

ABSCHLUSSBERICHT

Teilvorhaben TRANSPORTOPTIMIERUNG innerhalb des
BMBF-Verbundprojektes HYBRID

Februar 1997

Leiter des Teilvorhabens:
Dr. Heinz Mühlenbein

Projektmitarbeiter:
Dipl.-Math. Ulrich Bartling,
Dipl.-Inform. Andreas Reinholz

GMD Forschungszentrum Informationstechnik
D-53754 Sankt Augustin

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen **01 IB 405 A2** gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt bei den Autoren.

Inhaltsverzeichnis

1	Zusammenfassung	1
1.1	Wesentliche Ergebnisse	1
1.2	Vergleich des Standes des Vorhabens mit der ursprünglichen Arbeits-, Zeit- und Kostenplanung	1
1.3	F&E Ergebnisse	2
2	Transportoptimierung für ein Containertransportsystem CTS	3
2.1	Einleitung	3
2.2	Problemstellung	3
2.2.1	Begriffe	4
2.2.2	Optimierungsziele	4
2.2.3	Randbedingungen und Vorgaben	5
2.2.4	Parameter	7
2.2.5	Kostenfunktion	7
2.2.6	Optimierungsaufgabe	8
2.3	Entwurf eines Genetischen Algorithmus	8
2.3.1	Genetische Kodierung	8
2.3.2	Lokale Suche	9
2.4	Genetische Suche	9
2.4.1	Mutation	9
2.4.2	Crossover	10
2.5	Inkrementelles Optimieren	11
2.6	Ergebnisse	11
2.7	Transfer der Ergebnisse	13
2.8	Danksagung	13
3	Transportoptimierung für ein zentrales Auslieferungsdepot (ZAD) mit hybriden Evolutionären Algorithmen	15
3.1	Einleitung	15
3.2	Wissenschaftliche Problemstellung	16
3.3	Problemlösung	16
3.3.1	Vorgehensweise	17

3.3.2	Optimierungsziele	18
3.3.3	Datenstruktur	20
3.3.4	Initialisierung	20
3.3.5	Visualisierung	21
3.3.6	Operatoren und Verbesserungsverfahren in der Problemhierarchie	21
3.4	Problemstellung der Industrievertretung BUB	24
3.4.1	Erstellen der Eingabedaten für die Optimierung	25
3.4.2	Umsetzung in die Praxis	27
3.5	Leistungsbewertung anhand von Benchmarks	27
3.5.1	Tagesplanung	27
3.5.2	Periodenplanung	32
3.6	Schlußbemerkung	34

1 Zusammenfassung

1.1 Wesentliche Ergebnisse

Das Teilvorhaben *Transportoptimierung* besteht aus den beiden getrennt behandelten Arbeitspaketen

- Transportoptimierung für ein Container-Transport-System (CTS)
- Transportoptimierung für ein zentrales Auslieferungsdepot (ZAD)

Beide Arbeitspakete wurden durch Implementierung von problemspezifischen Heuristiken im Rahmen des Konzepts genetischer Algorithmen erfolgreich zum Abschluß gebracht.

An der CTS-Implementierung haben mehrere Paketdienste/Logistikunternehmen (Bahntrans, Dachser, DPD, German Parcel) ein echtes Interesse bekundet. Drei dieser Firmen haben zur Evaluation des Optimierungsprogramms entsprechende Benchmarks zur Verfügung gestellt, die vierte plant einen solchen Test. Das Programm hat sich bereits als praxistauglich erwiesen. Weitere Verhandlungen zwecks Nutzung der erarbeiteten Software (nach zusätzlichen, kundenspezifischen Anpassungen) werden z. Zt. geführt.

Sehr hilfreich für eine praxisrelevante Erstellung des ZAD-Programms ist der Kontakt zur Industrievertretung Bub im Siegerland gewesen, die einen werkseigenen Speditionsbetrieb unterhält, um Duscholux-Sanitärprodukte an ca. 400 Großkunden im Raum NRW, Hessen und Rheinland-Pfalz auszuliefern. Neben mehrmaligen Kundenbesuchen im Planungshorizont (z.B. 5 Tage-Woche) berücksichtigt die ZAD-Software insbesondere Restriktionen für Tourlänge, Fahrzeugkapazitäten sowie Entladezeiten. Ohne Schwierigkeiten, ja sogar teilweise mit Verbesserungen bisheriger Bestwerte, absolvierte das Programm auch die komplexeren Tests unter den Benchmarks aus der Operations Research Literatur.

1.2 Vergleich des Standes des Vorhabens mit der ursprünglichen Arbeits-, Zeit- und Kostenplanung

Die Arbeiten konnten wie geplant durchgeführt werden. Um eine erfolgreiche Umsetzung in die Praxis zu gewährleisten wurden die Ergebnisse des Projektes mit Mitteln der GMD einer Reihe von interessierten Unternehmen ausführlich präsentiert. Die Vertragsverhandlungen mit TNT Netzwerklogistik über einen Einsatz des entwickelten Programms sind gescheitert, da die POST AG den Vertrag mit TNT gekündigt hat.

1.3 F&E Ergebnisse

Das in diesem Vorhaben entwickelte Programmsystem CTS ist eine der größten bekannten Anwendungen genetischer Algorithmen. Die Praxistauglichkeit evolutionärer Algorithmen konnte damit exemplarisch nachgewiesen werden.

2 Transportoptimierung für ein Containertransportsystem CTS

Ulrich Bartling

2.1 Einleitung

Zur (suboptimalen) Lösung klassischer Transport- und Tourenprobleme wie etwa beim Vehicle Scheduling und in speziellen Fällen des Crew Scheduling stellt das Operations Research eine Reihe von allgemein akzeptierten Algorithmen bereit. Klassiker der Tourenprobleme ist dabei das Auffinden einer kostenminimalen Rundreise des Travelling Salesman. Ebenso klassisch ist der kostenminimale Transport eines Produkts, das an r Orten (Quellen) hergestellt und an s anderen Orten (Senken) nachgefragt wird. Hierzu gehört auch das Vehicle Scheduling für ein einzelnes oder mehrere Auslieferungslager. Die Aufgabenstellungen im Logistikbereich werden jedoch zunehmend komplexer. Neben der Planung von Transporten ist auch die Organisation von Warenverteilungs- und Beschaffungssystemen gefragt (just-in-time Bedarfsdeckung). Die Liberalisierung des Güterverkehrs innerhalb der EU und nicht zuletzt auch ökologische Argumente verstärken den Bedarf nach kostenoptimalen Einsatzplänen für einen gegebenen Fuhrpark.

Bislang ist der Einsatz von Personal-Computern in Speditionen auf Standardanwendungen wie Finanzbuchhaltung oder Lagerverwaltung beschränkt. Moderne Optimierungsverfahren aus dem Bereich der evolutionären Algorithmen eröffnen den immer leistungsfähigeren PC jedoch neue Einsatzmöglichkeiten bei der Transportoptimierung.

2.2 Problemstellung

Die Deutsche Post hat bis Anfang 1995 im Rahmen ihres neuen Frachtpostkonzepts 33 Frachtpostzentren (FZ) errichtet. Hinzukommen etwa 10 Großkunden, deren Auslieferungslager wie Frachtzentren behandelt werden. Die Paketzustellung findet in drei Arbeitsschritten statt: Im Vorlauf werden die Pakete im jeweiligen Einzugsgebiet eines FZ gesammelt und entsprechend der Empfängeradresse in Standard-Container sortiert. Im Hauptlauf werden die Container zu ihrem Ziel-FZ transportiert. Von dort findet dann im Nachlauf die Zustellung zum Empfänger statt.

Gegenstand der Optimierung ist hier der Hauptlauf. Für den Transport der Container stehen in den einzelnen Frachtzentren sowohl Motorwagen wie auch Lastzüge bereit; eine Obergrenze ist zunächst nicht zu beachten. Auf ausgewählten Strecken ist ein Transport per Bahn vorgeschrieben.

In einzelnen FZ werden voraussichtlich nicht immer so viele Container eintreffen wie

notwendig sind, um den Bedarf des nächsten Tages an leeren Containern zu decken. Vielmehr wird es FZ mit Überhang (Quellen) bzw. Fehlbestand (Senken) an leeren Containern geben. Es sind zwei eng miteinander verknüpfte Optimierungsaufgaben zu lösen. Zum einen sind geeignete Verbindungen zwischen Quellen und Senken zu finden, um leere Container bereitzustellen; zum anderen ist für die Gesamtheit der Transporte ein kostengünstiger Fahrplan zu erstellen.

Für eine detailliertere Beschreibung ist die Einführung einer Reihe von (Fach-) Begriffen unerlässlich.

2.2.1 Begriffe

- Sofern der Unterschied keine Rolle spielt, kann *Fahrzeug* sowohl einen einzelnen Motorwagen als auch einen Lastzug bezeichnen.
- Mit einer *Sendung* wird ein zuzustellender, beladener Container bezeichnet.
- Die Menge aller Sendungen wird auch als das Frachtaufkommen bezeichnet.
- *Ausgleichsfahrten* (AF) führen leere Container zu den Frachtzentren zurück, in denen ein Fehlbestand an Containern herrscht.
- Sind Start und Endpunkt einer Fahrt identisch, liegt eine *Rundfahrt* vor.

2.2.2 Optimierungsziele

Bei der Erstellung eines Fahrplans sind eine Reihe von Optimierungszielen zu gewährleisten. Ihnen vorangestellt sind zwei Randbedingungen, deren überragende Bedeutung es rechtfertigt, sie bei den mit unbedingter Priorität zu verfolgenden Zielen aufzuführen.

(O1) Zustellung sicherstellen

Keine zuzustellende Sendung darf liegenbleiben, weder im Ausgangs-Fz noch unterwegs in Zwischenstationen gleich welcher Art.

(O2) Anlieferung bis 5.00 Uhr

Die Anlieferung bis 5.00 Uhr hat absolute Priorität. Nur dann kann die Auslieferung an den Endkunden am auf die Einlieferung folgenden Tag ($E + 1$) gewährleistet werden.

Alle weiteren kostenrelevanten Randbedingungen stehen hinter den Bedingungen O1 und O2 zurück.

(O3) Anlieferung

Die Entladekapazität eines Fz darf nicht überschritten werden. Maximal 30 beladene Container können gleichzeitig entladen werden.

- (O4) Leerbehälter–Rückführung
Sie soll mit möglichst wenig zusätzlichem Aufwand erledigt werden. Die leeren Container sind bis spätestens 15 Uhr am Zielort anzuliefern.
- (O5) Fahrzeuge
Es sollen bevorzugt Lastzüge eingesetzt werden, einzelne Motorwagen sind eher die Ausnahme. Die Gesamtzahl der eingesetzten Fahrzeuge ist möglichst klein zu halten.
- (O6) Vorhaltung leerer Container
Jeder leere Container in der Vorhaltung kostet Standmiete, deren Betrag davon abhängt, ob der Container auf dem Gelände des Depots abgestellt werden kann oder extern Lagerkapazität angemietet werden muß. Zu viele Container in der Vorhaltung verursachen Kosten, zu wenig Container in einem Depot kann zur Folge haben, daß Ausgleichsfahrten nicht dann starten können, wenn es sinnvoll wäre, sondern dann starten müssen, wenn sie möglich sind.
Die Anzahl der benötigten Container ist möglichst gering zu halten.
- (O7) Rundfahrten
Wann immer möglich, kehren die Fahrzeuge am Ende einer Fahrt wieder an ihren Ausgangspunkt zurück, da die Fahrer wieder nach Hause kommen sollten.

2.2.3 Randbedingungen und Vorgaben

Bei der Erstellung eines Fahrplans sind die folgende Randbedingungen einzuhalten:

- (R1) Die Planungsperiode hat eine Länge von 26.5 Stunden. Sie beginnt um 15.00 Uhr am zu planenden Tag und reicht bis 17.30 Uhr des nächsten Tages.
- (R2) Container
Alle Container sind gleich, es sind keine speziellen Standorte, Wege oder Typen zu berücksichtigen.
- (R3) Frachtzentren
Desgleichen können auch alle Frachtzentren gleichbehandelt werden, sowohl was die Be- und Entladekapazität betrifft als auch die Dienstzeiten des Personals. Beladene wie leere Container können bis 15.00 Uhr angeliefert werden.
- (R4) Lenkzeitbeschränkung
Die gesetzlichen Vorschriften über die Beschränkung von Lenkzeiten und Einhaltung der Pausen sind bei der Erstellung des Fahrplans zu beachten (4h30 Fahrt, 45min Pause, 4h30 Fahrt, max. Lenkzeit 9h).
- (R5) Bahnverkehr
Auf vorgegebenen Strecken wird der Transport der Container mit der Bahn durchgeführt.

- (R6) Punkt-zu-Punkt Verkehr
 Ein Container soll von einem bestimmten Frachtzentrum zu einem anderen transportiert werden. Wenn es die zeitgerechte Anlieferung nicht gefährdet, aber kostengünstiger (weniger Fahrzeuge) durchführbar ist, kann ein Container auch über einen Umweg zugestellt werden.
- (R7) Fahrten ohne Container
 sind unerwünscht. Solange das globale Optimierungsziel möglichst niedriger Kosten gewährleistet bleibt, werden sie aber vom Optimierungsprozeß nicht generell ausgeschlossen.
- (R8) Gleichmäßige Anlieferung
 Der Planungszeitraum wird in Zeitintervalle der Länge $10min$ zerlegt. Die Ankunftszeiten der Fahrzeuge sind so zu verteilen, daß nicht mehr als 30 beladene Container in einem $10min$ -Intervall zu entladen sind.
- (R9) Arbeitszeit eines Fahrers
 Die *Schichtzeit* eines Fahrers beginnt mit dem Aufladen des ersten Containers im Start-Frachtzentrum; eine evtl. Anfahrt wird nicht berücksichtigt. Die *Lenkzeit* beginnt mit der Abfahrt im Start-Frachtzentrum, weitere Aufenthalte in Frachtzentren oder Treffpunkten bei Begegnungsverkehr sind eine Unterbrechung der Lenkzeit und werden nur auf die Schichtzeit angerechnet. Die Schichtzeit endet nach dem Abladen des letzten Containers. Sollte das Fahrzeug ohne Container abfahren (ankommen), so beginnt (endet) die Schichtzeit mit der Lenkzeit. Die max. Schichtzeit beträgt 12 Stunden.
- (R10) Entladen der Container
 Drei Stunden nach der Anlieferung ist ein beladener Container entladen und steht für Ausgleichsfahrten zur Verfügung.
 Ein Container wird nicht mehrfach innerhalb der Planungsperiode be- und entladen.
- (R11) Ladezeiten für Fahrzeuge
 Für das Auf- und Abladen werden jeweils 15 min pro Container benötigt. Ein Lastzug wird sequentiell ab-/beladen, seine Verweildauer in einem Fz beträgt somit maximal eine Stunde.
- (R12) Um das Frachtaufkommen transportieren zu können, gilt die Annahme, daß zu Beginn der Planungsperiode eine hinreichende Anzahl von Leercontainern in den einzelnen Frachtzentren bereit steht. Gibt es darüber hinaus noch weitere leere Wechselbehälter in den Fz, so sind diese Anzahl explizit dem Optimierungsprogramm bekannt zu geben. Siehe auch unter "Vorrat an Leerbehältern" im nächsten Abschnitt.

2.2.4 Parameter

Aus technischen Gründen werden weitere Parameter benötigt, beispielsweise eine Beschreibung der Ausgangssituation, wie sie zu Beginn der Planungsperiode vorliegt.

- Frachtaufkommen
- Distanz-Tabelle (in *km* wie in Minuten Fahrzeit)
- Vorrat an Leerbehältern pro Frachtzentrum zu Beginn der Planungsperiode
- Gefordeter Endbestand an Leerbehältern am Ende der Planungsperiode
- Kosten für den Einsatz bestimmter Ressourcen (Container, Motorwagen etc.)
- Preistabelle(n) zur Abrechnung der Fahrleistung
- Kosten für unterbliebene Rückfahrten
- Bahnrelationen
Zwischen welchen Fz eine Bahnverbindung besteht und zu welchen Zeiten die Züge verkehren, wird vorgegeben. Die entsprechenden Daten können vom Wochentag abhängen.

2.2.5 Kostenfunktion

Um verschiedene Fahrpläne bewerten zu können, wird eine Kostenfunktion benötigt. Details der Kostenfunktion sind anwendungsspezifisch, im Kern bewertet sie folgendes:

1. Fahrleistung

Zur Zeit das alles entscheidende Kriterium. Die Kosten pro km werden nach einer Preistabelle abgerechnet. Für jedes Fahrzeug wird die Preistabelle seines Heimat-Frachtzentrums zu Grunde gelegt.

2. Unterbliebene Rückfahrten

Kehrt ein Fahrzeug nicht zu seinem Heimat-Frachtzentrum zurück, obschon dies ohne Überschreitung der Lenkzeit und ohne Einsatz eines 2. Fahrers möglich wäre, werden Starfkosten pro km Entfernung zum Heimat-Frachtzentrum berechnet. Diese Strafkosten sind unabhängig vom Heimat-Frachtzentrum.

3. Einsatz von Betriebsmitteln

Hierzu zählen Motorwagem, Lastzüge und Wechselbehälter. Entsprechende Beträge für die Kosten sind vorzugeben.

4. Strafkosten für die Überschreitung der Entladekapazität eines Frachtzentrums.

2.2.6 Optimierungsaufgabe

Zur Beschreibung eines Transports sind Angaben über Start- und Ziel-FZ notwendig sowie über den frühesten Zeitpunkt, zu dem mit dem Transport begonnen werden kann (früheste Abzugszeit). Hinzukommt eine Status-Information, die angibt, ob ein *beladener* Container (Sendung), ein *leerer* Container oder *kein* Container transportiert werden soll. Die Zusammenfassung dieser Daten zu einem 4-Tupel $\{von, nach, abzug, status\}$ bildet einen *Transportauftrag* (TA). Für ein vorgegebenes Frachtaufkommen läßt sich sofort die zugehörige Menge S der entsprechenden TA angeben. Die Menge A der TA der Ausgleichsfahrten kann man auf vielfältige Weise erhalten; um eine Anfangsbelegung zu bekommen, verbinde man die jeweils nächstgelegenen Quellen und Senken miteinander. Fahrten ohne Container werden durch Transportaufträge mit Status *kein* Container beschrieben, sie bilden die Menge K . Die Mengen S , A und K sind paarweise disjunkt.

Die Abfolge der von einem Fahrzeug ausgeführten Transportaufträge wird *Lkw-Umlauf* oder nur kurz *Umlauf* genannt. Eine Menge von Umläufen bildet einen Fahrplan für $S \cup A \cup K$, wenn jeder Transportauftrag aus $S \cup A \cup K$ in genau einem Umlauf liegt.

Die Optimierungsaufgabe kann jetzt so formuliert werden: Bestimme eine Menge S von Transportaufträgen für ein vorgegebenes Frachtaufkommen und eine Menge A von Transportaufträgen für die notwendigen Ausgleichsfahrten. Fasse diese TA mit denen einer geeignet gewählten Menge K gemäss den o.g. Randbedingungen so zu Umläufen zusammen, daß ein kostenoptimaler Fahrplan entsteht.

2.3 Entwurf eines Genetischen Algorithmus

2.3.1 Genetische Kodierung

Aus der im vorigen Abschnitt hergeleiteten Formulierung der Optimierungsaufgabe ergibt sich in naheliegender Weise eine genetische Kodierung. Ein Chromosom repräsentiert einen kompletten Fahrplan, die Gene auf dem Chromosom entsprechen Lkw-Umläufen. Die Anzahl der Gene ist variabel, schließlich soll eine möglichst kleine Flotte von Fahrzeugen eingesetzt werden.

Eine Anfangsbelegung des Chromosoms eines Individuums erhält man, indem man jedem Transportauftrag aus $S \cup A$ genau einem Umlauf zuordnet. K ist anfangs leer. Der so entstandene Fahrplan ist gültig, aber extrem schlecht. Hinzukommt, daß für alle Individuen eine identische Anfangsbelegung generiert wird. Um die nötige genetische Vielfalt zu erzeugen, werden die im folgenden Abschnitt vorgestellten lokalen Suchverfahren eingesetzt.

2.3.2 Lokale Suche

Mit Blick auf einzelne Optimierungsziele lassen sich leicht einfache Strategien formulieren, die einen oder auch mehrere Lkw-Umläufe so modifizieren, daß weniger Kilometer gefahren werden oder weniger Fahrzeuge eingesetzt werden müssen. Eine detaillierte Beschreibung dieser Verfahren würde den Rahmen dieses Berichts sprengen, sodaß hier nur eine kurze Klassifikation gegeben wird. So lassen sich möglicherweise die Umläufe zweier Motorwagen zu einem Umlauf für einen Lastzug zusammenfassen (Parallelisieren) oder zwei Umläufe können hintereinander ausgeführt werden (Konkateneren). Unter bestimmten Voraussetzungen kann Begegnungsverkehr geplant werden oder es werden Rundfahrten erzeugt.

Diese Verfahren werden grundsätzlich so eingesetzt, daß sie keine Verschlechterung der Kosten eines von ihnen bearbeiteten Fahrplans zulassen. Kann keine lokale Verbesserung erzielt werden, wird die lokale Suche abgebrochen.

Wie oben erwähnt, werden diese lokalen Suchverfahren bei der Initialisierung eingesetzt, dort jedoch nur diejenigen, die eine Lastzugbildung forcieren. Messungen haben gezeigt, daß diese Beschränkung sich positiv auf die genetische Vielfalt während der nachfolgenden genetischen Suche auswirkt.

2.4 Genetische Suche

2.4.1 Mutation

Im Gegensatz zu den Heuristiken, die nur dann eine Aktion ausführen, wenn eine lokale Verbesserung garantiert ist, variiert die Mutation Lkw-Umläufe auch dann, wenn eine Verbesserung nur vermutet werden kann. Beispielsweise wird aus dem Umlauf eines Motorwagens der eines Lastzuges gemacht, indem die fehlenden TA aus anderen Umläufen herausgebrochen werden, ohne die Folgen für diese Umläufe zu bewerten. Die in den zerstörten Umläufen noch verbleibenden Transportaufträge werden - ähnlich wie bei der Initialisierung - zu einer Menge von Umläufen mit nur einem TA zusammengefaßt. Auf diese Umläufe werden nun die Heuristiken ange-
setzt, um sie neu zusammenzufassen.

Die Bedeutung dieser und ähnlicher Operationen liegt darin, neue Strukturen zu schaffen, aber auch darin, alte aufzubrechen und somit lokale Optima wieder zu verlassen.

Die Mutation kann aber auch auf die Menge der TA durchgreifen, um eine neue Ausgleichsfahrt zu erzeugen. Zwei komplementäre Strategien sind hierfür realisiert, Aufspalten bzw. Vereinigen. Beim Aufspalten wird eine Ausgleichsfahrt vom Frachtzentrum A zum Frachtzentrum B durch zwei neue Ausgleichsfahrten ersetzt. Dazu wird zufällig ein Frachtzentrum X ausgewählt und eine AF von A nach X erzeugt und eine zweite von X nach B. Als komplementäre Aktion werden zwei AF, die den

Umweg über ein Frachtzentrum Y machen, wieder zu einer Fahrt von A nach B zusammengefasst. Diese letzte Maßnahme reduziert die zu fahrende Strecke, aber auch die erste Aktion kann sehr sinnvoll sein. Liegen A und B weit von einander entfernt, kann der Transport eines leeren Containers schneller erfolgen, wenn einer von A nach X und gleichzeitig einer von X nach B transportiert wird. Oder aber es müßte für den Transport des leeren Containers eigens ein Fahrzeug eingesetzt werden, aber auf mindestens einer Teilstrecke reicht es aus, einen Anhänger zusätzlich einzusetzen.

Daneben werden noch weitere Variationen erzeugt; in allen Fällen sind Eingriffe in bestehende Umläufe erforderlich, die mit Hilfe der Heuristiken nachbehandelt werden.

2.4.2 Crossover

Für das Design des Crossover-Operators war der entscheidende Aspekt die Tatsache, daß das CTS zerlegbar ist. Wird das Gebiet, in dem die Transporte stattfinden, in eine nördliche und eine südliche Hälfte geteilt, so ist in jeder Hälfte ein eigenständiges CTS zu lösen; hinzu kommt der grenzüberschreitende Verkehr. Man kann sich leicht vorstellen, daß Transporte innerhalb Schleswig-Holsteins sicherlich unabhängig von denen sind, die in Bayern stattfinden. Für einen Crossover-Operator ergibt sich damit die ideale Situation, die von zwei Individuen repräsentierten Lösungen in sinnvoller Weise so zerlegen zu können, daß sich die Teile zu einer neuen Lösung zusammensetzen lassen.

Das Crossover zwischen zwei Eltern E_1 und E_2 sei hier kurz skizziert:

1. Wähle ein FZ und eine Anzahl von benachbarten Frachtzentren zufällig aus. Sie bestimmen eine Region R.
2. Übernehme alle Umläufe, die nur Frachtzentren in R besuchen, aus E_1 in das zu erzeugende Individuum.
3. Übernehme alle Umläufe, die nur Frachtzentren besuchen, die nicht in R liegen, aus E_2 in das zu erzeugende Individuum.
4. Umläufe, die Frachtzentren in R besuchen wie auch Frachtzentren, die nicht in R liegen, werden genau dann übernommen, wenn sie in beiden Individuen liegen.
5. Alle anderen Umläufe werden so zerlegt, daß sie nur noch einen Transportauftrag enthalten. Diese neuen Umläufe werden kopiert wie in den beiden vorherigen Schritten beschrieben. Im neuen Individuum werden diese Umläufe mit den lokalen Suchverfahren nachbehandelt.

Messungen haben gezeigt, daß dieses Crossover ein überaus wirkungsvoller Operator ist.

2.5 Inkrementelles Optimieren

Trotz aller Bemühungen, die Fahrplanerstellung zu beschleunigen, ist die Rechenzeit immer noch zu hoch, um an einen Einsatz in der täglichen Praxis zu denken. Wenn man den besten in allen bisherigen Läufen gefundenen Fahrplan als Optimum bezeichnet, so erreicht man auf einer SUN Ultra Sparc einen Plan, der 2% vom Optimum entfernt ist, nach etwa 30 Minuten, die 1% Schranke wird nach etwas mehr als 90 Minuten unterschritten. Für den Praktiker ist damit ein akzeptabler Fahrplan erreicht, für den jedoch viel zu lange gerechnet wird. Um eine Abweichung von weniger als 1 Promille zu erzielen, sind mehr als 12 Stunden Rechenzeit notwendig. Die vergleichsweise lange Phase mit kleinen Fortschritten liegt darin begründet, daß die km-Preise in den einzelnen Frachtpostzentren sich geringfügig unterscheiden. Falls es gelingt, den Startpunkt einer Tour in ein Zentrum mit geringeren Kosten zu verlegen - was aufgrund der Randbedingungen schwierig ist -, kann eine Verbesserung von etwa 1-10 DM erzielt werden (bei Gesamtkosten in einer Größenordnung von 1 Mill. DM). Werden alle km-Preise identisch vorgegeben, konvergiert die Suche deutlich schneller.

Ausgehend von der Überlegung, daß sich die Mengen der Transportaufträge von aufeinander folgenden Tagen nicht grundlegend unterscheiden, ist das Feature des inkrementellen Optimierens entwickelt worden.

Gegeben sei eine Menge von Fahrplänen. Dann läßt sich auf der Basis der aktuellen Transportauftragsdatei ermitteln, welche Aufträge im Fahrplan enthalten, aber nicht mehr gefahren werden sollen - diese werden dann gelöscht - und welche Aufträge gefahren werden sollen, aber nicht im Fahrplan enthalten sind - diese werden hinzugefügt. Technisch gesehen entsteht eine Situation, wie sie auch am Ende der genetischen Operatoren vorliegt. Neu zu entwickeln war der Fahrplanabgleich. Dabei ist sehr vorsichtig vorzugehen, um den vorgegebenen Fahrplan nicht allzu heftig zu "stören". Obschon die Struktur der Fahrpläne immer noch stärker zerstört wird, als unbedingt erforderlich, sind jedoch schon erfreuliche Ergebnisse festzuhalten. So dauert es nur 10 Minuten, um die 2%-Schranke zu unterschreiten und 40 Minuten bis zur 1%-Schranke.

2.6 Ergebnisse

Messungen

Um aussagekräftige Ergebnisse zu erzielen, haben wir aus Daten, wie sie in der täglichen Praxis angefallen sind, eine Referenz-Datensatz abgeleitet. Darin sollen ca. 2700 beladene und ca. 500 leere Container transportiert werden. Als relevante Größe für die Qualität einer Lösung (Fahrplan) geben wir die Anzahl der gefahrenen Kilometer an. Alle Messungen wurden auf einem PC mit Pentium 166 Mhz Prozessor durchgeführt.

Population	Durchschnittswerte für		
	Km	Fkt-Auswert.	Laufzeit
32	517692.8	7.644E+05	16.0h
64	515981.3	6.663E+05	10.5h
96	515251.9	7.190E+05	14.0h
128	513940.5	9.519E+05	19.5h

Tabelle 1: Größe der Population vs. Lösungsqualität (km)

Distanz zum vorl. Optimum	Durchschnittswerte für		
	Fkt-Auswert.	Std. Abw.	Laufzeit
5.0%	1.700E+04	1.241E+03	00:19
4.0%	2.670E+04	2.314E+03	00:29
3.0%	4.924E+04	4.984E+03	00:50
2.0%	1.419E+05	2.421E+04	02:30
1.5%	2.938E+05	6.756E+04	05:30
1.0%	6.238E+05	2.675E+05	12:00

Tabelle 2: Konvergenz mit PopSize 128

Um überhaupt eine Vorstellung davon zu bekommen, in welcher Größenordnung sich das Optimum befinden könnte, haben wir wiederholt Optimierungsläufe gestartet, die mit aus vorherigen Läufen generierten Fahrplänen als Startlösung aufgesetzt wurden (Sonderfall des inkrementellen Optimierens). Wir sind schließlich zu einem Fahrplan mit 511589 km Fahrleistung gekommen.

Es zeigt sich, daß große Populationen bessere Ergebnisse produzieren als kleine. Dies deutet stark daraufhin, daß der Crossover-Operator die Fitness-Landschaft gut exploriert. Entsprechende Ergebnisse werden in Tabelle 1 vorgestellt.

Für eine grosse Population mit 128 Individuen sind in Tabelle 2 Durchschnittswerte für die Laufzeit und die Anzahl der Auswertungen der Fitness-Funktion angegeben, die die zu erwartende Konvergenzgeschwindigkeit aufzeigen. Als bisheriges Optimum werden die *Kosten*, ausgedrückt in DM, des bisher besten Fahrplans angenommen; die Distanz zum bisherigen Optimum bezieht sich ebenfalls auf die Kosten der generierten Pläne.

Abschließend läßt sich feststellen, daß das Ziel, auf einem PC eine Tourenplanung für das Container Transport System erstellen zu können, klar erreicht wurde. Der Referenz-Datensatz ist deutlich größer als das durchschnittlich zu erwartende Frachtaufkommen der Deutschen Post AG. Die privaten Paketdienste (s.u.) liegen gemessen am Paketaufkommen hinter der Deutschen Post AG als Marktführer; sie werden also

weniger Container zu transportieren haben. Kleinere Problemgrößen erlauben eine kleinere Population, die Laufzeiten werden sich deutlich verringern. Wenn man den Tests Glauben schenken darf, ist die Verarbeitungsgeschwindigkeit eines Pentium Pro Prozessors doppelt so hoch wie die des bei uns eingesetzten Pentium 166 Mhz, so daß sich die oben genannten Laufzeiten glatt halbieren würden.

2.7 Transfer der Ergebnisse

Unter dem Namen Cargo/CT ist ein Optimierungsprogramm einschließlich einer graphischen Benutzeroberfläche entstanden, das auf großes Interesse gestoßen ist. Wir haben eine Präsentation auf der Tagung der Kommission für Logistik und Informatik des Bundesverbands Spedition und Lagerei (BSL) in Hilden durchgeführt. In der Folge ist es dann zu Kontakten zu

- Bahntrans GmbH, Niederlassung Offenburg
- DACHSER GmbH&Co, Karlruhe
- DPD Deutscher Paket Dienst GmbH, Aschaffenburg
- German Parcel Paket Logistik GmbH, Neuenstein

gekommen. Alle diese Firmen haben Daten für einen Benchmark zugesagt, teilweise sogar schon zugeschickt.

Einen ähnlichen Test haben wir bereits vor einem Jahr mit TNT netlog durchgeführt. Schon die damalige Version von Cargo/CT hat 10% bis 15% günstigere Fahrpläne erzeugt als die Planungssoftware von TNT.

2.8 Danksagung

Die vorgenannten Arbeiten wären nicht möglich gewesen ohne die gute Unterstützung durch TNT netlog, Troisdorf.

Literatur

- [Bodin 1981] Lawrence Bodin, Bruce Golden. *Classification in Vehicle Routing and Scheduling*. NETWORKS 11, 97-108 (1981)
- [Desrochers 1990] M. Desrochers, J.K. Lenstra, M.W.P. Savelsbergh. *A classification scheme for vehicle routing and scheduling problems*. European Journal of Operational Research 46, 322-332 (1990)

- [Mühlenbein 1992] H. Mühlenbein. *How Genetic Algorithms Really Work: Mutation and Hill-climbing* Parallel Problem Solving from Nature (PPSN II) Männer and Manderick (eds.), 15-26, North-Holland (1992)
- [Mühlenbein 1993] H. Mühlenbein, D. Schlierkamp-Voosen. *Predictive Models for the Breeder Genetic Algorithm*. Evolutionary Computation, 1(1): 25-49 (1993)
- [Mühlenbein 1994] H. Mühlenbein, D. Schlierkamp-Voosen. *The science of breeding and its application to the breeder genetic algorithm BGA* Evolutionary Computation, 1(4): 335-360 (1994)

3 Transportoptimierung für ein zentrales Auslieferungsdepot (ZAD) mit hybriden Evolutionären Algorithmen

Andreas Reinholz

3.1 Einleitung

Dem ZAD-Problem (Zentrales Auslieferungs Depot) liegt eine konkrete Aufgabenstellung der Industrievertretung BUB zugrunde, die in ihrem Vertriebsgebiet den Werksverkehr für DUSCHOLUX, einem führenden Hersteller aus dem Sanitärbereich, abwickelt. Die Industrievertretung BUB beliefert von ihrem Zentrallager in Siegen Großhändler in ganz NRW mit Waren unterschiedlicher Größe. Innerhalb einer Planungsperiode von einer Arbeitswoche sind diese Kunden unter bestimmten Nebenbedingungen zu bedienen. Jeder Kunde darf höchstens einmal pro Tag besucht werden. Vorgegebene wöchentliche Besuchshäufigkeiten, die vom Umsatz des Kunden abhängen, sind einzuhalten. Die Tage, an denen ein Kunde bedient wird, müssen ein für den Kunden erlaubtes Besuchsmuster bilden. Die Touren, in denen die Kunden angefahren werden, beginnen und enden am Depot. Die Fahrzeuge werden am vorabend beladen und können daher nur einmal pro Tag eingesetzt werden. Die Dauer jeder Tour ist durch die maximale Arbeitszeit der Fahrer beschränkt und das Frachtvolumen pro Tour durch die Kapazität der Fahrzeuge. Ziel der Optimierung ist ein Rahmenfahrplan, der alle Nebenbedingungen erfüllt und die Summe der Operationszeiten aller Fahrzeuge minimiert.

Damit entspricht diese Aufgabenstellung im wesentlichen dem in der einschlägigen Literatur als Period Vehicle Routing Problem (PVRP) bezeichneten Optimierungsproblem [CB84] mit der Periodenlänge 5. Das PVRP ist eine periodische Erweiterung des Capacitated Vehicle Routing Problem (CVRP).

Das Ziel dieses Teilprojektes ist die Entwicklung eines leistungsfähigen hybriden Evolutionären Algorithmus, mit dem einerseits *Real World* Problemstellungen und andererseits entsprechende Benchmarks des *Operation Research* gelöst werden können.

Dazu wurde ein Ansatz gewählt, bei dem Problemspezifika in die Komponenten Kodierung, Genetische Operatoren und *Neighborhood Search* eingewoben werden. Die Tragfähigkeit dieses Ansatzes wurde anhand der wichtigsten Benchmarkprobleme aus der einschlägigen Literatur für das CVRP und das PVRP überprüft. Bei allen bearbeiteten Probleminstanzen der beiden NP-vollständigen Probleme konnten die besten bisher veröffentlichten Lösungen erreicht bzw. verbessert werden.

Für die Lösung von Problemstellungen aus der Praxis ist das Optimierungsverfahren in ein interaktives Visualisierungstool integriert worden, mit dem der Disponent durch die Vorgabe bestimmter Teiltouren auf Sondersituationen eingehen kann.

3.2 Wissenschaftliche Problemstellung

Eine konkrete Instanz des ZAD-Problems für N Kunden wird definiert über

- die Menge der Kundenindizes K_i mit $0 \leq i \leq N$
- die Zeit-Entfernungsmatrix $(D_{i,j})$ mit $0 \leq i, j \leq N$
- den Fracht-Anforderungsvektor (A_i) mit $0 \leq i \leq N - 1$
- den Entladezeitenvektor (E_i) mit $0 \leq i \leq N - 1$
- eine Menge der gültigen Besuchsmuster B_i mit $0 \leq i \leq N - 1$
- die Kapazität der Fahrzeuge (Kapazitätsbeschränkung) K und die Sicherheitsmarge K^+
- die Anzahl der Fahrzeuge L .
- die Arbeitszeit der Fahrer (Tourlängenbeschränkung) Z und die Sicherheitsmarge Z^+
- die Periodenlänge P .

Das zentrale Auslieferungsdepot wird in der Distanzmatrix als spezieller Kunde mit dem Index N berücksichtigt.

3.3 Problemlösung

Eine vielversprechende Methode, komplexe Probleme zu bearbeiten, ist die Unterteilung in eine Hierarchie von weniger komplexen Teilproblemen.

Bei Lösungen des ZAD-Problems wird die Menge der Kunden in Tagesklassen unterteilt, die jeweils durch einen Tagesplan zu bedienen sind. Dadurch wird ein Partitionierungsproblem definiert. Für die Bewertung einer jeden Tagesklasse ist jeweils ein bester Tagesplan heranzuziehen.

Das Bestimmen eines optimalen Tagesplanes ist wiederum ein Partitionierungsproblem. Hierbei werden die Kunden des Tagesplans in Tourenklassen unterteilt, die jeweils von einem LKW in einer Tour bedient werden. Dabei darf die gesamte Fracht der Kunden in jeder Tourenklasse nicht die jeweilige Ladekapazität des LKW übersteigen. Jede Tourenklasse muß der Frachtbedingung genügen.

Die Bewertung einer jeden Tourenklasse erfolgt dann über die jeweils beste Rundtour durch alle Kunden dieser Klasse. Damit entspricht dieses Problem dem klassischen *Travelling Salesman Problem* (TSP).

Die Unterteilung des ZAD-Problems in diese Hierarchie von Teilproblemen ermöglicht die Entwicklung von Operatoren, die zielgerichtet einzelne Aspekte der Problemhierarchie in den Vordergrund stellen. Die Bewertung der durch einen Operator generierten Lösung erfolgt jedoch für das Gesamtproblem.

Daher wird die Lösung des ZAD-Problems erarbeitet über die Entwicklung von Lösungsansätzen für das

1. *Travelling Salesman Problem*
2. *Capacitated Vehicle Routing Problem* (Tagesplanung)
3. *Period Vehicle Routing Problem* (Wochen- bzw. Periodenplanung)

3.3.1 Vorgehensweise

Bei der Entwicklung eines effizienten Genetischen Algorithmus für das ZAD-Problem wird folgendermaßen vorgegangen:

1. Definition einer Fitness- bzw. Kosten-Funktion, welche die Ziele der Optimierung widerspiegelt.
2. Entwicklung einer Datenstruktur, welche eine effiziente Implementierung von genetischen Operatoren und *Neighborhood Search* Verfahren ermöglicht, die auf verschiedenen Ebenen der Problemhierarchie arbeiten.
3. Erzeugen von guten Startlösungen durch klassische Konstruktionsverfahren.
4. Implementierung eines Entwicklungs- und Visualisierungstools, mit dem eine konkrete Suche animiert und die gefundenen Problemlösungen graphisch dargestellt werden können.
5. Definition von einstelligen plausiblen Operatoren, die zielgerichtet auf den verschiedenen Ebenen der Problemhierarchie arbeiten und möglichst gültige Teillösungen wieder in gültige Teillösungen transformieren. Diese können sowohl als Mutationsoperatoren als auch in *Neighborhood Search* Verfahren verwendet werden.

Dabei wird für jede Ebene der Problemhierarchie wie folgt vorgegangen:

- (a) Visualisierung der Wirkungsweise der Operatoren.
- (b) Testen der Einsatzfähigkeit der Operatoren
 - i. in Standardverbesserungsverfahren wie z.B. *steepest* oder *next ascent*,
 - ii. als Mutationsoperatoren in einfachen Evolutionsverfahren wie z.B. dem (1+1) Verfahren

- (c) Bestimmung einer Vorauswahl leistungsfähiger Operatoren.
- (d) Erarbeiten von deterministischen und zufallsgesteuerten *Neighborhood Search* Verfahren für alle drei Ebenen der Problemhierarchie.

Hierbei ist besonderer Wert auf eine geeignete Integration der Operatoren in die *Neighborhood Search* Verfahren zu legen, so daß bei einem Suchschritt anstelle einer kompletten Neuberechnung der Kostenfunktion eine effizientere Berechnung über Differenzterme verwenden. Diese Vorgehensweise ermöglicht es, die Komplexität des Gesamtverfahrens erheblich zu verringern.

- (e) Integration der Mutationsoperatoren und der *Neighborhood Search* Verfahren in einen hybriden Evolutionären Algorithmus für die jeweilige Ebene der Problemhierarchie.
6. Kombination Integration der so gewonnenen Optimierungsverfahren zu einem hybriden Evolutionären Algorithmus für das ZAD-Problem.

3.3.2 Optimierungsziele

Das Ziel der Optimierung ist ein Rahmenfahrplan, der alle Nebenbedingungen erfüllt und die Summe der Operationszeiten aller Fahrzeuge minimiert. Dazu ist eine geeignete, fiktive Fitness- bzw. Kostenfunktion zu designen, die die Ziele der Optimierung widerspiegelt. Eine wichtige Bedeutung dabei hat die Handhabung der Nebenbedingungen des Optimierungsproblems.

Nebenbedingungen

Die Nebenbedingungen werden bei dem vorgestellten Lösungsansatz auf zwei Arten berücksichtigt. Falls möglich wird versucht, die Einhaltung der Nebenbedingungen durch eine geeignete Wahl der Operatoren und der Codierung des Suchraumes zu gewährleisten. Für Nebenbedingungen, bei denen dies nicht möglich ist, werden additive Strafterme eingeführt, die in der fiktiven Kostenfunktion berücksichtigt werden.

Bei Evolutionären Algorithmen wird die Suche durch die Fitness- bzw. Kostenfunktion gesteuert. Daher haben diese Strafterme einen entscheidenden Einfluß auf den Optimierungsprozeß. Eine ungeeignete Wahl kann zur Folge haben, daß

- der Suchprozeß in ungültigen Bereichen des Suchraumes stagniert,
- gültige Bereiche des Suchraumes zu stark voneinander isoliert werden,
- optimale bzw. sehr gute Lösungen bezüglich der fiktiven Kostenfunktion Nebenbedingungen verletzen.

Ein idealer Fall liegt vor, wenn die Strafterme so gewählt werden können, daß die aufgeführten Folgen vermieden werden und den ungültigen Bereichen des Suchraumes die Kosten so zugeordnet werden können, daß sich ein iterativer Suchprozess über Verbesserungen von ungültigen zu gültigen Lösungen hin entwickeln kann.

Bei dem ZAD-Problem sind folgende Nebenbedingungen zu beachten:

1. Jeder Kunde wird höchstens einmal pro Tag auf einer Rundtour besucht, die am Depot beginnt und endet.
2. Jeder Fahrzeug der Flotte wird höchstens einmal pro Tag eingesetzt.
3. Alle Kunden werden gemäß der erlaubten Besuchsmustern angefahren.
4. Für alle Kunden werden die vorgeschriebenen Besuchshäufigkeiten eingehalten.
5. Für jede Tour ist die Summe der Frachtanforderungen aller ihrer Kunden kleiner als die Kapazität des eingesetzten Fahrzeuges.
6. Die Operationszeit für jede Tour ist kleiner als die maximal zulässige Tourlänge.

Während das Einhalten der ersten vier Nebenbedingungen in dem verfolgten Lösungsansatz durch eine geeignete Wahl der Operatoren und der Codierung des Suchraumes erzwungen wird, kann nicht verhindert werden, daß während des Suchprozesses Lösungen generiert werden, die die Tourlängen- bzw. Kapazitätsbeschränkung verletzen. Die Handhabung dieser Nebenbedingungen erfolgt daher über die Einführung von additiven Straftermen, die in der fiktiven Kostenfunktion berücksichtigt werden.

Die Idee für eine geeignete Wahl der Straffunktion basiert darauf, ungültigen Lösungen die Kosten von speziellen gültigen Lösungen zuzuweisen, die durch einen einfachen Transformationsprozeß gefungen werden können.

Dieser Prozeß sollte die gleiche Zeitkomplexität wie die Funktionsauswertung haben und durch die verwendeten Operatoren des Optimierungsverfahrens nachgebildet werden können.

Dies hat zur Folge, daß

- optimale Lösungen bezüglich der Kostenfunktion gültig sind,
- aus ungültigen Lösungen immer gültige Lösungen mit gleichen oder geringeren Kosten generiert werden können,
- gültige Bereiche nicht zu stark isoliert werden.

Kostenfunktion

In die Fitneß- bzw. Kostenfunktion gehen im wesentlichen die Fahrzeiten, die Entladezeiten beim Kunden und die Bestrafungsterme für Verletzungen der Nebenbedingungen ein.

Die Kosten für einen Wochenplan ergeben sich aus der Summe der Kosten aller Tagespläne. Die Kosten für einen Tagesplan berechnen sich wiederum aus den Kosten für alle darin enthaltenen Touren. Die Kosten für eine Tour setzen sich zusammen aus der Fahrzeit, den Entladezeiten und den zusätzlichen Kosten für Verletzungen der Nebenbedingungen (Tourlängen- und Kapazitätsbeschränkung).

Die Verletzung der zeitlichen Tourlängenbeschränkung wird in zwei Stufen bestraft. Wird die normale Arbeitszeit des Fahrers überschritten, so wird ein Überstundenzuschlag berechnet. Ist jedoch die Fahrzeit größer als die maximale Arbeitszeit des Fahrers, werden die Kosten so berechnet, als ob mehrere gültige Touren gefahren würden. Dabei wird nach folgendem Schema vorgegangen :

Eine Tour T_i wird solange in kleinere Touren $T_{i,j}$ aufgespalten, bis jedes $T_{i,j}$ die Nebenbedingungen erfüllen. Dieser Aufspaltungsprozeß hat die gleiche Berechnungskomplexität wie die Funktionsauswertung und kann in die diese integriert werden.

Bei der Kapazitätsbeschränkung der Fahrzeuge wird ähnlich vorgegangen. Die angegebene Frachtkapazität hat eine Sicherheitsmarge von 10 Prozent. Überschreitungen der Normkapazität in diesem Rahmen werden moderat mit einem geringen Kostenfaktor geahndet. Darüberhinausgehende Überschreitungen werden wiederum so berechnet, als ob mehrere gültige Touren verwendet würden.

3.3.3 Datenstruktur

Als Datenstruktur für das ZAD-Problem ist ein Array ausgewählt worden, dessen Elemente über eine doppelt verkettete Liste verbunden sind. Eine doppelt verkettete Liste ermöglicht eine effiziente Implementierung von Operatoren und *Neighborhood Search* Verfahren für Rundtouren. Durch die Einbettung dieser Struktur in ein Array können schnelle Kopien angefertigt werden.

Durch diese Datenstruktur werden die Vorteile von doppelt verketteten Listen und Arrays vereinigt und bildet damit die Grundlage für eine leistungsfähige Implementierung der Optimierungsverfahren.

3.3.4 Initialisierung

Für das Erzeugen von Initialkonfigurationen sind vier Verfahren implementiert worden.

1. *Primitive Tours*

Es werden genau N Touren erzeugt, wobei jede genau einen Kunden bedient.

2. *In Order Initialisation*

Die Kunden werden nach ihrer Kunden-Nr. den Touren zugeordnet. Falls eine Tour die Fracht-Grenze verletzt, wird eine neue Tour erzeugt.

3. *Random Initialisation*

Wie in Punkt (2), jedoch wird die Reihenfolge der Kunden zufällig bestimmt.

4. *SWEEP*

Die Kunden werden nach ihrem Winkel zum Depot sortiert. Dann wird wie in Punkt (2) fortgefahren.

Durch diese Vorgehensweise wird eine regionale Separierung der Kunden verschiedener Touren erreicht.

Die so produzierten Startlösungen werden anschließend mit lokalen Optimierungsverfahren verbessert.

3.3.5 Visualisierung

Bei der Entwicklung leistungsfähiger Optimierungsverfahren für das ZAD-Problem war die Implementierung eines Visualisierungstools wichtig, mit dem eine konkrete Suche animiert und gefundene Problemlösungen graphisch dargestellt werden können.

Ein geschulter menschlicher Beobachter hat ein intuitives Verständnis, wie gute Problemlösungen aussehen und wo schlechte Problemlösungen ihre Schwachstellen haben. Daher kann das Visualisierungstool für die Beurteilung der Leistungsfähigkeit verschiedener Operatoren und die für Aufdeckung potentieller synergetischer Wirkungen verwendet werden.

Außerdem ist das Visualisierungstool wichtig als visuelle Kontrolle für die korrekte Implementierung der Operatoren und als interaktives Kontrollinstrument, mit durch die Vorgabe bestimmter Teiltouren Sondersituationen behandelt werden können. Darüber hinaus kann der Benutzer interaktiv in den Suchprozeß eingreifen, so daß das *Know How* eines erfahrenen Disponenten dem Optimierungsprozeß zur Verfügung gestellt werden kann.

3.3.6 Operatoren und Verbesserungsverfahren in der Problemhierarchie

Die Verfahren zur Lösung der einzelnen Ebenen der Problemhierarchie sind mit der gleichen Herangehensweise entwickelt worden.

Zuerst wurden problemspezifische einstellige Operatoren implementiert, die einerseits als Mutationsoperatoren und andererseits in *Neighborhood Search* Verfahren

verwendet wurden. Dann wurde für diese Operatoren eine effizientere Berechnung der Kostenfunktion über Differenzterme entwickelt.

Anschließend wurden die Mutationsoperatoren und *Neighborhood Search* Verfahren in einen hybriden Evolutionären Algorithmus für die jeweilige Ebene der Problemhierarchie integriert und mit den hybriden Evolutionären Algorithmen der in der Hierarchie tieferliegenden Teilprobleme verknüpft.

Travelling Salesman Problem

Operatoren

Die Basis für die Verfahren zur Lösung des *Travelling Salesman Problem* sind folgende Kantenaustauschoperatoren, die gültige Rundtouren wiederum in gültige Rundtouren transformieren.

1. Zwei-Kantenaustausch (*2-Opt*)
2. K-Kantenaustausch (*k-Opt*)
3. *OR-Opt* Schritt

Verbesserungsverfahren

Die Kantenaustauschoperatoren sind in folgende Standardverbesserungsverfahren integriert worden:

1. *next ascent* Verfahren
2. *steepest ascent* Verfahren
3. Kombination aus *next ascent* und *steepest ascent*

Die Operatoren und Verbesserungsverfahren wurden anschließend in einen hybriden Evolutionären Algorithmus für das *Travelling Salesman Problem* integriert. Bei der Bearbeitung von Benchmarkproblemen aus der TSPLIB95 stellte sich heraus, daß für die zu bearbeitenden Problemgrößen die Verwendung des *2-Opt* Operators ausreicht.

Tagesplanung

Operatoren

Die Grundlage für Verfahren zur Lösung der Tagesplanung (CVRP) sind Kundenaustauschoperatoren zwischen Touren (Tourenklassen) eines Tagesplans, die gültige Tagespläne wiederum in gültige Tagespläne überführen. Die folgenden Operatoren für die Tagesplanung sind so implementiert worden, daß die Randbedingungen der darunterliegenden TSP-Probleme nicht verletzt werden:

1. Austausch von einzelnen Partitionselementen zwischen Touren:
 - (a) Verlege den Besuch des Kunden K_1 aus Tour T_1 hinter den von K_2 aus Tour T_2 .
 - (b) Vertausche den Besuch von K_1 aus Tour T_1 mit dem von K_2 aus Tour T_2 .
2. Austausch von Teiltouren:
 - (a) Vertausche eine Teiltour $Depot \rightarrow \dots \rightarrow K_1$ von Tour T_1 mit einer Teiltour $Depot \rightarrow \dots \rightarrow K_2$ von Tour T_2 .
 - (b) Vertausche eine Teiltour $K_{11} \rightarrow \dots \rightarrow K_{12}$ von Tour T_1 mit einer Teiltour $K_{21} \rightarrow \dots \rightarrow K_{22}$ von Tour T_2 .
3. Verschmelzen von zwei Touren zu einer Tour.
4. Aufspalten einer Tour in zwei Touren.

Verbesserungsverfahren

Bei den Verbesserungsverfahren wird versucht, einen gegebenen Tagesplan durch Anwendung eines Operators zu verbessern. Dies wird solange fortgesetzt, bis keine Verbesserung durch diesen Operator mehr möglich ist bzw. eine Zeitschranke überschritten wird.

Die oben definierten Operatoren sind in folgende Standardverbesserungsverfahren integriert worden:

1. *next ascent* Verfahren
2. *steepest ascent* Verfahren
3. Kombination aus *next ascent* und *steepest ascent*

Die Wirkungsweise dieser Verfahren ist mit dem Visualisierungstool veranschaulicht und bewertet worden.

Die Operatoren und Verbesserungsverfahren wurden anschließend in einen hybriden Evolutionären Algorithmus integriert, dessen Leistungsfähigkeit anhand von bekannten Benchmarkproblemen aus der einschlägigen Literatur überprüft wurde.

Wochen-/Periodenplanung

Operatoren

Die Grundlage für die Verfahren zur Lösung der Periodenplanung sind zwei Operatoren, die einzelne Kunden zwischen verschiedenen Tagesplänen (Tagesklassen) austauschen.

1. Verlege den Besuch des Kunden K_1 am Tag D_1 hinter den von K_2 am Tag D_2 .
2. Vertausche den Besuch von K_1 am Tag D_1 mit dem von K_2 am Tag D_2 .

Um zu erreichen, daß hierbei gültige Periodenpläne wiederum in gültige Periodenpläne überführt werden, wurde eine Hilfsdatenstruktur, der *Pattern Transition Graph* (PTG_i), für jeden Kunden K_i eingeführt.

Die Knoten des PTG_i repräsentieren erlaubte Besuchsmuster für den Kunden K_i . Die Kante zwischen zwei Knoten dieses vollständigen gerichteten Graphen sind mit einer Liste der Kundenaustauschoperationen markiert, die das eine Besuchsmuster in das andere überführen.

Verbesserungsverfahren

Die Integration dieser Operatoren in *Neighborhood Search* Verfahren erfolgt analog wie bei der Tagesplanung. Die beiden oben definierten Operatoren sind wiederum in die Standardverbesserungsverfahren integriert worden:

1. *next ascent* Verfahren
2. *steepest ascent* Verfahren
3. Kombination aus *next ascent* und *steepest ascent*

Anschließend wurden diese Verbesserungsverfahren mit den Verfahren für die Tagesplanung in einen hybriden Evolutionären Algorithmus für die Periodenplanung integriert.

3.4 Problemstellung der Industrievertretung BUB

Bei den Besprechungen mit Vertretern der Industrievertretung BUB wurde die Problemstellung präzisiert folgende Aufgabenverteilung vorgenommen.

Die Fa. BUB erklärte sich bereit, die Beschaffung und Aufbereitung der Eingabedaten zu übernehmen, so daß sich die Arbeitsgruppe der GMD auf die Lösung des Optimierungsproblems konzentrieren konnte.

Die Rahmendaten für die Optimierung des *Real World Problems* sind:

Anzahl der Kunden	:	ca. 300
Periodenlänge	:	5 Werktage (eine Arbeitswoche), wobei der Freitag gesondert behandelt wird
Anzahl der Besuche	:	1-3 mal die Woche
Anzahl der Fahrzeuge	:	6
Frachtraum	:	120 Volumeneinheiten pro Fahrzeug
Entfernungen	:	Straßennetz (Geographische Datenbank)
Entladezeiten pro Kunde	:	5-60 Minuten
Maximale Operationszeit der Fahrzeuge	:	8 Stunden

Bisher verwendet die Fa. BUB einen festen Einsatzplan für 24 Touren mit jeweils 15-30 Kunden. Die Auslastung der Fahrzeuge liegt hierbei bei ca. 60%

3.4.1 Erstellen der Eingabedaten für die Optimierung

Die Datenbeschaffung zur Konkretisierung der Aufgabenstellung umfaßt folgende Bereiche:

- Das Aufstellen einer Kundenliste mit
 - Frachtanforderung,
 - individueller Entladezeit
 - erlaubte Besuchsmuster
- Das Erzeugen der Zeit-Entfernungsmatrix, in der die Fahrzeit von einem Kunden zum anderen erfaßt ist.

Dabei stellte sich heraus, daß das Erzeugen einer brauchbaren Zeit-Entfernungsmatrix die weitaus komplexere Aufgabe ist.

Zeit-Entfernungsmatrix

Zuerst wurde bei der Fa. BUB der Ansatz verfolgt, die Matrix mit Hilfe der vorhandene Software SPEDIA zu erstellen, die sogenannte Speditionskilometer zwischen Postleitzahlenregionen angibt. Diese Speditionskilometer sollen dann mit der empirisch ermittelten Durchschnittsgeschwindigkeit der Fahrzeuge in eine Zeit-Entfernungsmatrix konvertiert werden. Für Kunden mit der gleichen Postleitzahl sollten Erfahrungswerte gesammelt werden.

Parallel dazu wurde in der GMD ein Satz von Plausibilitätstests entwickelt, mit der die von der Fa. BUB erarbeitete Entfernungsmatrix auf Konsistenz überprüft wurden.

Mit Hilfe dieser Plausibilitätstests konnten in der von der Spedition berechneten Entfernungsmatrix sowohl Eingabefehler als auch grundsätzliche Fehler lokalisiert werden, wobei letztere auf das Programm SPEDIA zurückgehen. Ein Dreiecksungleichungstest ergab z.B. daß über die Hälfte der Daten Abweichungen von 10 – 90% aufwiesen.

Die Genauigkeit der Entfernungsmatrix ist wesentlich dafür, inwieweit die theoretische Problemstellung die reale Problemstellung nähert. Daher wurde beschlossen, für die Berechnung der Entfernungsmatrix eine geographische Datenbanksoftware zu benutzen, die folgendes leistet:

- Verschiedenen Strassentypen (Autobahn, Landstrasse etc.) können unterschiedliche Geschwindigkeiten zugeordnet werden.
- Innerstädtischer Informationen für größere Städte und Ballungszentren.
- Pflege des Datenmaterials.

Dazu wurde das geographische Datenbankprogramm PLANTOUR 2.6 mit der digitalen Landkarte für NRW im Maßstab 1:50000 des Anbieters PLS Unternehmensberatung GmbH, Stuttgart, erworben. Zur Berechnung der Zeit-Entfernungsmatrix mit PLANTOUR 2.6 muß die "Geokodierung der Kunden" vorgenommen werden, bei der jeder Kunde der nächsten Kreuzung innerhalb der geographische Datenbank zugeordnet wird.

Bei der Geokodierung der Kunden können zwei Arten von Zuordnungsfehlern vorkommen:

1. Die Straße der Kundenanschrift allein reicht dabei nicht aus, um die nächste zum Kunden gelegene Kreuzung eindeutig zu bestimmen. Dies kann der Fall sein, wenn der Kunde z.B. an einer Hauptstraße liegt, die bei Kreuzungen in der geographischen Datenbank vorkommt.

Um die nächste zum Kunden gelegene Kreuzung eindeutig zu bestimmen, wurden die Fahrer der Fa. BUB instruiert, sich beim nächsten Besuch des Kunden die fehlende, zweite Straße der Kreuzung für eine genaue Geokodierung zu merken.

2. Die Straße der Kundenadresse ist zu klein für den Maßstab 1:50000 und kann in der Datenbank nicht gefunden werden.

Um diese Fälle abzudecken, wurden die Fahrer angewiesen, die Straßennamen der nächsten größeren Kreuzungen zu ermitteln.

Bis zum Abschluß des Projektes konnte die Fa. BUB die Geokodierung für 70 % der Kunden mit der erforderlichen Genauigkeit durchführen.

3.4.2 Umsetzung in die Praxis

Eine praktische Umsetzung maschinell generierter Wochenpläne konnte noch nicht erfolgen, da bis zum Abschluß des Projektes nicht alle notwendigen Eingabedaten zur Verfügung standen. Haupthindernis dafür ist, daß die Berechnung der Kundenentfernungen, basierend auf dem realen Straßennetz, noch nicht für alle Kunden beendet wurde. Dies ist jedoch eine wesentliche Voraussetzung dafür, daß ein mit einem Computer optimierter Tourenplan auch in der Praxis kostengünstig und umsetzbar ist.

Optimierungsläufe für ausgesuchte Tage (Tagesplanung) ergaben jedoch schon ein Einsparungspotential von bis zu 20 % gegenüber dem bisher verwendeten Einsatzplan.

3.5 Leistungsbewertung anhand von Benchmarks

In diesem Abschnitt wird eine Leistungsbewertung der implementierten Evoluti-
onären Algorithmen anhand von Benchmarkproblemen durchgeführt. Zuerst wird die Tagesplanung behandelt, die in der einschlägigen Literatur *Capacitated Vehicle Routing Problem* (CVRP) genannt wird. Anschließend wird das Verfahren für die Periodenplanung, dem sogenannten *Period Vehicle Routing Problem* (PVRP) bewertet. Als Vergleichsgrundlage wird die maximal erreichte Qualität der Lösungen verwendet. Das CVRP und das PVRP sind wichtige Optimierungsprobleme aus dem Bereich des *Operations Research*. In der Literatur lassen sich dazu viele Benchmarkprobleme und Lösungsverfahren finden, die zum Vergleich herangezogen werden können.

3.5.1 Tagesplanung

Die Testprobleme zeichnen sich durch unterschiedliche Charakteristika und Schwierigkeitsgrade aus, die sich aus folgenden Punkten ergeben:

1. Anzahl der Kunden (Problemgröße N)
2. Geographische Verteilung der Kunden:
 - (a) Zufällige Verteilung
 - (b) Cluster von Kunden in der Nähe des Depots (*shortdistance* Kunden).
 - (c) Cluster von Kunden weit weg vom Depot (*longdistance* Kunden).
 - (d) Mischformen aus *shortdistance* und *longdistance* Kunden.
3. Verteilung des Frachtaufkommens.

4. Maximale Kapazität der Fahrzeuge.
5. Position des Depots:
 - (a) z.B. in der Mitte oder
 - (b) am Rand des Auslieferungsgebietes.
6. Betonung des im CVRP enthaltenen TSP-Problems. Dies sind Probleminstanzen, deren Lösungen lange Rundtouren enthalten.
7. Betonung des im CVRP enthaltenen Partitionierungs-Problems Dies sind Probleminstanzen, deren Lösungen viele kurze Rundtouren enthalten.

Eine Probleminstanz wird definiert durch die maximale Kapazität der Fahrzeuge K , die maximale Tourlänge Z , die Entladezeit E und eine Liste, die die geographischen Koordinaten und die Frachtanforderung der Kunden enthält. Die Entfernung zwischen zwei Kunden wird hierbei berechnet über den Euklidischen Abstand zwischen den Kundenkoordinaten. Dabei sind in der Literatur zwei Varianten verbreitet:

1. Der Abstand zwischen zwei Kunden wird allein in reelwertiger (*floating-point*) Darstellung berechnet.
2. Der Abstand zwischen zwei Kunden wird zunächst reelwertig berechnet und anschließend ganzzahlig (*Integer*) gerundet.

Das ganzzahlige Runden kann sich derart auf die Berechnung der Kostenfunktion auswirken, daß Abweichungen bis zu mehreren Prozent entstehen können. Darüber hinaus kann sich durch das Runden die Problemstruktur so ändern, daß ein gemäß des einen Entfernungsmaßes optimaler Tourplan nicht optimal bezüglich des anderen Maßes ist.

Um den Leistungsvergleich des eigenen genetischen Verfahrens mit den Verfahren anderer Autoren auf eine möglichst breite Basis zu stellen, muß jede Problemstellung aus der Literatur als ganzzahlige Probleminstanz und als reelwertige Probleminstanz betrachtet werden.

Die wichtigsten Benchmarks für das CVRP (siehe [CMTS79] und [CE69]) sind aus mehreren Internet Quellen abrufbar, werden dort allerdings unter verschiedenen Namen geführt. Tabelle 3 ordnet die verschiedenen Namen den Probleminstanzen zu und gibt die Werte für die Kundenanzahl N und die Nebenbedingungen Kapazitätsschranke K , Tourlängenschranke Z und Entladezeit E an.

Die Probleminstanzen 1-5, 11-12 und ihre korrespondierenden 6-10, 13-14 unterscheiden sich nur darin, daß bei den letzteren Entladezeiten und die Tourlängenbeschränkung berücksichtigt wird.

Pr.	TSPLIB95	ORLIB	ORLIB	Taillard	N	K	Z	E
1	eil51	VRP8	VRPNC1	c50	50	160	∞	0
2	eilA75	VRP9	VRPNC2	c75	75	140	∞	0
2b	eilB75	VRP9	-	-	75	100	∞	0
2c	eilC75	VRP9	-	-	75	180	∞	0
2d	eilD75	VRP9	-	-	75	220	∞	0
3	eilA101	VRP10	VRPNC3	c100	100	200	∞	0
3b	eilB101	VRP10	-	-	100	112	∞	0
4	-	-	VRPNC4	c150	150	200	∞	0
5	-	-	VRPNC5	c199	199	200	∞	0
6	-	-	VRPNC6	c50	50	160	200	10
7	-	-	VRPNC7	c75	75	140	160	10
8	-	-	VRPNC8	c100	100	200	230	10
9	-	-	VRPNC9	c150	150	200	200	10
10	-	-	VRPNC10	c199	199	200	200	10
11	-	-	VRPNC11	c120	120	200	∞	0
12	-	-	VRPNC12	c100b	100	200	∞	0
13	-	-	VRPNC13	c120	120	200	1040	90
14	-	-	VRPNC14	c100b	100	200	720	50

Tabelle 3: Die Standard-Benchmarks des CVRP (Pr. 1 - 10 aus [CE69] und Pr. 11 - 14 aus [CMTS79]) in verschiedenen Internet Quellen

Pr.	GRASP Hj	TS Hj	GA Hj	GHL	Osman	RT	(1+1)
1	524,61	524,61	524,61	524,61	524,61	524,61	524,61
2	845,50	835,26	835,32	836,37	838,62	835,26	835,26
3	829,18	826,14	827,39	826,14	829,62	826,14	826,14
4	1049,60	1032,71	1032,51	1034,90	1044,35	1028,42	1028,42
5	1342,74	1315,49	1316,91	1329,29	1334,55	1291,45	1291,45
6	555,43	555,43	555,43	555,43	555,44	555,43	555,43
7	914,40	909,68	909,68	913,23	909,68	909,68	909,68
8	865,94	865,94	865,94	865,94	866,75	865,94	865,94
9	1177,67	1164,52	1164,98	1189,79	1164,12	1162,55	1162,55
10	1449,48	1407,21	1410,01	1421,88	1417,85	1395,85	1395,85
11	1045,30	1073,32	1042,12	1043,94	1042,11	1042,11	1042,11
12	820,92	819,56	820,92	822,85	819,59	819,56	819,56
13	1549,11	1546,55	1550,25	1551,63	1547,51	1541,14	1541,14
14	907,08	898,32	898,32	866,37	866,37	866,37	866,37

Tabelle 4: Vergleich der besten Lösungen des hybriden (1+1) Verfahrens mit den besten Lösungen aus [Hjo95] [GHL94] [Osm93] [RT95] für die *floating-point* Instanzen der Standard-Benchmarks des CVRP

Pr.	GHL	HR	NMP	SKL	XK	Best	(1+1)	Diff. (%)
1	521	521	521	521	521	521	521	0
2	832	847	845	831	830	830	830	0
2b	-	1042	1042	1022	1023	1022	1022	0
2c	-	751	739	739	736	736	735	+0.14
2d	-	695	690	690	685	685	682	+0.44
3	815	825	831	815	822	815	815	0
3b	-	1113	-	1082	1080	1080	1067	+1.20
4	1024	1070	1057	1025	1024	1024	1015	+0.88
5	1316	1397	-	1307	1296	1296	1275	+1.62
11	1035	1045	1037	1035	1034	1034	1034	0
12	824		824	820	820	820	820	0

Tabelle 5: Vergleich der besten Lösungen des hybriden (1+1) Verfahrens mit den besten Lösungen aus [GHL94], [HR91], [NMP94], [SKL94], [XK96] für die *Integer* Instanzen der Standard-Benchmarks des CVRP

Problem	N	K	RT	(1+1)	Diff. (%)
tai75a	75	1445	1618.36	1618.36	0
tai75b	75	1679	1344.64	1344.64	0
tai75c	75	1122	1291.01	1291.01	0
tai75d	75	1699	1365.42	1365.42	0
tai100a	100	1409	2047.90	2047.90	0
tai100b	100	1842	1940.61	1940.61	0
tai100c	100	2043	1407.44	1406.20	+0.09
tai100d	100	1297	1581.25	1581.25	0
tai150a	150	1544	3055.23	3055.23	0
tai150b	150	1918	2727.99	2727.77	+0.01
tai150c	150	2021	2362.79	2341.84	+0.89
tai150d	150	1874	2655.67	2645.39	+0.39
tai385	385	65	24435.5	24431.44	+0.02

Tabelle 6: Vergleich der besten Lösungen des hybriden (1+1) Verfahrens mit den besten Lösungen von [RT95] für die *floating-point* Instanzen der Benchmarks für das CVRP nach Rochat und Taillard [RT95]

Die Qualität der besten, mit dem hybriden Evolutionären Algorithmus ((1+1)-Verfahren) gefundenen Lösungen für die *floating-point* Instanzen sind in Tabelle 4 aufgeführt und werden dort mit den Ergebnissen der leistungsfähigsten Verfahren aus der Literatur verglichen. Das Verfahren von Yves Rochat und Eric Taillard [RT95] ist den anderen Verfahren aus der Vergleichsliteratur in Bezug auf die erreichbare Qualität der Lösungen bei den größeren Probleminstanzen (Pr. 4,5,9,10 und 13) überlegen. Der in der GMD entwickelte hybride Evolutionäre Algorithmus konnte bei allen Probleminstanzen die besten Lösungen von Rochat und Taillard finden.

Rochat und Taillard haben in [RT95] noch andere Probleminstanzen bearbeitet, bei denen ausschließlich die Kapazitätsbeschränkung der einzelnen Fahrzeuge berücksichtigt werden. Bei diesen Benchmarks konnte der hybride Evolutionäre Algorithmus die Ergebnisse von Rochat und Taillard erreichen und bei den größeren Probleminstanzen sogar noch verbessern (siehe Tabelle 6).

Der hier verfolgte Genetische Ansatz für die Tagesplanung (CVRP) erwies sich auch bei *Integer* Probleminstanzen als sehr erfolgreich. Bei allen bearbeiteten Probleminstanzen konnten die besten Lösungen aus der Literatur erreicht und in vielen Fällen sogar verbessert werden. Ein Auszug der Ergebnisse dieses Vergleichs ist in Tabelle 5 aufgeführt.

Pr.	L	N	P	K	Z	CGW	CGL	Best	(1+1)	Diff. (%)
p01	3	51	2	160	∞	524.6	524.61	524.61	524.61	0
p02	3	50	5	160	∞	1322.9	1322.87	1322.87	1318.42	+0.34
p03	1	50	5	160	∞	524.6	524.61	524.61	524.61	0
p04	5	75	2	140	∞	840.2	835.43	835.43	835.26	+0.02
p05	6	75	5	140	∞	2046.2	2035.80	2035.80	2027.00	+0.43
p06	1	75	10	140	∞	847.2	836.37	836.37	835.32	+0.13
p07	4	100	2	200	∞	831.1	826.14	826.14	826.14	0
p08	5	100	5	200	∞	2042.0	2034.15	2034.15	2031.82	+0.11
p09	1	100	8	200	∞	828.3	827.84	827.84	826.14	+0.21
p10	4	100	5	200	∞	1611.9	1595.84	1595.84	1581.66	+0.89
p11	4	139	5	235	∞	785.7	784.64	784.64	773.10	+1.47
p12	3	163	5	140	∞	1219.6	1207.84	1207.84	1205.49	+0.19
p13	9	417	7	2000	∞	3538.0	3511.62	3511.62	3497.09	+0.41

Tabelle 7: Vergleich der besten Lösungen des hybriden (1+1) Verfahrens mit den besten Lösungen aus [CGW95] und [CGL95] für die Benchmarkprobleme des PVRP nach [CB84] und [RG93]

3.5.2 Periodenplanung

Die Leistungsbewertung des hybriden Evolutionären Algorithmus für das PVRP wurde anhand der besten gefundenen Lösungen von 42 Benchmarkprobleme aus der Literatur durchgeführt. Als Referenz werden die besten Lösungen der beiden leistungsfähigsten veröffentlichten Verfahren von Chao, Golden und Wasil [CGW95] und von Cordeau, Gendreau und Laporte [CGL95] verwendet.

Eine Probleminstanz wird definiert durch die Periodenlänge P , die Anzahl L und die maximale Kapazität K der Fahrzeuge, die maximale Turlänge Z und eine Kundenliste, die die geographischen Koordinaten, Frachtanforderung, individuelle Entladezeit und erlaubte Besuchsmuster der Kunden enthält. Die Entfernung zwischen zwei Kunden ist der Euklidische Abstand zwischen den Kundenkoordinaten in *floating-point* Darstellung.

Bei allen bearbeiteten Probleminstanzen konnten die besten Lösungen aus der Literatur erreicht und in vielen Fällen sogar verbessert werden. Die Ergebnisse dieses Vergleichs sind in den Tabellen 7, 8 und 9 aufgeführt.

Benchmarks von Cristofides, Beasley, Russel und Gribbin

In Tabelle 7 sind die Ergebnisse zu den Benchmarks von Cristofides und Beasley [CB84] (p01-p11) bzw. Russel und Gribbin [RG93] (p12, p13) aufgeführt.

Pr.	L	N	P	K	CGW	CGL	Best	(1+1)	Diff. (%)
p14	2	20	4	20	954.81	954.81	954.81	954.81	0
p15	2	38	4	30	1862.6	1862.63	1862.63	1862.63	0
p16	2	56	4	40	2875.2	2875.24	2875.24	2875.24	0
p17	4	40	4	20	1614.4	1597.75	1597.75	1597.75	0
p18	4	76	4	30	3217.7	3147.24	3147.24	3147.24	0
p19	4	112	4	40	4846.5	4834.34	4834.34	4834.34	0
p20	4	184	4	60	8367.4	8367.40	8367.40	8367.40	0
p21	6	60	4	20	2216.1	2184.04	2184.04	2167.65	+0.75
p22	6	114	4	30	4436.4	4271.11	4271.11	4266.59	+0.11
p23	6	168	4	40	6769.0	6602.59	6602.59	6578.20	+0.37
p24	3	51	6	20	3773.0	3687.46	3687.46	3680.63	+0.19
p25	3	51	6	20	3826.0	3777.15	3777.15	3763.35	+0.37
p26	3	51	6	20	3834.0	3795.33	3795.33	3795.32	+0.00
p27	6	102	6	20	23401.6	21956.46	21956.46	21934.08	+0.10
p28	6	102	6	20	23105.1	22934.71	22934.71	22751.16	+0.80
p29	6	102	6	20	24248.2	22909.36	22909.36	22892.24	+0.07
p30	9	153	6	20	80982.1	75016.58	75016.58	74764.90	+0.34
p31	9	153	6	20	80279.1	78179.89	78179.89	77706.51	+0.61
p32	9	153	6	20	83838.7	80479.20	80479.20	79769.49	+0.88

Tabelle 8: Vergleich der besten Lösungen des hybriden (1+1) Verfahrens mit den besten Lösungen aus [CGW95] und [CGL95] für die Benchmarkprobleme des PVRP nach [CGW95]

Pr.	L	N	P	K	Z	CGL	(1+1)	Diff. (%)
pr01	2	48	4	200	500	2209.02	2199.37	+0.44
pr02	4	96	4	195	480	3799.28	3786.58	+0.33
pr03	6	144	4	190	460	5218.13	5207.75	+0.20
pr04	8	192	4	185	440	6012.79	5975.91	+0.61
pr05	10	240	4	180	420	6769.80	6693.52	+1.13
pr06	12	288	4	175	400	8422.64	8359.90	+0.74
pr07	3	72	6	200	500	4997.41	4953.62	+0.88
pr08	6	144	6	190	475	7094.52	6990.49	+1.47
pr09	9	216	6	180	450	10370.45	10111.63	+2.50
pr10	12	288	6	170	425	13370.04	12910.72	+3.44

Tabelle 9: Vergleich der besten Lösungen des hybriden (1+1) Verfahrens mit den besten Lösungen aus [CGL95] für die Benchmarkprobleme des PVRP nach [CGL95]

Die Problemstellungen p01-p10 sind periodische Erweiterungen der Benchmarks eil51, eil75 und eil100 des CVRP. Die Instanzen p01, p03, p04, p06, p07, und p09 sind allerdings nur einfache periodische Erweiterungen, bei denen die Kunden nur einmal innerhalb der Planungsperiode anzufahren sind und die erlaubten Besuchsmuster keine Einschränkungen bilden. Die besten Lösungen des PVRP sind hierbei identisch zu denen des CVRP.

Bei den anderen Probleminstanzen müssen mehrfache Besuche innerhalb der Planungsperiode und unterschiedlich Besuchsmuster berücksichtigt werden. Hier konnte der hybride Evolutionäre Algorithmus bei allen Instanzen neue beste Lösungen berechnen.

Benchmarks von Chao, Golden und Wasil

In Tabelle 8 werden die Ergebnisse zu den Benchmarks von Chao, Golden und Wasil [CGW95] gezeigt, bei denen die Verteilung der Kunden in der Ebene die geometrischen Muster *Windmill* (p14 - p23) bzw. *Star of David* (p14 - p23) bilden.

Während der hybride Evolutionäre Algorithmus bei *Windmill* die Resultate von [CGL95] bei drei Instanzen verbessern konnte, fand er bei allen *Star of David* Instanzen neue beste Lösungen.

Benchmarks von Cordeau, Gendreau und Laporte

Die Benchmarkprobleme pr01 - pr10 sind von Cordeau, Gendreau und Laporte in [CGL95] aufgestellt und mit ihrem Tabu-Search Verfahren bearbeitet worden. Gegenüber den Instanzen p01 - p32 zeichnen sie sich dadurch aus, daß sie eine effektive Tourlängenbeschränkung besitzen. Aus Tabelle 9 ist zu entnehmen, daß der hybride Evolutionäre Algorithmus hier bei allen Instanzen neue beste Lösungen mit Verbesserungen bis zu 3.44 % finden konnte.

3.6 Schlußbemerkung

Durch die Kombination von *Neighborhood Search* und genetischer Suche sind leistungsfähige Optimierungsverfahren für das *Capacitated Vehicle Routing Problem* (CVRP) und das *Period Vehicle Routing Problem* (PVRP), entwickelt worden. Dazu wurde ein Ansatz gewählt, bei dem Problemspezifika in die Komponenten Kodierung, Genetische Operatoren und *Neighborhood Search* eingewoben werden.

Die Leistungsfähigkeit der beiden implementierten Verfahren wurden in einer umfangreichen Untersuchung durch die wichtigsten Benchmarks aus der einschlägigen Literatur bewertet. Bei allen bearbeiteten Probleminstanzen der beiden NP-vollständigen Probleme konnten die besten bisher veröffentlichten Lösungen erreicht und bei vielen sogar verbessert werden.

Abschließend läßt sich sagen, daß die in der GMD entwickelten hybriden Evoluti-onären Algorithmen für die beiden *Vehicle Routing* Probleme mit den betrachteten Nebenbedingungen in Bezug auf die Qualität der berechneten Lösungen wohl die im Augenblick leistungsfähigsten implementierten Verfahren sind.

Literatur

- [CB84] N. Christofides and J. E. Beasley. The period routing problem. *Networks*, 14:237–256, 1984.
- [CE69] N. Christofides and S. Eilon. An algorithm for the vehicle dispatching problem. *Operations research*, 20, 1969.
- [CGL95] J.F. Cordeau, M. Gendreau, and G. Laporte. A tabu search heuristic for periodic and multi-depot vehicle routing problem. Technical Report CRT-95-76, Centre de recherche sur les transports, Universite de Mon-treal, 1995.
- [CGW95] I-Ming Chao, Bruce Golden, and Edward Wasil. An improved heuristic for the period vehicle routing problem. *Networks*, 26:25–44, 1995.
- [CMTS79] N. Christofides, A. Mingozzi, P. Toth, and C. Sandi. *The vehicle routing problem*, chapter 11, Combinatorial optimization. John Wiley, Chiche-ster, 1979.
- [GHL94] M. Gendreau, A. Hetz, and G. Laporte. A tabu search heuristic for the vehicle routing problem. *Management Science*, 40:1276–1290, 1994.
- [Hjo95] Curt A. Hjorring. *The Vehicle Routing Problem and Local Search Meta-heuristics*. PhD thesis, Department of Engineering Sciences, University of Auckland, 1995.
- [HR91] F. Harche and P. Raghavan. A generalized exchanged heuristic for the capacitated vehicle routing problem. Working paper, Stern School of Business, New York University, 1991.
- [NMP94] C.E. Noon, J. Mittenenthal, and R. Pillai. A TSSP+1 decomposition stra-tegy for the vehicle routing problem. *European Journal of Operational Research*, 79, 1994.
- [Osm93] J.H. Osman. Metastrategy simulated annealing and tabu search algo-rithms for the vehicle routing problem. *Annals of Operations Research*, 41, 1993.

- [RG93] R.A. Russel and D. Gribbin. A multi-phase approach to the period routing problem. *Networks*, 23(9):1–17, 1993.
- [RT95] Yves Rochat and Éric. Taillard. Probabilistic diversifications and intensifications in local search for vehicle routing. *Journal of Heuristics*, 1(1):140–167, 1995.
- [SKL94] W.R.Jr. Steward, J.P. Kelly, and M. Laguna. Solving vehicle routing problems using generalized assignments and tabu search. Working paper, College of Buisness Administration, University at Boulder, 1994.
- [XK96] Jifeng Xu and James P. Kelly. A network flow-based tabu search heuristic for the vehicle routing problem. Working paper, Graduate School of Buisness, University at Boulder, 1996.

ANLAGE

zum Teilvorhaben TRANSPORTOPTIMIERUNG innerhalb des BMBF-Verbundprojektes HYBRID

Förderkennzeichen: 01 IB 405 A2

Veröffentlichungen

H. Mühlenbein, D. Schlierkamp-Voosen. *The science of breeding and its application to the breeder genetic algorithm BGA* Evolutionary Computation, 1(4): 335-360 (1994)

H. Mühlenbein. *Das Verbundprojekt HYBRID*. in: Statusseminar des BMBF zur Bioinformatik (1995).

A. Reinholz, J. Bendisch. *Single Depot/Multiple Visit Transportoptimierung mit genetischen Algorithmen*. in: Statusseminar des BMBF zur Bioinformatik (1995).

Geplante Veröffentlichungen

U. Bartling, H. Mühlenbein. *Optimization of large parcel distribution systems by the Breeder Genetic Algorithm*. in: Proceedings of the 7th International Conference on Genetic Algorithms (1997).