Vehicle Routing

Martin Spoden

13. Juli 2000

Inhaltsverzeichnis

1 Modell und Realität **2**

1.1 Die Standarddarstellung des *vehicle routing problem (VRP)* 2

1.2 Das reale Einzeltransportproblem 4

1.3 Ausgewählte technische Lösungen 5

1.4 Zeitvorgaben 5

1.5 Kosten 5

1.6 Einschränkungen des Modells gegenüber der Realität 7

1.7 Bedarf an rechnergestützter Optimierung von VRPs 9

2 Algorithmen und Heuristiken **10**

2.1 Formale VRP-Formulierung 10

2.2 Übersicht und Geschichtliches 13

2.3 Suchtechniken 18

2.3.1 tabu search 18

2.3.2 simulated annealing 19

2.4 general assignment problem = VRP 19

2.5 K-Baum-Relaxation 20

3 Vergleich der besten Lösungen, Optima **24**

Kapitel 1

Modell und Realität

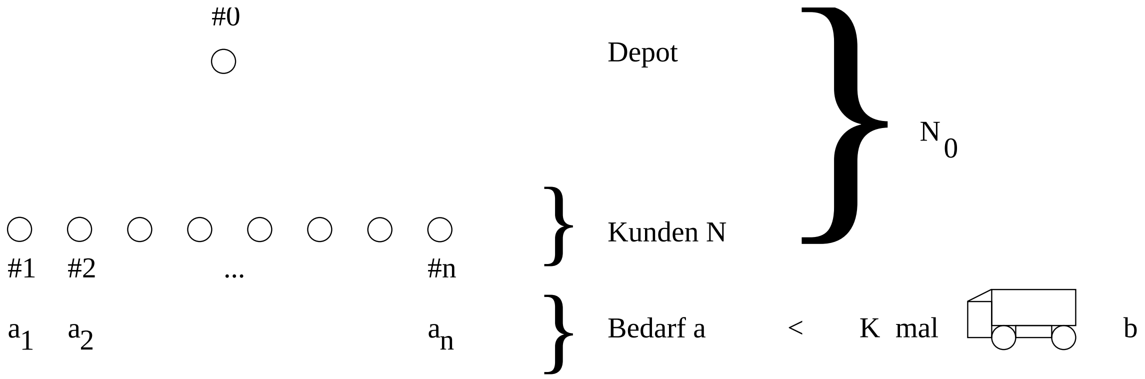
1.1 Die Standarddarstellung des *vehicle routing problem (VRP)*

Definition 1.1.1 (VRP) *Das VRP besteht darin, K Fahrzeugrouten anzugeben, wobei eine Route am Depot startet, eine Untermenge von Kunden in angegebener Reihenfolge durchläuft und beim Depot wieder endet. Jeder Kunde muß dabei genau einer der K Fahrzeugrouten zugeordnet werden, und die Gesamtmenge der Lieferungen der den einzelnen Fahrzeugen zugeordneten Kunden darf die jeweilige Fahrzeug-Kapazität b nicht überschreiten. Die Routen sollen zu minimalen Gesamtkosten gewählt werden.*

Weitere Grundannahmen:

- *Das Transportnetz ist als ungerichteter Graph* $G = (V, E), V := N_0, E := N_0 \times N_0$ *gegeben.*
- *Die Kosten sind nur von den gefahrenen Einzelstrecken abhängig. Sie sind symmetrisch:*
  Kostenfunktion $c \geq 0$ *definiert auf* E *mit*

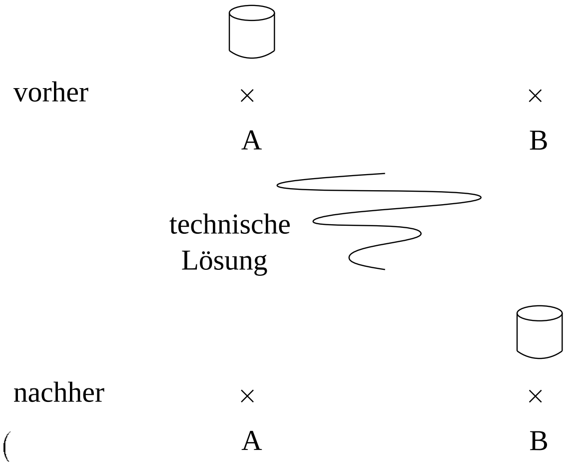
$$\bigwedge_{e=ij \in E} c_e = c_{ij} = c_{ji} := c(e)$$



Anmerkung:
Der Spezialfall $K = 1$ führt dazu, daß $\Sigma_{i=1}^{n} a_i \leq b$ sein muß; damit fällt die Kapazitätsrestriktion weg, und das VRP ist zum *traveling salesman problem (TSP)* ausgeartet. Da für $K > 1$ eine Lösung gesucht wird, die global minimal ist, sind auch die K einzelnen Touren optimal anzugeben, folglich ist das VRP $\mathcal{NP}$schwer.
Anmerkung:
Die korrekte Dateneingabe, wenn die Bestellmenge aufteilbar ist, wäre, a_i Einheiten viele Kunden zu entwerfen, um zu modellieren, daß der reale Kunde auch von mehreren Fahrzeugen (teil-)beliefert werden kann...

Da in der Definition implizit diverse Annahmen zum allgemeinen Transportproblem hinzugefügt werden und alle Annahmen bzw. Vorgaben in der Praxis nicht allgemeingültig sind, lohnt es sich, diese Einschränkungen in einer Modellbildung zu erarbeiten, um mögliche Erweiterungen des Modells oder auch der Algorithmen einschätzen zu können.

1.2 Das reale Einzeltransportproblem



(mit gewissen Zeitvorgaben)

Lösungsvarianten

- $\underline{A \equiv B}$
  Die Standortwahl geschieht anhand von Standortfaktoren (z.B. Nähe zu Lieferanten, Kunden oder zur Konkurrenz, je nach Arbeitsmarkt, politischer Situation, etc.). Standorte werden wirtschaftlich so gewählt, daß die entstehenden Transportkosten minimiert werden oder zumindest moderat bleiben.
  Meist ist die Standortwahl schon gefallen, und die Standorte sind im Vergleich zum Transportproblem (Zeiträume von Stunden bis Monaten) quasi statisch (durch langfristig finanzierte Investitionssummen, Genehmigungsverfahren, etc.).

- Pipelines
  Für strömende Güter mit begrenztem Fluß → Netzwerkflußprobleme
  Anmerkung: Sind Ein-, Aus- und Umlade- sowie Leerfahrtkosten zu vernachlässigen, gilt die Pipeline-Modellierung auch für Fahrzeuge.

- bewegliche Transportmittel
  allgemein oder auf spezielle Transportgüter zugeschnitten, je mit beschränkter Kapazität, typischer Geschwindigkeit und typischer Bewegungsfreiheit; meist ist Transportpersonal notwendig, in jedem Fall jedoch Wartung und Reparatur;

1.3 Ausgewählte technische Lösungen

Der Bewegungsfreiraum ist meist eingeschränkt durch verschiedenste Vorgaben (nur Beispiele):

Infrastruktur	*Transportmittel*	*mögliche Einschränkungen*
Wasser (Meere, Seen, Flüsse)	Tanker, Frachter, Hovercraft	Land (z.B. Inseln), 7-Meilen-Zone, Untiefen (Wattenmeer, Riffe, etc.), Strömung
Land (Straße)	LKW, PKW, Busse	vorgegebenes Straßennetz, Einbahnstraßen, Gewichtsbeschränkungen
Land (Schienen)	Deutsche Bahn, U-Bahn, S-Bahn	Gleissystem, Fahrplan, Sicherheitsmaßnahmen
Luft (zwischen Flughäfen)	Zeppelin Flugzeug, Hubschrauber,	Start- und Landefenster der Flughäfen; Luftraum, Militär, Sperrgebiete

Der Transport über Wasser ist relativ langsam, dafür aber für große Mengen auch über lange Strecken geeignet. Größere Mengen sind sonst höchstens noch über die Bahn abzuwickeln. Das Schienennetz ist entsprechend etwas dünner als das Straßennetz, das pro Fahrzeug nur geringe Mengen, diese allerdings sehr schnell und präzise transportieren kann. Lufttransporte sind für kleine bis mittlere Mengen geeignet und bei langen Strecken sicherlich die schnellste Alternative, dafür aber auch entsprechend teuer.

1.4 Zeitvorgaben

In der Praxis gibt es diverse Zeitvorgaben, die unter allen Umständen oder evtl. nur ungefähr einzuhalten sind. Das bezieht sich zum einen auf Zeitfenster, die am Depot oder beim Kunden auftreten, und zum anderen auf Zeitbeschränkungen für die gesamte Tour (z.B. wegen verderblicher Ware oder maximaler Arbeitszeit). Verletzungen können im Modell erlaubt werden, wobei meist eine Auswirkung auf die Kosten auftritt.

1.5 Kosten

Alle beteiligten Objekte können Kosten verursachen; die wirkliche Kostenfunktion ist meist unbekannt, so daß man Schätzungen und Heuristiken benutzen muß!

Beispiele:

- Kosten für die Verkehrsmittel
  - Parkplätze/Liegestellen
  - Wartung und Reparatur (auch nicht-benutzte Fahrzeuge)
  - Je nach zurückgelegter Wegstrecke
- Kosten für das Transportgut
  - Versicherungen
  - Technische Spezialanforderungen (z.B. Umbau der Ladefläche eines LKWs)
  - Je nach Qualität und Menge (z.B. Kühlung)
- Kosten für das Transportpersonal
  - Je nach Zahl und Qualifikation
  - Überstunden-, Nacht-, Wochenend-, Akkordzuschläge
- Kosten für das Be- und Entladen
  - Je nach Güterklasse und Menge
- Kosten für den Verkehrsweg
  - MAUT, Plaketten (kann Transportdauer beeinflussen!)
  - Schleusen, Kanäle
- Kosten für das Verletzen von Terminvorgaben
  - Konventionalstrafe bei verspäteter Lieferung
  - Zeitverzögerung bei anderen Projekten (z.B. Großbaustelle EXPO 2000 in Hannover)
  - Verfall verderblicher Güter vor dem Ausliefern

Selbst wenn man nur die streckenbezogenen Kosten hat (oder alle anderen Kosten entsprechend umlegen kann), bleibt die Frage, wie man diese bestimmt. Man kann beispielsweise über den Durchschnittsverbrauch argumentieren, daß man nur die Weglänge der Tour kennen muß, um die Transportkosten zu bestimmen. In diesem Fall werden zwei Alternativen eingesetzt: In der ersten werden einfach die euklidischen Abstände benutzt (evtl. mit Tabuflächen), die zweite bestimmt den kürzesten Weg auf einem vorgegebenen

Straßennetz. Hier müssen wiederum heuristisch verschiedene Straßenarten anders bewertet werden (ggf. über eine Durchschnittsgeschwindigkeit, die dann wieder von Staus etc. abhängt). Allein die Fahrtkosten zu schätzen ist also eine recht schwierige Aufgabe, so daß man bei groben Fehlern in den Daten auch nicht unbedingt die beste Alternative suchen muß, sondern vielmehr die Nähe zur besten bekannten unteren Schranke angeben kann. Dieser Ansatz, der auch die Qualität der Lösung einzuschätzen hilft, wird erst in Algorithmen der „dritten Generation“ (s.u.) gemacht.

1.6 Einschränkungen des Modells gegenüber der Realität

Das eingesetzte Grundmodell hängt maßgeblich von den aktiven Constraints und der Kostenstruktur ab. Es kommt also darauf an, die im Optimum aktiven Beschränkungen herauszufinden (zur Not geht das auch per trial and error) und die tatsächlichen Transportkosten genau zu erfassen.
Da man meist auf Schätzungen angewiesen ist, ist es legitim, für die Algorithmen später wahlweise ganzzahlige oder kontinuierliche Kosten anzunehmen. In der Kostenfunktions-Definition ist die Zielmenge deswegen nicht angegeben worden. Hier sollte man auch entscheiden, ob man manche Kostenarten nicht generell vernachlässigt und in einem konstanten Shift aller Streckenkosten versteckt.
Was dagegen ausgeschlossen ist, sind negative Kosten bzw. Erträge, womit man nicht mehr modellieren kann, daß sich der Transport zu besimmten Kunden ganz besonders lohnt.
Im Flugverkehr führen die Constraints und die Kostenstruktur zum *crew scheduling*, während bei der Bahn große, skalierte Echtzeitsysteme den Verkehr regeln.
???Linienbusse, Schulbusse???
Das Trennen von Depots (Orten, an denen beladen wird) und Kunden (Orten, an denen entladen wird) stellt noch keine Einschränkung dar. Erst wenn man die Zahl der Depots auf 1 setzt, verliert man eine Vielzahl von Anwendungen. Z.B. kann man nach Lieferung die wiedergewonnene Kapazität auf dem Rückweg für weitere Transporte (nicht vom Depot aus) nutzen.
Die Annahme, daß, wenn man wirklich nur eine Güterklasse zu Kunden transportieren will, nur ein Depot für diesen zuständig ist, ist zulässig, bestimmt dies doch die Standortwahl.
Die gravierendste Abweichung von der Realität ist das völlige Vernachlässigen der Zeitkomponente, die de facto eine sehr wesentliche ist.

Die Modellierung, die hier verwendet wird, eignet sich sehr gut, um das Vertreiben bestimmter Güter von einem zentralen Lager aus zu modellieren. Dasselbe Modell kann sowohl für die Belieferung von Kunden von regionalen Verteilungszentren(RVZ) aus verwendet werden als auch für die Belieferung dieser RVZs vom einzigen Produktionsstandort aus als auch für das Einsammeln von Rohstoffen oder Einzelteilen aus der Zulieferindustrie.
Um spezifische Details, die in der Praxis wichtig sind, darin zu berücksichtigen, muß man im Einzelfall doch wieder das Modell abändern.
Beispiele hierfür sind asymmetrische Kosten (z.B. durch Einbahnstraßen oder Berg-Tal-Konfigurationen), unterschiedliche Kapazitäten b_K der Fahrzeuge sowie mehrdimensionale oder geteilte Kapazitäten der Fahrzeuge (die bei unterschiedlichen zu transportierende Gütern wichtig werden, z.B. Gewicht-Volumen oder Diesel-Benzin-...).
Manche Fahrzeuge haben nicht nur ganz andere Kosten, sondern können manche Kunden gar nicht beliefern (z.B. Tieflader), während andere innerhalb eines Planungszeitraums mehrere Routen fahren können, dafür können sie nicht alle Güter transportieren.
Wenn eine Belieferung zu teuer ist, kann man einen Fremdauftrag zu gegebenen Kosten in die Optimierung miteinbeziehen.
Aus diversen Gründen kann es notwendig sein, bestimmte Kunden zuerst bzw. zuletzt zu beliefern. Eine Anordnung ist beim VRP auch nicht modelliert.
Z.B. zur Kundenbindung können periodische Lieferungen von Interesse sein (immer ähnliche Mengen vom gleichen Personal zu einer regelmäßigen Zeit geliefert). Beliefert man Zwischenlager, entsteht typischerweise die Klasse der *inventory routing problems*, bei denen man über die Zeit so liefern muß, daß ein Produkt nie „ausgeht“.
Ich will auch betonen, daß dieses Modell ein reines Kostenoptimierungsmodell ist und daß andere Bereiche des Managements unberücksichtigt bleiben (z.B. die Belastung der Fahrer). Die Lösungen, die das Modell liefert sind dann auch nur genau so zu verstehen; sie bieten also eine Entscheidungshilfe zum kosteneffizienten Einsatz der Ressourcen, jedoch keinen Entscheidungsersatz!

Zusammenfassung

Das Modellproblem ist sehr stark spezialisiert und bildet sozusagen die Schnittmenge (und leider nicht die Vereinigungsmenge) der praxisnahen Fälle. Im Einzelfall sollten unbedingt wesentliche Merkmale dem Modell hinzugefügt werden, um danach die passenden Algorithmen auszuwählen, die für diese Erweiterungen einfach anzupassen sind.
Ein Anhaltspunkt dafür ist für den Modellierer, sich die notwendigen Daten zu beschaffen:

- $K, \bigwedge_{1\leq k\leq K} b_k = b \rightarrow$ einfach
- $a_i \rightarrow$ einfach
- $c_E, t_E \rightarrow$ schwierig

Wie gut der eingesetzte Algorithmus sein muß, das entscheidet sich je nach dem, ob man ein Problem, mit regelmäßigen Aufträgen hat, das optimal gelöst werden soll, oder ob man ein Problem möglichst gut innerhalb einer gewissen Zeit lösen soll, da die Kunden bis zum Vortag bestellen dürfen.

1.7 Bedarf an rechnergestützter Optimierung von VRPs

[Fisher] schreibt darüber, daß etwa 15% des BSPs der USA dem Transport zugeschrieben werden, und leitet daraus ab, daß es einen entsprechend großen Bedarf an Optimierung im Rahmen der *operation research* und der *management science* gibt. Etwas weiter unten erklärt er, daß der durchschlagende Erfolg bislang vor allem deshalb ausgeblieben sei, da in den Anfängen der Entwicklung die Modelle sehr unzureichend waren und die Algorithmen keine garantierte Qualität gehabt hätten.
Daß für einen großen Teil dieser 15% eine algorithmische Bearbeitung allein durch die Erfassung der notwendigen Daten zu teuer sein könnte, beachtet er jedoch nicht...
Bezeichnend ist doch, daß bisher nur die Großindustrie (z.B. Ölfirmen) sowie wenige mittlere Unternehmen (Logistik und Transport) überhaupt Optimierung einsetzen. Im letzten Fall ist anzunehmen, daß Referenzen aufgebaut werden sollen. Die Optimierung solcher Probleme ist kaum entwickelt und noch recht teuer und bedarf immer der Anpassung an den Anwender. Solange diese nicht automatisch geschieht, wird es keine uniforme Lösung für alle geben und teure Mathematiker werden Hand anlegen müssen... Nichtsdestotrotz ist der Markt längst nicht ausgereizt.

Kapitel 2

Algorithmen und Heuristiken

2.1 Formale VRP-Formulierung

Definition 2.1.1 (VRP)
Gegeben sei die Kundenzahl $n \in \mathbb{N}$; Definiere die Kundenmenge

$$N := \{1, 2, ..., n\}, \tag{2.1}$$

das Depot

$$0, \tag{2.2}$$

die Menge aller Orte

$$N_0 := N \cup \{0\} \tag{2.3}$$

das Transportnetz als vollständiger, ungerichteter Graph

$$G = (V, E) \text{ mit } V := N_0 \text{ und } E := N_0 \times N_0 - \{ii | i \in N_0\} \tag{2.4}$$

Gegeben sei weiterhin eine positive Kostenfunktion c

$$c : E \longrightarrow \mathbb{R}_0^+ \tag{2.5}$$

oder wahlweise

$$c : E \longrightarrow \mathbb{N};$$

Gegeben seien die Fahrzeugzahl $K \in \mathbb{N}$ und eine einheitliche Kapazität b für jedes Fahrzeug. Gegeben seien die Aufträge bzw. Auftragsmengen a_N. (Z.B. a_3 ist die Größe des Auftrags von Kunde 3)
Eine Untermenge T von E heißt Route (Tour), wenn jeder besuchte Knoten genau bei zwei Kanten aus T auftaucht, also formal:

$$T \subset E \text{ heißt Route} \Leftrightarrow \begin{array}{l} \bigwedge_{ij \in T} |\{ij_1 \in T | j_1 \neq i\}| = 2 \\ \wedge \bigwedge_{ij \in E-T} |\{ij_1 \in T | j_1 \neq i\}| = 0 \end{array} \tag{2.6}$$

Bemerkung
Für jeden Knoten gilt also, da er entweder genau einmal auf der Tour liegt oder gar nicht. Die Angabe einer Tour besteht in der Aufzählung der benutzten Kanten, die Reihenfolge ergibt sich bis auf Umkehrung mit der Angabe eines Startknotens.
Schreibweise
Sei $i \in N_0$, T eine Tour;

$$i \in T :\Leftrightarrow \bigvee_{j \in N_0} ij \in T \tag{2.7}$$

(Sonst gelte $i \notin T$!)

Das VRP besteht darin, K Touren $\mathbb{T} := \{T_1, T_2, ..., T_K\}$ anzugeben, wobei folgende Bedingungen erfüllt sein müssen:

- *Jede Tour startet und endet am Depot.*

$$\bigwedge_{k=1}^{K} 0 \in T_k \tag{2.8}$$

- *Jeder Kunde gehört zu genau einer Tour.*

$$\bigwedge_{i \in N} \bigvee_{T \in \mathbb{T}}^{1} i \in T \tag{2.9}$$

- *Jedes Fahrzeug darf nur bis zur Kapazitätsgrenze beladen werden.*

$$\bigwedge_{k=1}^{K} \sum_{i \in T_k} a_i \leq b \tag{2.10}$$

- *Die Routen sollen zu minimalen Gesamtkosten gewählt werden.*

$$\bigwedge_{\mathbb{T}'} \mathbb{T}' \textit{erfüllt die obigen Voraussetzungen} \Rightarrow$$

$$\sum_{T \in \mathbb{T}} \sum_{e \in T} c_e \leq \sum_{T' \in \mathbb{T}'} \sum_{e' \in T'} c_{e'} \tag{2.11}$$

Definition 2.1.2 (VRP als nicht-lineares Zuweisungsproblem)
Um die Touren zu bestimmen, muß man sich bei jedem Fahrzeug für bzw. gegen bestimmte Kunden und bestimmte Wege entscheiden. Anders gesagt, man ordnet jedem Fahrzeug Kunden und Wege zu und bestimmt so die Kosten. Dies führt zur Formulierung des VRPs als nicht-lineares Zuweisungsproblem (nonlinear generalized assignment problem).
Führe Binärvariablen ein:

$$\bigwedge_{i \in N_0} \bigwedge_{k=1}^{K} y_{ik} := \begin{cases} 1, & \text{wenn Fahrzeug } k \text{ den Ort } i \text{ besucht} \\ 0, & \text{sonst} \end{cases} \tag{2.12}$$

$$\bigwedge_{k=1}^{K} y_k := (y_{0k}, ..., y_{nk}) \tag{2.13}$$

$f(y_k)$ bestimme die Kosten einer optimalen TSP-Lösung durch die Punkte in $N(y_k) := \{i | y_{ik} = 1\}$. Damit läßt sich das VRP definieren:

$$min \sum_k f(y_k) \tag{2.14}$$

„such that“

$$\bigwedge_{k=1}^{K} \sum_i a_i y_{ik} \leq b \tag{2.15}$$

$$\sum_k y_{0k} = K, \bigwedge_{i \in N} \sum_k y_{ik} = 1 \tag{2.16}$$

Um nun die Funktion $f(y_k)$ darzustellen, definiere:

$$\bigwedge_{ij \in E} \bigwedge_{k=1}^{K} x_{ijk} := \begin{cases} 1, & \text{wenn das Fahrzeug } k \text{ direkt von } i \text{ nach } j \text{ fährt} \\ 0, & \text{sonst} \end{cases} \tag{2.17}$$

Damit ergibt sich für alle k die Funktion $f(y_k)$ aus

$$f(y_k) = min \sum_{ij} c_{ij} x_{ijk} \tag{2.18}$$

„such that“

$$\bigwedge_{j \in N_0} \sum_i x_{ijk} = y_{jk} \tag{2.19}$$

$$\bigwedge_{i \in N_0} \sum_j x_{ijk} = y_{ik} \tag{2.20}$$

$$\bigwedge_{\substack{S \subset N(y_k) \\ 2 \leq |S| \leq n}} \sum_{ij \in S \times S} x_{ijk} \leq |S| - 1 \tag{2.21}$$

Diese Formulierung ist äquivalent zur ersten. (Ohne Beweis)

Anmerkung:
Es ist nicht sichergestellt, daß es überhaupt eine mögliche Tourenwahl für das VRP mit gegebenen Daten gibt. Wir sprechen daher schon von einer *Lösung*, wenn sie sämtliche Constraints bis auf die Optimalität einhält. Die gesuchte Lösung nennen wir entsprechend *optimale Lösung*.

2.2 Übersicht und Geschichtliches

Nachdem zum ersten Mal Computer zur Verfügung standen, versuchte man, möglichst einfache Daten und Programme laufen zu lassen. (Programme wurden früher oft gelocht, anstatt getippt ;-)) In diesem Umfeld (70er Jahre) entstanden Algorithmen, die darauf aus waren, überhaupt erst einmal Touren zu finden bzw. möglichst gute bzw. mit wenig Programmieraufwand oder mit wenig Rechenzeit. Diese Algorithmen faßt [Fisher] in der *Generation 1* zusammen.
Dann entwickelte man Probleme und Algorithmen, die man sehr gut mit dem Computer lösen konnte, z.B. LP oder IP. Die Algorithmen der *Generation 2* (80er Jahre) versuchten meist, das Modell den Fähigkeiten anzupassen und machten heuristische Ansätze, damit einfachere und damit effizient lösbare Modelle entstanden, in der Hoffnung, daß die dort optimale Lösung etwas mit der gesuchten des VRPs zu tun hat. In der Praxis sah das so aus, daß man mit Testdaten so lange experimentiert hat, bis die Algorithmen wirklich gute Lösungen brachten; doch im Einsatz zeigte sich, daß von Zeit zu Zeit – je nach Eingabedaten – dermaßen schlechte Ergebnisse ausgegeben wurden, daß man per Hand besser optimieren konnte. Am schlimmsten war allerdings, daß diese Effekte bei nur minimal anderen Eingangsdaten verschwanden!
(Es ist sehr leicht für die einzelnen Heuristiken (Gegen-)Beispiele zu konstruieren, die schlechte Ergebnisse oder Performanz verursachen.)
Diese Abhängigkeiten von Nuancen, die fehlende Robustheit der Algorithmen und das nur unzureichende Modell (damit konnte man auch funktionierende Lösungen nicht auf andere Anwender eins zu eins übertragen) haben dem Ruf der Optimierung sehr geschadet! In der Folgezeit wurde versucht, robuste Werkzeuge zu schaffen. Inzwischen stand für mehr Qualität auch Rechenzeit zur Verfügung. Eine wesentliche Idee hierbei ist die Berechnung

möglichst guter unterer Schranken, um das Verbesserungspotential einer gefundenen Lösung abschätzen zu können. Erstmals konnte man Optimalität außer durch Enumerierung nachweisen!
Auch diese Algorithmen setzen auf allgemein bekannte Grundideen, man kann sie allerdings nur über die Entstehungszeit klassifizieren. Die *Generation 3* (80er und 90er Jahre) der Algorithmen benutzen nach wie vor Heuristiken, um das $\mathcal{NP}$schwere VRP zu lösen.
Im Endeffekt sollte man Ideen aus allen Generationen verwenden, um gegebene Probleme zu lösen, denn jede Heuristik hat das Potential ganz schnell zum Optimum zu kommen bzw. versagt bei bestimmten Konstellationen.
Im folgenden wird versucht, die wesentlichen Ideen herauszuarbeiten – und zwar unabhängig von der geschichtlichen Entwicklung.
Das VRP hat mehrere Aspekte und damit verschiedene, mehr oder minder natürliche Lösungsansätze:

- *(n Kunden zu K Kunden)-Ansatz*
  Clarke, Wright (1964)
  Stichwort: „sequentielle Clarke und Wright-Methode"; n Touren iterativ möglichst kostensparend zu K Touren machen, indem man Kunden zusammenfaßt, solange die Kapazitätsgrenze nicht erreicht ist; keine garantierte Lösung, bisher bester routenbildender Algorithmus, viele Erweiterungen
  Gaskell (1967), Yellow (1970)
  Stichwort: mehrere Durchläufe mit 0-abstand-gewichten Einsparungen
  Golden, Magnanti, Nguyen (1977)
  Stichwort: technische Verbesserungen der Laufzeit
  Altinkemer, Gavish (1991)
  Stichwort: mehrere Zusammenfassungen pro Schritt per maximum matching problem zwischen bestehenden Teiltouren mit Gewicht der Einsparungen

- *gültige Touren entwerfen, beste Kombination auswählen*
  Balinski, Quandt (1964) ohne Implementierung für großskalierte Probleme
  Stichwort: set partitioning (Teilmenge auswählen aus J Alternativen) mit erfüllten Nebenbedingungen, Problem: gute Touren erzeugen, Dimension, Vorteile: komplexe Nebenbedingungen einfach modellierbar (z.B. Kostenfunktion); gut, wenn wenig zulässige Touren (z.B. bei Zeitfenstern)
  Callen, Jarvis, Ratliff (1981)
  Stichwort: interaktive Lösung über GUI, Anwender kann gültige Touren eingeben

Foster, Ryan (1976), Fleuren (1988)
Stichwort: truck scheduling, Zeitfenstervorgaben
Fisher, Rosenwein (1989)
Stichwort: ship scheduling
Fisher, Kedia (1990)
Stichwort: $n = 200, J = 10.000$
Agarwal, Mathur, Salkin (1989, Desrociers, Solomon (1990)
Stichwort: exakte Lösungen durch set partitioning und column generation; Zeitfenster-Constraints
Bramel, Simchi-Levi (1993)
Stichwort: Theoretische Betrachtungen zur Effektivität

- *Touren verbessern durch Kantentausch (lokale Optimierung wie TSP)*
  Lin (1965); Lin, Kernigham (1973)
  Stichwort: TSP, k-optimal; $k = 3$ ist schnell und gut
  Christofedes, Eilon (1969)
  Stichwort: K-TSP, K Kopien des Depots, TSP mit Dimension $n + K$, $k = 3$; geprüfte Zulässigkeit
  Russell (1977)
  Stichwort: Manchmal $k > 3$ also mehr als 3 Kanten getauscht Savelsberg (1985), Thompson (1988)

- *Lösung verbessern durch Nachbarschaftssuche*
  Stichwort: Suche iterativ von einer Startlösung gültige Nachbarlösungen, die meistens Verbesserungen liefern müssen; z.B. tabu search, simulated annealing
  Gendreau, Hertz, Laporte (1993)
  Stichwort: tabu search mit Verschieben einzelner Kunden
  Osman (1993)
  Stichwort: tabu search und simulated annealing mit Verschieben und Austausch von einzelnen Kunden
  Taillard (1993)
  Stichwort: tabu search wie Osman mit zusätzlichem Bestrafen mehrmaligen Tauschens
  Stichwort: parallele Variante implementiert tabu search in Kundengruppen
  s.u.

- *(Kunden zu Fahrzeug und dann TSP für jedes Fahrzeug)-Ansatz*
  Gillett, Miller (1974)
  Stichwort: „Sweep“, Zuordnung per Kreissektor nach Ordnung in Polarkoordinaten; neues Fahrzeug, sobald Kapazität erreicht

Christofides, Mingozzi, Toth (1979); Tyagi (1968)

- *(Lineare Approximation der Zuweisungsproblemformulierung)-Ansatz*
  Fisher, Jaikumar (1981)
  Stichwort: $f(y_k) \approx \sum_i d_{ik} y_{ik}$; i_K „Saat“-Kunden, an denen die K Touren „wachsen“ (euklidisch durch Kunden-Sektoren, Mittelung und euklidische Punkte als „seeds“; sonst immer „entfernteste“ (maxmin)); $d_{ik} = c0i + c_{ii_k} - c_{0i_k}$, sequentielles Aneinanderfügen je nach Nähe zum Depot oder je nach Unterschied bester und zweitbester Fahrzeuge
  s.u.

- *(Expertensystem)-Ansatz*
  Stichwort: Constraints, Beispiel Zuweisung vs. set partitioning, durchschnittlich $1,5$ Kunden pro Fahrzeug vs. harte Kapazitätsgrenze, ...
  Desrochers, Lenstra, Savelsbergh (1990)
  Stichwort: Nach Eingabe des Problems Parameterbestimmung und-Wahl des geeigneten Algorithmus durch Metasystem
  Kadaba, Nygard, Juell (1991)
  Stichwort: Neuronale Netze, Genetische Algorithmen zur „seed“-Wahl
  Potvin, Lapalme, Rousseau (1989)
  Stichwort: Interaktives Zuordnungsverfahren

- *(„polyhedral combinatorics“)-Ansatz*
  Chvatal (1973), Grötschel (1980), Grötschel, Padberg (1979), Padberg, Hong (1980), Grötschel, Pulleyblank (1986);
  (Buchempfehlung von [Fisher]: [1985])
  Stichwort: symmetrisches n-TSP
  Laporte, Nobert, Desrochers (1985)
  Stichwort: Anpassung an VRP, Implementierung mit
  a posteriori-Generierung verletzter Constraints
  ($x_{ij} \sim$ wie oft von i nach j)

$$V(S) := \lceil \sum_{i \in S} \frac{a_i}{b} \rceil \text{ Mindestanzahl der Fahrzeuge}$$

$$min \sum_{i<j} c_{ij} x_{ij}$$

$$\sum_{j<i} x_{ij} + \sum_{j>i} x_{ji} = 2$$

$$\bigwedge_{\substack{S \subset N \\ 2 \leq |S| \leq n-1}} \quad \sum_{\substack{ij \in S \times S \\ i<j}} x_{ij} \leq |S| - V(S)$$

$$\sum_j x_{0j} = 2K$$

$$x_{ij} \text{ binärwertig bzw. für i=0 auch } \in \{0,1,2\}$$

- *verallgemeinerte Kammgleichungen und b-matching relaxation*
  Chvatal (1973), Grötschel, Padberg (1979), Laporte, Nobert (1987), Laporte, Bourjolly (1984) Stichwort: Hinzunahme weiterer Ungleichungen zu „polyhedral combinatorics", Dualisierung zu b-matching problem; $W_i \subset N$; Kamm: W_0, k Zähne: W_k, die paarweise genau ein Element gemeinsam haben; damit:

  $$\sum_{l=0}^{k} \sum_{i,j \in W_l} x_{ij} \leq \sum_{l=0}^{k} |W_l| - \lceil \frac{1}{2} \sum_{l=1}^{k} V(W_l) + V(W_l - W_0) + V(W_l \cap W_0) \rceil$$

  Cornuejols, Harche (1989)
  Stichwort: Modellproblem #1 (n=50, K=5) optimal gelöst.
  Campos, Corberan, Mota (1991), Araque, Kudva, Morin, Pekny (1994)
  Stichwort: Spezialisierung auf $\bigwedge_{i \in N} a_i = 1$
  Miller (1993)
  Stichwort: Dualisierung der Bedingung, daß die Summe der ein- und ausgehenden Kanten gleich 2 sein muß, zu b-Matching Problem
  Miller, Pekny (1994)
  Stichwort: Algorithmus, der #1 ganzzahlig optimal löst.

- *Kürzeste-Wege-Relaxation zur Bestimmung unterer Grenzen*
  Christofides, Mingozzi, Toth (1981)
  Stichwort: Neuer Graph mit Knoten $(i,q), q \leq b$ Verbindungen von (i,q) zu (j,r) mit Kosten c_{ij} (wenn $q < r$); $F(i,q)$:= Länge eines kürzesten Wegs von $(0,0)$ nach $(i,q) + c_{i0}$ ist untere Grenze, um q Einheiten an i zu liefern. Tour gültig bis auf Mehrfachlieferungen; Berechnung von

  $$v(i,q,k) := \begin{array}{l} \text{Minimale Kosten von k Touren mit Fracht q} \\ \text{zu verschiedenen Endkunden aus}\{1,...,i\} \end{array}$$

  durch

  $$v(i,q,k) = min\{v(i-1,q,k); min_{q'}\{v(i-1,q-q',k-1) + F(i,q')\}\}$$

  und damit ist eine untere Grenze für das Liefern von q Einheiten mit i als letztem Kunden. Eine untere Grenze für das VRP ist dann durch

$v(n, \sum_i a_i, K)$ gegeben; diese ist zugleich eine bis auf Mehrfachlieferungen an einen Kunden zulässige Lösung! Die Grenze kann noch verbessert werden, indem man diese Verletzungen bestraft, indem man solche Lösungen höher bewertet („Lagrangesche Strafen“).
Kolen, Rinnooy Kan, Trienekens (1987)
Stichwort: Verallgemeinerung mit Zeitfenstern
Christofides (1985)
Stichwort: „state space relaxation“, $n = 53$, $K = 8$, Fahrzeit- und lockere Zeitfensterbedingungen

- *K-Baum-Relaxation zur Bestimmung unterer Grenzen*
  Fisher (1990)
  Stichwort: K-Bäume
  s.u.

- *andere Ansätze*
  Haimovich, Rinnooy Kan (1985), Altinkemer, Gavish (1987), Bramel, Simchi-Levi (1992), Christofides, Mingozzi, Toth (1981), Lucena (1986)

2.3 Suchtechniken

Ausgehend von einer zulässigen Startlösung S konstruiert man sich eine gewisse Nachbarschaft $N(S)$, in der man sich eine möglichst bessere Alternative S' sucht. Dann iteriert man das Verfahren mit S'.
Die Suchtechniken unterscheiden sich je nachdem, wie diese Nachbarschaft konstruiert wird, und je nachdem, welche Alternative daraus (zu welcher Wahrscheinlichkeit) gewählt wird. Traditionellerweise erlaubt man nur eine strikte Verbesserung; tabu search und simulated annealing erlauben aber auch eine kontrollierbare Verschlechterung, um aus lokalen Minima herauszukommen (heuristisch).
Üblicherweise bedeutet Nachbarschaft das Austauschen von n_1 Kunden einer gegebenen Tour mit n_2 Kunden einer anderen Tour, wobei n_1, n_2 nach oben beschränkt sind. Meistens sind diese Schranken gleich 1, also $n_1 \leq 1$, $n_2 \leq 1$. In Worten bedeutet das, daß einzelne Kunden zwischen Touren wechseln oder ausgetauscht werden dürfen.

2.3.1 tabu search

Die Tabu-Suche sucht sich in jeder Iteration die beste zulässige Alternative in der gesamten Nachbarschaft. Innerhalb der nächsten t Iterationen darf diese Wahl nicht rückgängig gemacht werden (daher der Name). (Beispielsweise darf ein Kunde c_1, der von Tour T_j nach T_i gewechselt hat, innerhalb der nächsten t Iterationsschritte nicht zurückwechseln, wohl aber mit einem anderen Kunden c_2 aus T_j ausgetauscht werden!) Das Abbruchkriterium der Tabu-Suche kann eine vorgegebene Iterationszahl sein.

2.3.2 simulated annealing

Simulated annealing durchsucht $N(S)$ in einer vorgegebenen Ordnung, bestimmt bei jeder Alternative S' die Kostendifferenz zu S als Δ. S' wird sofort als neue Startlösung gewählt, wenn $\Delta \leq 0$, aber auch im Falle $\Delta > 0$, wenn zusätzlich $e^{-\Delta/T} \geq \theta$ gilt, wobei θ ein einheitlicher Zufallsparameter (zwischen 0 und 1) ist und T als „Temperatur“ ein Kontrollparameter ist, über dessen „Kühlung“ man die Wahl einer verschlechternden Alternative immer unwahrscheinlicher machen kann. Wenn man keine Verbesserung mehr findet, kann man T wieder etwas erhöhen, um aus lokalen Minima wieder herauszulaufen. Eine mögliche Abbruchbedingung könnte hier sein, innerhalb einer gewissen Iterationszahl keine Verbeserungen mehr zu erhalten.

2.4 general assignment problem = VRP

Dieser Abschnitt geht auf Details der linearen Approximation der Zuweisungsproblemformulierung ein. Das Problem ist, daß man exponentiell viele Möglichkeiten hat, die Kunden auf die Fahrzeuge aufzuteilen; dann fallen einige Möglichkeiten wegen der Kapazitätsbedingung heraus, danach spielt diese keine Rolle mehr. Unter diesen exponentiell vielen gültigen Touren müssen nun K Touren ausgewählt werden, die Kosten sind dabei zu optimieren. Für jede Teilmenge von Kunden, die auf einer gültigen Tour liegen, soll jetzt also ein TSP gelöst werden.
Das ist das Vorgehen im Fall der nichtlinearen Zuweisung: Ordne den Fahrzeugen die zu besuchenden Kunden zu, optimiere die Tour durch Zuweisung der zu benutzenden Wege:

$$min \sum_k f(y_k),$$

$$f(y_k) = min \sum_{ij} c_{ij} x_{ijk}$$

Diese 2-Phasen-Methode wird linear approximiert, indem man f durch $f(y_k) \approx \sum_i d_{ik} y_{ik}$ ersetzt. Die d_{ik} sollen also angeben, um wieviel sich eine

Tour verlängert, wenn man den Ort i zu einer bestehenden Tour hinzufügt.

$$min \sum_{\substack{k=1,...,K\\ i=0,...,n\\ j=0,...,n}} c_{ij}x_{ijk} \approx min \sum_{k,i} d_{ik}y_{ik}$$

Am Anfang muß man also bestimmte Kunden $i_1, ..., i_K$ auswählen, die die K Touren initialisieren. Dies geschieht im euklidischen Fall über folgendes Verfahren:
Man halbiere die Winkel zwischen polar benachbarten Kunden und gewichte die entstehenden Sektoren mit den entsprechenden a_is. Dann bestimmt man am Depot (K) zusammenhängende Winkelintervalle, die je mit $\overline{b} = \sum_i a_i/K$ gewichtet sind. In jedem Intervall wird nun ein Radius bestimmt, so daß $3/4\ \overline{b}$ innerhalb des entsprechenden Kreissektors liegen. Der das Intervall halbierende Strahl geschnitten mit dieser Kreislinie bestimmt dann den Samen („seed") (der nicht mit einem Kunden zusammenfallen muß!).
Im nicht-euklidischen Fall versucht man, die Samen möglichst weit voneinander entfernt zu setzen: Sei s_1 der vom Depot entfernteste Kunde. Bestimme nun rekursiv bis s_K:

$$max_i min\{c_{i0}, min_{j=1,...,k}\ c_{is_j}\}$$

[Fisher] beschreibt noch eine Heuristik von Bramel, Simchi-Levi(1992), die sowohl die Wahl der „seed"-Kunden als auch die Zuordnung der anderen Kunden zu den „seeds" liefert. Sie haben dabei eine Nebenbedingung Lagrange-relaxiert (alles auf eine Seite gebracht und diese Null auf die Kostenfunktion addiert).
Dieses Vorgehen läßt neben diversen Heuristiken, wie man die d_{ij} schätzen kann (z.B. für seeds: $2c_{0j}$ sonst $c_{0i} + c_{ij} - c_{0j}$ (!)) eine Vielfalt von Nebenbedingungen zu.
Beispiele mit Zeitfenstern: Nygard, Greemberg, Bolkan, Swenson (1988), Koskosidis, Powell, Solomon (1989)
mit aufgeteilten Lieferungen: Brenninger-Goethe (1989)

2.5 K-Baum-Relaxation

Spannende Bäume lassen sich verallgemeinern zu (spannenden) K-Bäumen: Sei ein Graph mit $n + 1$ Knoten gegeben, dann ist ein K-Baum eine Untermenge von Kanten, die genau $n + K$ Kanten enthält und den Graphen aufspannt.

Das VRP ist modelliert als ein kostenminimaler, knotengradbeschränkter K-Baum mit Nebenbedingungen. Diese werden Lagrange-relaxiert, also dualisiert.
Neben den üblichen Annahmen zu G, N, N_0 beschreibt die Binärvariable x_{ij}, ob die ungerichtete Kante ij in der Lösung vorkommt. Eine Lösung kann man also durch einen Vektor

$$x := (x_{01}, x_{02}, ..., x_{0n}, x_{12}, x_{n-2,n-1}, x_{n-1,n})$$

darstellen. Außerdem definiere die Menge

$$X := \{x \mid x \text{ binärwertig, definiert einen K-Baum mit } \sum_{i=1}^{n} x_{0i} = 2K\}$$

Damit läßt sich das VRP darstellen als

$$Z^* = min_{x \in X} \sum_{ij \in E} c_{ij} x_{ij} \tag{2.22}$$

s. t.

$$\bigwedge_{i \in N} \sum_{\substack{j \in N_0 \\ j \neq i}} x_{ij} = 2 \tag{2.23}$$

$$\bigwedge_{\substack{S \subset N \\ |S| \geq 2}} \sum_{i \in S} \sum_{j \in N_0 - S} x_{ij} \geq 2r(S), r(S) := \lceil \frac{\sum_{i \in S} a_i}{b} \rceil \tag{2.24}$$

In dieser Formulierung sind Touren mit nur einem Kunden nicht erlaubt! Dies ist bei einer relativ zu den Bestellmengen großen Kapazität b kein Problem, denn ein Kunde j kann nicht alleine auf einer Tour liegen, wenn

$$\left(\sum_{i=1}^{n} a_i \right) - a_j > (K-1)b$$

gilt, da sonst für die anderen Kunden nicht mehr genügend (Transport-) Kapazität übrig ist.

$$\Leftrightarrow \bigwedge_{j} \frac{a_j}{b} < \sum_{i=1}^{n} \frac{a_i}{b} - (K-1)$$

(Man bemerke, daß im entgegengesetzten Fall die Zahl möglicher Touren sehr klein ist — das wäre dann ein Fall für set partitioning!)

Nach Lagrange-Relaxierung und Hinzufügen verletzter Constraints zur Kostenfunktion, sobald sie verletzt sind, wird eine Strategie angewandt, die die Kosten für die Verletzungen hochtreibt (durch „Lagrange-Multipliers“).
Bei gegebener Untermenge $S \subset N$ kann man durch $a(S)$ und die inzidenten Kanten zu anderen Kunden $T \subset N - S$ und deren $a(T)$ die Anzahl der notwendigen Fahrzeuge für diese Untermenge nach unten abschätzen und erhält eine weitere Ungleichung:

$$\sum_{i \in S} \sum_{j \in \overline{S}} x_{ij} \geq 2 \left\lceil \frac{1}{b}(a(S) + \sum_{i,j \in S \times \overline{S}} a_j x_{ij}) \right\rceil$$

Aus dieser nichtlinearen Ungleichung kann man andere ableiten, bei der die rechte Seite linear wird. Z.B. mit

$$\bigwedge_{s \subset N} S' := \{j \in \overline{S} | j \geq 1 \text{ and } a_j > br(S) - a(S)\}$$

und

$$e_j := \begin{cases} 0, & j \in S \\ 0, & j \in S' \text{ and } |S'| \leq 2 \\ \frac{r(S)}{r(S)+1}, & j \in S' \text{ and } |S'| > 2 \\ 1, & j \in \overline{S} - S' \end{cases}$$

ergibt sich die Fahrzeug-Kapazitäts-Beschränkung

$$\bigwedge_{\substack{S \subset N \\ |S| \geq 2}} \sum_{j=0}^{n} e_j \sum_{i \in S} x_{ij} \geq 2\ r(S)$$

Der K-Baum-Ansatz liefert nicht nur untere Schranken, er kann auch zur Lösung des Problems eingesetzt werden: In einem Branch-And-Bound-Algorithmus kann man diejenigen Zweige abschneiden, deren lower bound schlechter ist als eine schon vorhandene zulässige Lösung. Die Frage ist nur, wonach man verzweigt.
Die klassische Variante, sich für oder gegen eine bestimmte Kante zu entscheiden (*edge branching*) und dann weiterzurechnen, ist hier unangebracht, da Verzweigungen teuer sind und die Entscheidung an der Verzweigung zu wenig Informationen enthält.
customer branching, also das Auswählen eines Kunden pro Verzweigung, korrespondiert mit dem Anfügen an eine bestehende Tour, die erst dann fertig ist, wenn sie am Depot anfängt und endet; eine Tour darf nur geschlossen

werden, wenn die Restkapazität ausreicht, um die übrigen Kunden zu beliefern. Auch diese Variante läßt den Verzweigungsbaum sehr groß werden.
Eine Verzweigungsmethode von Christofides, Mingozzi, Toth (1981) verwaltet Knoten, die Sequenzen von Kunden beinhalten, die abgeschnitten werden, sobald es eine Sequenz derselben Kunden mit geringeren Kosten gibt (mit gleichen Anfangs- und Endkunden). Implementiert wurde das mit der 3-opt-Regel von Lin, Kernighan (1973). Sie nannten dieses Verfahren „Dominanztest“.
Außerdem kann man verzweigen je nach Anzahl der Fahrzeuge, die eine Untermenge von Kunden bedienen. Hieraus lassen sich weitere scharfe Ungleichungen und effiziente Dominanztests entwickeln.
Die traditionelle Variante muß bei Clustern entsprechend versagen, da bei m Kunden im Cluster $m!$ Verästelungen entstehen, bei der Methode von Christofides jedoch diese Teilbäume sofort abgeschnitten werden.
[Fisher] stellt noch weitere Verzweigungsmöglichkeiten vor und entwickelt noch weitere lineare und nichtlineare Ungleichungen bzw. Funktionen. Dieser Aufwand ist dadurch zu rechtfertigen, daß eine Verzweigung immer teuer im Sinne von Speicherplatz und Rechenzeit ist und man versuchen sollte, Arbeit, die man in einem Teilbaum erledigt hat, in andere miteinzubeziehen (indem man möglichst dort rechnet, wo es später weitergeht). Branch-And-Bound bzw. Branch-And-Cut-Algorithmen sind also nicht unbedingt die straightforward-Lösung, sondern kommen vor allem dann zum Einsatz, wenn man die Qualität einer vorhandenen Lösung abschätzen will oder eine sehr gute noch verbessern will; denn einerseits findet man mit etwas Aufwand gute untere Grenzen für das VRP, und andererseits wird der Verzweigungsbaum bei bekannten guten Lösungen von Anfang an klein-gestutzt. Manche Heuristiken finden heute gute Lösungen selbst, um das Divergieren des Baumes zu vermeiden, sie lösen Probleme optimal für $n \approx 50$ bis 100.

Kapitel 3

Vergleich der besten Lösungen, Optima

[Fisher] führt neben den einzelnen Algorithmen und Heuristiken die Kosten der besten gefundenen Lösungen einer Hand voll Modellprobleme an, die sämtlich planar sind. Die ersten drei Probleme sind künstlich generierte, die auf einer statistischen (zufälligen) Verteilung beruhen (von Christofides, Eilon (1969)). Problem 4 entsteht durch die Hinzunahme der Kunden von 1 zum Problem 3 und Einsparung eines Fahrzeugs. Kommen dann noch die ersten 49 Kunden von Problem 2 hinzu, entsteht Problem 5.
Problem 6 stammt von Christofides, Mingozzi, Toth (1979) und soll die in der Wirklichkeit vorkommende Clusterung der Kunden nachempfinden.
Das sind die klassischen 6 Probleme, an denen neue Methoden immer wieder getestet werden. Das ist auch der Grund, warum zu den anderen Problemen nur Daten von Bramel, Simchi-Levi vorliegen:
Fisher hat tatsächliche Daten an Bramel und Simchi-Levi geschickt, um Vergleichswerte zu seinen Ergebnissen zu erhalten. Man erkennt leicht, daß die zufälligen Verteilungen für alle Algorithmen den *worst case* bedeuten. Sobald eine Clusterung auftaucht (die in der Wirklichkeit gegeben ist), kann man die Probleme optimal lösen.
Die Datensätze zu den Problemen 7, 8 und 9 beschreiben den Transport von Industriegasen, die in Zylindern verpackt sind (euklidisch). Hier konnte ohne Branching Optimalität nachgewiesen werden.
Das zehnte und das zwölfte Problem stammen aus dem Lebensmittelgroßhandel, das elfte ist ein Problem von Exxon, bei dem Reifen, Batterien und Zubehör an die eigenen Tankstellen geliefert werden.
Die Algorithmen unterscheiden sich gravierend in den CPU-Kosten, die nur sehr eingeschränkte Aussagen liefern können, da sie meist auf verschiedenen Rechnern implementiert wurden, und selbst wenn diese gleich waren, wurden

sie mit evtl. verschiedenen Motiven implementiert (unterstelle, daß man an der eigenen Idee mehr arbeitet und daß man fremde Ideen nicht unbedingt optimal implementiert). Bezeichnend ist auch, daß Taillards Tabu-Suche, die immer die beste Lösung unter den aufgeführten liefert, keine CPU-Kosten angibt, zu denen man sich diese Ergebnisse besorgen kann.
In der Tabelle auf der nachfolgenden Seite liefert die parallele Variante der Clarke & Wright-Methode den Wert 1351 (*), außerdem hat Osmans Tabu-Suche nur $K = 16$ Fahrzeuge eingesetzt, woraus eine Auslastung der Kapazität von $0,999$ resultiert (**). Letzteres erklärt auch den Ausreißer im Verhältnis $\frac{lower}{upper}$: Die K-Baum-Relaxation bringt eben in jedem Falle eine drastische Einsparung der Kosten im Vergleich zum Lösen des VRPs mit den harten Constraints. Es ist also die Frage, ob man hier mit einem anderen Verfahren bessere untere Schranken bekommen kann oder ob unsere bisherigen Ansätze einfach versagen bei strengen Kapazitätsgrenzen und $n = 199$.

Problem			Generation 1				Generation 2			Generation 3							
										Tabu search			Simul. ann.	K-Baum			
n	K	$\frac{\sum a_i}{Kb}$	Clarke and Wright	Gillett, Miller „Sweep“	Christo-fides, Eilon	Russell	Fisher, Jaiku-mar	Bramel, Simchi-Levi	Foster, Ryan	Gendr., Hertz, Laporte	Osman	Taillard	Osman	optimal oder besser	untere Grenze	lower — upper	Subgra-dient Iter.
50	5	0,97	585	532	556	524	524	524,6	523	524,61	524	*524,61*	528	-	507,09	0,97	3000
75	10	0,97	900	874	876	854	857	848,2	864	835,32	844	*835,26*	538	-	755,5	0,90	3000
100	8	0,93	886	851	863	833	833	832,9	825	*826,14*	835	*826,14*	829	-	785,86	0,95	3000
150	12	0,94	1204	1079	-	-	1014**	1088,6	-	1031,07	1044	*1028,42*	1058	-	932,68	0,91	3000
199	17	0,95	1540*	1389	-	-	1420	1461,2	-	1311,35	*1334*	1298,79	1378	-	1096,72	0,82	3000
100	10	0,91	831	937	-	-	824	826,1	-	*819,56*	819	*819,56*	826	819,56	817,77	0,998	2000
25	3	0,92						*3104*						3070	3070	1,0	769
25	3	0,92						*5830*						5829	5829	1,0	2444
25	3	0,92						*5032*						4961	4961	1,0	320
25	3	0,92						*723,54*						723,54	720,76	0,996	2000
25	3	0,92						*244,92*						241,97	237,76	0,98	2000
25	3	0,92						*1216,66*						1163,60	1133,73	0,97	2000

Literaturverzeichnis

[Fisher] Ball, Magnanti, Monma, Nemhauser *Network Routing*, 8. Band der *Handbooks in operaton research and management science*, Elsevier, 1995
erstes Kapitel (S.1-33) von Marshall Fisher *Vehicle Routing*

[1985] Lawler, Lenstra, Rinnooy Kan, Shmoys *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization* , Wiley, Chichester, 1985