedergeben. Anschließend wird ein Ausblick auf eine Reihe von Aspekten zur Vorhersage der Ergebnisstruktur bei bestimmten Veränderungen gegeben, die in dieser Arbeit nicht behandelt wurden, jedoch als interessant erscheinen.

### 6.1 Die wichtigsten Ergebnisse dieser Arbeit

Es ist möglich eine sehr gute Repräsentation der Ergebnisstruktur bei Optimierungsproblemen durch folgende drei Faktoren zu erhalten:

#### 6.1.1 Informationsqualität

Zu Beginn eines Entscheidungsprozesses müssen sämtliche Informationen, welche in Abhängigkeit zum vorliegenden Problem stehen, evaluiert werden. Diese Phase der Informationsgewinnung ist sehr bedeutend für den weiteren Verlauf eines Optimierungsproblems. Die durch die Auswertung gewonnenen Informationsmengen tragen wesentlich zum Erfolg geplanter Maßnahmen bei. *„Die Informationsgewinnung ist durch motivierte Suchprozesse gekennzeichnet und beginnt infolge realisierter oder erwarteter Zielerfüllungen.“*<sup>27</sup> Infolge sollte die Modellierung des Optimierungsproblems auf eine ausführliche und umfangreiche Informationsgrundlage basieren.

#### 6.1.2 Modell

Bei der Formulierung des Modells muß auf mehrere Aspekte geachtet werden. Zumal bereits die Struktur vorhandener Informationen Aufschlüsse über die Auswahl des Entscheidungsmodells gibt. Die Zielfunktion und die Nebenbedingungen bilden den Kern des mathematischen Modells. In der Arbeit wurde gezeigt, dass ein kombiniertes Standort-Tourenproblem die beiden Zielvorgaben optimal erfüllt. Ziel war es die zwei Zielfunktionen zu einer zusammenzuführen, um so das Problem auch mit geeigneter Software programmieren zu können. In der konkreten Aufgabenstellung wurden zudem zwei wichtige Nebenbedingungen formuliert. Diese bezogen sich allerdings ausschließlich auf die vorliegende Probleminstanz. Um weiter Erkenntnisse diesbezüglich zu erlangen, werde ich anschließend diskutieren, wie die Ergebnisse bei Veränderungen an den Nebenbedingungen beeinflusst werden.

---

<sup>27</sup> Bamberg, Coenenberg [2002], WiSo Kurzbücher – Reihe: Betriebswirtschaftliche Entscheidungstheorie, 11. Auflage

### 6.1.3 Lösungsverfahren

Im Gegensatz zu exakten Lösungsverfahren wurden unter Verwendung heuristischer Lösungsverfahren akkurate Ergebnisse in akzeptabler (polynomialer) Rechenzeit erzielt. Änderungen der Zielfunktionskoeffizienten brachten keine nennenswerten negativen Auswirkungen auf die Rechenzeiten.

## 6.2 Veränderung anderer Parameterwerte

Folgende, mögliche Veränderungen bzw. untersuchenswerte Punkte zum vorgestellten Optimierungsmodell bieten sich an:

- Veränderung der rechten Seite bei Nebenbedingungen eins: Beispielsweise sollen mehr als vier Standorte pro Tag angefahren werden. Welche Auswirkungen hätte so eine Veränderung auf das Ergebnis und auf die Rechenzeit?  
In der vorliegenden Arbeit wurde impliziert, dass genau ein Klein-LKW zur Verfügung steht, dessen Kapazität mit 100.000 Stück Produktproben beschränkt ist. Diese Beschränkung hatte bei der Wahl, vier Standorte anzufahren, noch keinen Einfluss. Bei mehr als vier besuchten Standorten hingegen müsste der eingesetzte Fuhrpark erweitert werden. In diesem Fall würden sich die zu erwarteten Sichtkontakte um rund 3-5% erhöhen. Doch die Projektkosten würden vergleichsweise viel sensibler auf derartige Veränderungen reagieren. Respektive würden die Kosten um 30-40% steigen. Es wäre unwirtschaftlich fünf Standorte anzufahren. Würde man beispielsweise sieben oder acht Standorte anfahren, dann verringern sich die marginalen Kosten. Auf der anderen Seite erhöht sich der Nutzen (Maximierung der Sichtkontakte). Hieraus wird ersichtlich, dass durch diese Veränderung das Tourenproblem aufwendiger wird. Würde die andere Nebenbedingung gleich bleiben, wäre der zulässige Lösungsraum kleiner und infolge würden sich auch die Rechenzeiten erhöhen.
- Veränderung der rechten Seite der Nebenbedingung Zwei: Ein Standort darf mehr als drei Mal angefahren werden. Hier stellt sich allen voran die entscheidende Frage, welche Aspekte und Projektziele für das Unternehmen im Vordergrund stehen. Zielt man auf Quantität ab, dann könnte man diese Beschränkung lockern bzw. erweitern. Dürfte ein Standort beispielsweise dann viermal besucht werden, könnte man in einem geringen Ausmaß mehr Sichtkontakte erzielen. Allerdings würde diese leichte Ergebnisverbesserung auf Kosten der Diversität zurückzuführen sein. Aufgrund des größeren Lösungsraumes würden sich die Rechenzeiten verringern.  
Reduziert man hingegen die Zahl der Standortbesuche auf maximal zwei pro Tag, verkleinert sich der zulässige Lösungsraum deutlich und die Ergebnissuche würde mehr Zeit in Anspruch nehmen.

Mit dem in dieser Arbeit gewählten methodischen Verfahren wurden für die konkrete Problem Instanz (drei Besuchen pro Tag und pro Standort) sowohl optimale Ergebnisse als auch akzeptable Rechenzeiten erzielt. Im Falle einer deutlichen Erhöhung bei der Zahl der Standortbesuche, würde sich aber die Lösungsstruktur wesentlich verändern. Bei vier anzufahrenden Standorten pro Tag, konnte noch eine vollständige Enumeration aufgrund der geringen Anzahl der Gesamtmöglichkeiten (hier waren es 12 Permutationen) durchgeführt werden. Welche Vorgangsweise

würde man wählen, wenn beispielsweise 10 Standorte pro Tag besucht werden sollen. Wendet man die Formel aus Kapitel 3.3 zur Berechnung der Permutationen an, dann würde das 1.814.400 zulässige Reihenfolgen ergeben. Es müßten also pro Iteration 1.814.400 Touren durchsucht werden. Eine vollständige Aufzählung wäre hier unwirtschaftlich und zugleich äußerst ineffizient. Zur Lösung eines solchen Problems, müsste ein spezieller Algorithmus wie etwa das Branch-and-Cut Verfahren (exakt) oder die Tabu Search (heuristisches Lösungsverfahren) implementiert werden.<sup>28</sup>

---

<sup>28</sup> Literaturverweis: Heinrici [1996], Leistungsvergleich von Nachbarschafts Suchverfahren, VWF Verlag für Wirtschaft & Forschung, Berlin beschäftigt sich ausführlich mit den heuristischen Lösungsverfahren kombinatorischer Optimierungsprobleme  
Schiemann [2005], Lösungsverfahren für das 2-dimensionale Travelling Salesman Problem, Logos Verlag Berlin, 1. Auflage

## 7. LITERATURVERZEICHNIS

Wayne L. Winston [2005], Operations research – Applications and Algorithms, International Student Edition, Fourth Edition

Klaus Neumann, Martin Morlock [2002], Operations research, Carl Hanser Verlag München Wien, 2. Auflage

Stephan Dempe, Heiner Schreier [2006], Operations research – Deterministische Modelle und Methoden, International Student Edition, 1. Auflage

Frederick Hiller, Gerald Liebermann [1988], Operations Research, R. Oldenbourg Verlag München Wien, 4. Auflage

Wolfgang Grundmann [2002], Operations research – Formeln und Methoden, Teubner Verlag Stuttgart/Leipzig/Wiesbaden, 1. Auflage

Domschke, Drexl [2005], Einführung in die Operations Research, Springer Verlag, 6. Auflage

Werners Brigitte [2008], Grundlagen des Operations Research, Springer Verlag, 2. Auflage

Klaus-Peter Kistner [1993], Optimierungsmethoden, Physica Verlag Heidelberg, 2. Auflage

Helmut Laux [1982], Entscheidungstheorie – Grundlagen, Springer Verlag Berlin Heidelberg New York, 1. Auflage

Helmut Laux [1995], Entscheidungstheorie, Springer Verlag Berlin Heidelberg New York, 3. Auflage

Günter Bamberg, Adolf Coenenberg [2002], Betriebswirtschaftliche Entscheidungslehre, Franz Vahlen Verlag München, 11. Auflage

Jukka Siedentopf [1996], Eine exemplarischer Vergleich lokaler Suchverfahren (Teil I), Universität Leipzig – Institut für Produktionswirtschaft und Industrielle Informationswirtschaft

Hokey Min, Vaidyanathan Jayaraman, Rajesh Srivastava [1997], Combined location-routing problems – A synthesis and future research directions, European Journal of Operational Research, 108(1998) 1-15

Dilek Tuzun, Laura I. Burke [1998], A two-phase tabu search approach to the location-routing problem, European Journal of Operational Research, 116(1999) 87-99

Gabor Nagy, Said Salhi [2006], Location-routing – Issues, models and methods, European Journal of Operational Research, 177 (2007) 649-672

Domschke, Scholl [2006], Heuristische Verfahren (Arbeits- und Diskussionspapiere)  
Universität Jena

Fink, Rothlauf [2006], Heuristische Optimierungsverfahren in der Wirtschaftsinformatik, Universität Mannheim

## Kurzfassung

Diese Arbeit beschäftigt sich mit der Optimierung eines konkreten Entscheidungsproblems, welches bei der Einführung eines neuen Produktes im Lebensmittelbereich auftritt. Das Konzept der integrierten Logistik- und Vertriebssysteme ist bei solchen Entscheidungssituationen aus dem Management global agierender Unternehmen nicht mehr wegzudenken. Es trägt vor allen dazu bei, in diesem speziellen Fall auch bei einer Produktneueinführung, die Wirtschaftlichkeit und Effizienz während des Vertriebsprozesses zu steigern. Die zwei tragenden Säulen bei einem integrierten Logistik- und Vertriebssystem bilden die Standortwahl und die Tourenplanung. Bei der Standortwahl zielt man allgemein auf die Nutzenmaximierung ab, wogegen man bei der Tourenoptimierung versucht Kosten zu minimieren. Diese beiden Optimierungsansprüche werden in der vorliegenden Arbeit kombiniert und zu einem einzigen Standort-Tourenproblem zusammengefaßt. Der Grund dafür ist die gegenseitige Abhängigkeit zwischen der optimalen Standortwahl der Einrichtungen (hier: temporäre Sampling Werbestände), der Berücksichtigung von kapazitätsbedingter Restriktionen und der Touren bzw. Routenstruktur.

Die Arbeit geht von folgender Problemstellung aus: Es liegt ein reales Entscheidungsproblem eines Molkereiunternehmens vor, dessen Aufgabe darin besteht, die optimalen Standorte zu ermitteln, um möglichst viele Endkonsumenten auf die Produkteinführung aufmerksam zu machen. Ein Standort ist dann optimal, wenn einerseits die Sichtkontakte maximiert und andererseits die Projektkosten minimiert werden.

Zunächst werden in der Arbeit einige Grundlagen auf dem Gebiet der Operations Research erläutert, die später dazu beitragen werden, ein entsprechendes mathematisches Modell für dieses Problem aufzustellen. Mit der Formulierung des Standortmodells wird als nächster Schritt ein geeignetes Lösungsverfahren gesucht. Aufgrund der Absicht einerseits die Rechenzeiten so gering wie möglich zu halten und andererseits akzeptable Ergebnisse zu liefern, wurde ein iterativer heuristischer Lösungsansatz ausgewählt. Dieser entspricht einer angepaßten Nachbarschaftsheuristik und ist in der Literatur als Iterated Local Search bekannt.

Um beide Zielvorgaben gleichermaßen befriedigen zu können, wird ein gewichtetes Kriterium eingeführt, welches anschließend maximiert wird. Den Kern dieser Arbeit bildet die Implementierung des Modells und der Lösungsmethode in das Softwareprogramm SCILAB Version 4.1.2 (Das Programm wird unter Lsg-version\_7.0.sce gespeichert).

## Abstract

This paper deals with the optimization of a concrete decision problem which appears often when a new product will be launched on the market. The concept of integrated logistics and distribution-oriented systems has emerged to one of the central part of transnational companies. It aims to increase distribution efficiency and supports the economic behaviour. The supporting elements of such a concept are facility location decision and tour-planing. Regarding to the facility location problem it's goal is to maximize the utility whereas the routing problem intends to minimize the project costs. Instead of working with to two different problems, this paper declares one problem – known as a Combined Location-Routing Problem. The reason for this are the interdependences among the location of the facilities (actually in this paper the facilities are temporary sampling stands), the consideration of restrictions and the vehicle route structure around depots.

At the beginning we are faced with a decision problem of a dairy enterprise which tries to pursue the estimate goal of calling best attention for the new product on the market. In this special case the marketing department decided to plan a sampling promotion cross Austria starting with the capital. The challenge for the assigned agency will be to find the optimal locations. A facility location is optimal if on the one hand the intervisibility will be maximized and on the other hand the project costs will be minimized.

Initially a few basics referring to optimalization will be defined. They will help us to formulate the mathematical standard model. The proximate emphasis during the decision process will be the creation of a suitable solving method. The requirements for such a method were to avoiding long computing times and to achieving optimal results. By using the method of an Iterated Local Search we fulfill this requirements. To meet both objective functions in equal measure the implementation of a weighted criteria is essential. This criteria combines the location with the touring problem and will be maximized. The objective of this paper is to implement the model and the solution procedure in the softwareprogramm SCILAB 4.1.2 (Saved as the File Lsg\_Version\_7.0.sce).

## Danksagung

Bei Herrn Univ. -Prof. Dr. Walter Gutjahr möchte ich mich herzlich dafür bedanken, dass er mir die Möglichkeit gegeben hat, meine Diplomarbeit über dieses interessante Thema zu schreiben. Zumal ich ihm zu großem Dank für seine Unterstützung und Betreuung während der Erstellung dieser Arbeit verpflichtet bin.

Den Mitarbeitern der Contact Werbe- und Promotionagentur GmbH möchte ich ebenfalls danken. Sie lieferten mir die relevanten Daten für die konkrete Aufgabenstellung und gaben mir nützliche Hinweise bei der Planung, Organisation und Durchführung solcher Projekte.

Abschließend möchte ich meinen großen Dank gegenüber meinen Eltern und Mag. Andrea Konstantinova aussprechen, die mich auf meinen bisherigen Lebensweg stets mit vollen Einsatz unterstützt haben. Ihnen ist diese Arbeit gewidmet.

## Lebenslauf

### Persönliche Angaben

Name: **Christian Strobach**  
 Adresse: Nussdorferstrasse 55/6  
 A-1090 Wien  
 Telefon: +4368110734393  
 E-Mail: christian.strobach@gmail.com  
 Geburtsdatum: 21.09.1974  
 Geburtsort: Wien, Österreich



### Ausbildung

---

**2009 - 2002**      **Diplomstudium Internationale Betriebswirtschaft**  
 an der Hauptuniversität Wien, Österreich  
 Kernfachkombination und Diplomarbeit: **Operations Research**

**2000 - 1997**      **Psychologie**  
 an der Hauptuniversität Wien, Österreich

**1995 -1989**      **HTBLA (Höhere Technische Berufsschule)**  
 Department: **Nachrichtentechnik**

### Berufserfahrung

---

**seit Dez. 2007**      **Handelsgewerbe**

Großkundenbetreuung  
 Kunden: Unilever, Sony, Siemens  
 Umsetzung verkaufsfördernder Marketingmaßnahmen  
 Optimierung von Entscheidungsprozessen  
 Projektplanung und Koordination  
 Monitoring und Überwachung für aktive Vermarktung

**Jän.2009-Jul.2009**      **JULAND BarcelonaVienna, Wien, Österreich**

Assistenz der Geschäftsleitung  
 Projektmanagement und Produkt Marketing Management  
 Projektplanung und Koordination

**Jun. 2007 – Jun. 2001    Contacts Werbeagentur GmbH, Wien, Österreich**

Produkt Marketing Management  
Umsetzung verkaufsfördernder Marketingmaßnahmen  
Mitarbeiterführung Außendienst  
Planung und aktive Umsetzung von POS (Visual Merchandising)  
Tourplanung

**Jun. 2001 –Jun. 2000    Cruz Roja (Rotes Kreuz), Santa Cruz, Teneriffa, Spanien**

Marketing Management  
Projekt- und Eventplanung

**Mai. 2000 – Dez. 1999    UTA Telekommunikation GmbH**

Großkundenbetreuung  
Beratung und Betreuung bestehende Verträge

---

**Sprachen**

---

Deutsch (Muttersprache)  
Englisch (Verhandlungssicher)  
Spanisch (Verhandlungssicher)

---

**Informatik**

---

**Scilab / Matlab** (Numerische Mathematik),  
PC/ MAC: MS-Office Paket (Word, Excel, Outlook, Powerpoint), Adobe Photoshop,  
Flash

---

**Hobbies**

---

Kultur, Kunst, Reisen, Sport