

Evolutionäre Metaheuristiken als
Problemlösungsmethoden für kombinatorische
Optimierungsprobleme - dargestellt am Beispiel
des Standardproblems der Tourenplanung
mit Zeitfensterrestriktionen

Jörg Homberger und Hermann Gehring

Diskussionsbeitrag Nr. 254

Mai 1998

Diskussionsbeiträge des Fachbereichs Wirtschaftswissenschaft
der FernUniversität Hagen

Herausgegeben vom Dekan des Fachbereichs

Evolutionäre Metaheuristiken als Problemlösungsmethoden für kombinatorische Optimierungsprobleme - dargestellt am Beispiel des Standardproblems der Tourenplanung mit Zeitfensterrestriktionen

Jörg Homberger und Hermann Gehring, FernUniversität Hagen

Zusammenfassung:

Metaheuristiken stellen approximative Problemlösungsmethoden dar, mit denen sich komplexe kombinatorische Optimierungsprobleme effektiv lösen lassen. Im folgenden Beitrag wird aufgezeigt, wie Evolutionsstrategien als evolutionäre Metaheuristiken konzipiert werden können. Dies geschieht am Beispiel des Standardproblems der Tourenplanung mit Zeitfensterrestriktionen. Konkret werden zwei Evolutionsstrategien entwickelt. Ihr Ziel ist die Berechnung eines optimalen Tourenplans, der die vollständige Bedienung der Kunden ermöglicht und die Problemrestriktionen – insbesondere die Zeitfensterrestriktionen – erfüllt. Die Zielfunktion besteht aus der Kombination der Minimierung der Fahrzeuganzahl (primäres Zielkriterium) und der Minimierung der Gesamtentfernung (sekundäres Zielkriterium). Die beiden entwickelten Evolutionsstrategien werden anhand von 58 Problembeispielen aus der Literatur getestet. Die Problemgrößen variieren von 100 bis 417 Kunden und von 2 bis 58 Fahrzeugen. Die berechneten besten Lösungen deuten darauf hin, daß Evolutionsstrategien geeignet sind, sowohl die Fahrzeuganzahl als auch die Gesamtentfernung zu minimieren.

Schlüsselworte:

Evolutionsstrategien, Metaheuristiken, Tourenplanung, Zeitfensterrestriktionen.

Fachbereich Wirtschaftswissenschaft, FernUniversität Hagen
Profilstr. 8, D-58084 Hagen

Tel.: 0049-2331-987-4435

Fax: 0049-2331-987-4447

E-mail: Joerg.Homberger@FernUni-Hagen.de

Evolutionäre Metaheuristiken als Problemlösungsmethoden für kombinatorische Optimierungsprobleme - dargestellt am Beispiel des Standardproblems der Tourenplanung mit Zeitfensterrestriktionen

Jörg Homberger und Hermann Gehring, FernUniversität Hagen

1 Einführung und Problemformulierung

Im vorliegenden Beitrag wird das Standardproblem der Tourenplanung mit Zeitfensterrestriktionen betrachtet, das auch als Vehicle Routing Problem with Time Window (VRPTW) bezeichnet wird. Das VRPTW ist eine Erweiterung des bekannten Standardproblems der Tourenplanung. Es läßt sich wie folgt beschreiben (vgl. Domschke 1990):

Von einem Depot aus sind n Kunden mit Fahrzeugen gleicher Kapazität Q zu bedienen. Für jeden Kunden i , $i = 1, \dots, n$, sind ein Bedarf q_i , eine Bedienungszeit s_i und ein Bedienungszeitfenster $z_i = [e_i, f_i]$ gegeben. Die Intervalluntergrenze e_i bezeichnet den frühest möglichen und die Obergrenze f_i den spätest möglichen Bedienungszeitpunkt. Der Bedarf q_i eines Kunden i ist durch genau eine Bedienung innerhalb des Zeitfensters z_i zu decken. Weiterhin bezeichnen e_0 den frühesten Zeitpunkt der Abfahrt eines Fahrzeugs vom Depot i , $i = 0$, und f_0 den spätesten Zeitpunkt der Ankunft am Depot. Die Standorte des Depots und der Kunden sowie die kürzesten Entfernungen $d_{i,j}$ und die entsprechenden Fahrzeiten $d'_{i,j}$ zwischen je zwei Kunden bzw. Orten seien gegeben. Gesucht ist ein Tourenplan TP , der vorrangig die Anzahl der Fahrzeuge und nachrangig die gesamte Entfernung minimiert. Die Minimierung der Fahrzeuganzahl wird daher auch als primäres Optimierungskriterium und die Minimierung der Gesamtentfernung als sekundäres Optimierungskriterium bezeichnet.

Eine Bedienung von Kunden außerhalb ihrer Zeitfenster ist nicht zulässig. Jedoch können die Zeitfensterrestriktionen noch eingehalten werden, wenn ein Fahrzeug einen Kunden vor dem frühesten Bedienungszeitpunkt erreicht. In diesem Fall muß das Fahrzeug lediglich bis zum frühest möglichen Bedienungszeitpunkt warten. Für die spätere Betrachtung zeitlicher Zusammenhänge seien noch einige Größen eingeführt: Mit den Gleichungen (1) und (2) die frühest mögliche Abfahrtszeit δ_i , sowie mit den Gleichungen (3) und (4) die spätest zulässige Ankunftszeit α_i für das Depot $i = 0$ bzw. den Kunden i , $i \neq 0$ (vgl. Solomon et al. 1988). Der Ort $i-$ bezeichnet dabei den Vorgänger des Ortes i in einem Tourenplan und der Ort $i+$ den Nachfolger.

$$\delta_0 = e_0 \quad \text{für das Depot } i = 0, \quad (1)$$

$$\delta_i = \max \{ \delta_{i-} + d'_{i-,i}, e_i \} + s_i \quad \text{für den Kunden } i, i \geq 1. \quad (2)$$

$$\alpha_0 = f_0 \quad \text{für das Depot } i = 0, \quad (3)$$

$$\alpha_i = \min \{ \alpha_{i+} - d'_{i,i+} - s_i, f_i \} \quad \text{für den Kunden } i, i \geq 1. \quad (4)$$

Nach Lenstra und Rinnooy Kan (1981) sind sowohl das Standardproblem der Tourenplanung, als auch die um Zeitfensterrestriktionen erweiterte Problemstellung der Klasse der NP-harten kombinatorischen Optimierungsprobleme zuzurechnen. Für diese Problemklasse existieren bislang keine exakten Lösungsverfahren, deren Berechnungsaufwand lediglich polynomial mit der Problemgröße ansteigt. Bekannt sind vielmehr nur exakte Verfahren mit exponentiell ansteigenden Berechnungsaufwand. Aufgrund der Zeitfensterrestriktionen ist das erweiterte Standardproblem der Tourenplanung nach Solomon et al. (1988) "wesentlich schwieriger" zu lösen als das Standard-

problem ohne Zeitfenster. Es verwundert daher nicht, wenn insbesondere zur Lösung größerer Problembeispiele des Standardproblems mit Zeitfensterrestriktionen primär heuristische Verfahren vorgeschlagen werden. In jüngerer Zeit werden vor allem mit Metaheuristiken wie Tabu Search (vgl. Osman 1993, Taillard et al. 1996), Simulated Annealing (vgl. Chiang und Russell 1996) und Genetischen Algorithmen (vgl. Thangiah et al. 1991) recht gute Ergebnisse für das Standardproblem bzw. für die um Zeitfensterrestriktionen erweiterte Problemstellung erzielt. Diese Metaheuristiken haben eine gemeinsame Eigenschaft: Sie lenken den Einsatz einer untergeordneten Heuristik gemäß einem aus der Künstlichen Intelligenz, Biologie, Mathematik, Natur oder Physik abgeleiteten Konzept, um ihre Leistungsfähigkeit zu verbessern (vgl. Osman 1993). Zur Gruppe der Metaheuristiken gehören auch die Evolutionsstrategien: Einerseits gehen sie gemäß einem aus der biologischen Evolution entlehnten Konzept vor und andererseits setzen sie hierbei untergeordnete problembezogene Heuristiken ein. Über die Anwendung von ES auf das Standardproblem der Tourenplanung mit Zeitfensterrestriktionen wurde in der Literatur noch nicht berichtet. Ablay (1979) und Nissen (1994) wenden ES jedoch auf andere Problemstellungen aus dem Bereich der kombinatorischen Optimierung an.

ES wurden in den 70er Jahren von Rechenberg (1973) und Schwefel (1977) zur Lösung von Optimierungsproblemen mit reellwertigen Variablen entwickelt und seitdem auch erfolgreich eingesetzt. Als besonders geeignete Verfahrensvariante konnte sich die (μ, λ) -Evolutionsstrategie von Schwefel (1977) etablieren (vgl. Nissen 1994). Sie wird daher im folgenden zur Lösung des Standardproblems der Tourenplanung mit Zeitfensterrestriktionen eingesetzt. Zuvor werden jedoch einige grundlegende Eigenschaften von ES beschrieben und es wird eine allgemeine, aus der (μ, λ) -Evolutionsstrategie bestehende Metaheuristik vorgestellt. Die Anpassung dieser allgemeinen Metaheuristik an das Standardproblem der Tourenplanung mit Zeitfensterrestriktionen erfolgt auf zweierlei Weise. Die beiden resultierenden Verfahrensvarianten werden im Detail beschrieben und einem vergleichenden Test unterzogen. Der Test wird mit den von Solomon (1987) und Russell (1995) veröffentlichten Problembeispielen durchgeführt.

2 Evolutionsstrategien als Metaheuristiken

2.1 Grundlagen von Evolutionsstrategien

Evolutionsstrategien bilden zusammen mit den Genetischen Algorithmen (GA) die Klasse der evolutionären Algorithmen (vgl. Nissen 1994). Andererseits stellen ES und GA auch Metaheuristiken dar. Sie können daher – zwecks Abgrenzung gegenüber anderen Metaheuristiken – auch als evolutionäre Metaheuristiken bezeichnet werden.

Ähnlich wie die GA manipulieren auch ES Populationen von Individuen, wobei die Individuen Lösungen eines Optimierungsproblems repräsentieren. Durch die iterative Berechnung einer Folge von Populationen wird ein stochastischer Suchprozeß auf dem Lösungsraum realisiert. Ein hierbei angewandter Selektionsmechanismus begünstigt die Erzeugung besserer Lösungen. Unterschiede zu GA bestehen vor allem hinsichtlich der Problemrepräsentation und der Verfahrensoperatoren. So verzichten ES auf die für GA charakteristische Codierung von Individuen und simulieren statt dessen den Evolutionsprozeß unmittelbar auf der Ebene von Problemlösungen. Entsprechend werden mit den Verfahrensoperatoren nicht codierte Individuen, sondern unmittelbar Problemlösungen manipuliert. Anders als bei GA kommt dabei den Mutationsoperatoren im Vergleich zu den Rekombinationsoperatoren eine übergeordnete Rolle zu (vgl. hierzu Gehring und Schütz 1994).

Ursprünglich wurden ES zur Lösung von Optimierungsproblemen mit reellwertigen Entscheidungsvariablen entwickelt. Im Falle der Optimierung einer Zielfunktion $F(X)$ mit n reellwertigen Entscheidungsvariablen $x_i, i = 1, \dots, n$, läßt sich ein Individuum als ein n -dimensionaler reellwertiger Lösungsvektor $X \in \mathbb{R}^n$ darstellen. Zur Manipulation von Populationen reellwertiger Lösungsvektoren wurden verschiedene Evolutionsstrategien entwickelt. Als Beispiele genannt seien die (1+1)-Evolutionsstrategie von Rechenberg (1973) und die (μ, λ) -Evolutionsstrategie von Schwefel (1977). Die Unterschiede zwischen diesen Verfahren betreffen primär die Populationsstärke, die Repräsentation von Individuen und die Verfahrensoperatoren. Die (1+1)-Evolutionsstrategie arbeitet mit einer Populationsstärke von nur einem Individuum, die (μ, λ) -Evolutionsstrategie mit einer Populationsstärke von mehreren Individuen. Außerdem verzichtet die (1+1)-Evolutionsstrategie völlig auf einen Rekombinationsoperator und erzeugt die Nachkommen ausschließlich durch Mutation von reellwertigen Lösungsvektoren. Bei der (μ, λ) -Evolutionsstrategie umfaßt ein Individuum dagegen neben einem reellwertigen Lösungsvektor noch eine zweite Komponente, den ebenfalls reellwertigen Vektor σ der sogenannten Strategieparameter. Beide Komponenten werden im allgemeinen bei der Generation neuer Populationen mittels Rekombinations- und Mutationsoperatoren verändert. Einzelheiten sind den nachfolgenden, auf die (μ, λ) -Evolutionsstrategie beschränkten Ausführungen zu entnehmen.

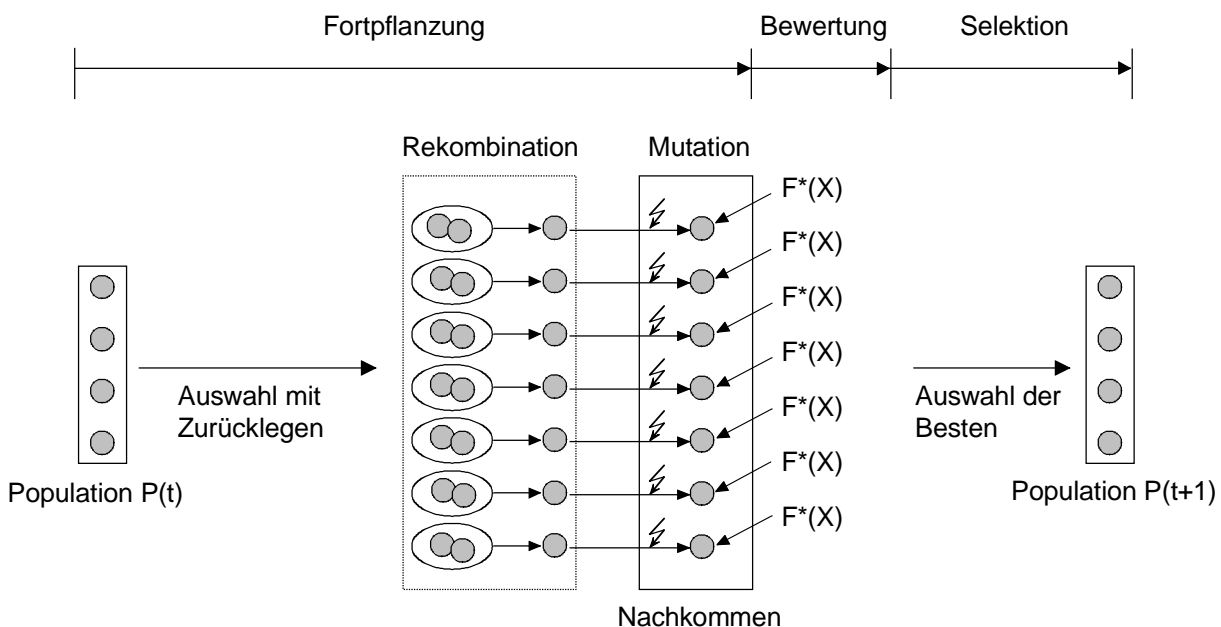


Abb. 1. (μ, λ) -Evolutionsstrategie.

Die (μ, λ) -Evolutionsstrategie basiert auf dem in Abb. 1 gezeigten Fortpflanzungsprozeß. Demnach wird ausgehend von einer Population $P(t)$ mit μ Individuen eine Anzahl von λ , $\lambda > \mu$, Nachkommen erzeugt. Zur Berechnung eines Nachkommen werden aus $P(t)$ mehrere Individuen, auch Eltern genannt, ausgewählt. Die Auswahl erfolgt zufällig und mit Zurücklegen. Die Wahrscheinlichkeit der Auswahl ist für jedes Individuum gleich und unabhängig von den Fitnesswerten $F^*(X)$ der Individuen. Die Fitnesswerte $F^*(X)$ sind das Ergebnis einer Bewertung der Individuen; im einfachsten Fall entsprechen sie den Zielfunktionswerten $F(X)$. Durch Rekombination der selektierten Eltern wird nun genau ein Nachkomme berechnet und anschließend einer Mutation unterworfen. Aus der auf diese Weise erzeugten Menge von λ Nachkommen werden schließlich μ Individuen der Folgepopulation $P(t+1)$ selektiert. Als Selektionskriterium dienen nun die Fitnesswerte $F^*(X)$; es werden also die n Individuen mit den höchsten Fitnesswerten $F^*(X)$ ausgewählt. Da die Eltern nicht am Selektionsprozeß teilnehmen, werden Verschlechterungen während der Suche zugelassen. Lokale Optima können daher auch wieder verlassen werden.

Mit dem aus den beiden Parametern μ und λ gebildeten Koeffizienten μ/λ steht ein einfacher Mechanismus zur Verfügung, mit dem der globale Charakter der Suche – explorativ oder exploitativ – beeinflusst werden kann. Ein hoher Wert dieses Quotienten führt zu einem geringen Selektionsdruck. Entsprechend weisen die Individuen berechneter Folgepopulationen eine größere Diversität auf und die Suche erhält einen mehr explorativen Charakter (vgl. Hoffmeister und Bäck 1992).

Auch bei der (μ, λ) -Evolutionstrategie kommt der Mutation eine besondere Rolle zu. Der Lösungsvektor X eines Individuums I wird mutiert, indem an X kleine zufällige Änderungen mittels einer Mutationsvorschrift vorgenommen werden. Als kritisch erweist sich hierbei die Wahl der mittleren Größe der Änderungen, der sogenannten Mutationsschrittweite. Charakteristisch für die (μ, λ) -Evolutionstrategie ist die Dynamisierung der Mutationsschrittweite in einer Weise, die sich positiv auf den Erfolg der Suche auswirkt. Zu diesem Zweck wird ein Individuum I um den bereits erwähnten Vektor σ von Strategieparametern erweitert. Die Repräsentation eines Individuums lautet also:

$$I = (X, \sigma), \text{ mit: } X = \{x_i | i = 1, \dots, n\}, \sigma = \{\sigma_i | i = 1, \dots, n\}. \quad (5)$$

Jedem Element x_i des reellwertigen Lösungsvektors X ist genau ein Element σ_i des reellwertigen Vektors σ von Strategieparametern zugeordnet. Im gegebenen Fall stellen die Strategieparameter σ_i , $i = 1, \dots, n$, Standardabweichungen normalverteilter Zufallsgrößen mit dem Erwartungswert 0 dar. Ein Individuum wird mutiert, indem u.a. die Elemente des Lösungsvektors gemäß der in Gleichung (6) dargestellten Mutationsvorschrift verändert werden.

$$X' = X + NORM(\vec{0}, \sigma), \text{ mit: } NORM(\vec{0}, \sigma) = \{NORM_i(0, \sigma_i) | i = 1, \dots, n\}. \quad (6)$$

Für jedes Element x_i des Lösungsvektors X stellt also der Wert der Änderung eine Realisation der zugehörigen Normalverteilung $NORM_i(0, \sigma_i)$ dar.

Im Fortplanzungsprozeß werden nun beide Teile, X und σ , der jeweiligen Eltern an die erzeugten Nachkommen vererbt. Neben den Lösungen werden damit auch Informationen über die Werte der Strategieparameter, die der Erzeugung dieser Lösungen zugrunde lagen, an die Folgepopulation weitergegeben. Auf diese Weise wird ein zweigleisiger Lernprozeß ermöglicht, der einerseits zu tendenziell besseren Lösungen und andererseits zu geeigneteren Werten der Strategieparameter bzw. Mutationsschrittweiten führt. Um das Erlernen geeigneter Werte der Strategieparameter zu begünstigen, wird der Vektor der Strategieparameter σ durch Rekombination der Strategieparameter der jeweiligen Eltern verändert.

2.2 Eine allgemeine evolutionäre Metaheuristik für kombinatorische Optimierungsprobleme

Auf kombinatorische Optimierungsprobleme läßt sich die beschriebene (μ, λ) -Evolutionstrategie nicht unmittelbar anwenden. Erforderlich sind vielmehr einige Modifikationen, welche die Ganzzahligkeit der Lösungen von kombinatorischen Optimierungsproblemen berücksichtigen. Dabei ergibt sich bei den einzelnen Komponenten der (μ, λ) -Evolutionstrategie, nämlich

- dem der Metabene zuzuordnenden (μ, λ) -Selektionsmechanismus,
- der aus Lösungsvektoren und Strategieparametern zusammengesetzten Repräsentation von Individuen und
- der mittels untergeordneter Heuristiken wie z.B. Mutation realisierten lokalen Suche im Lösungsraum, ein unterschiedliches Bild.

Die Strategie der (μ, λ) -Selektion abstrahiert völlig von der konkreten Problemrepräsentation. Sie kann daher ungeändert übernommen werden.

Änderungen ergeben sich zwangsläufig bei der Repräsentation von Individuen. Der Lösungsvektor X besteht nun aus n ganzzahligen Elementen. Nachfolgend werden ganzzahlige Lösungsvektoren mit S , $SVektor$ oder SV abgekürzt. Je nach Problemstellung kann es sinnvoll sein, die Vektorelemente in Gruppen zu unterteilen. Zwar kann die bisherige Form des Vektors σ der Strategieparameter nicht beibehalten werden, wohl aber die dahinterstehende Idee der Mutation und der Anpassung der Mutationsschrittweite. An die Stelle des Vektors σ treten nun Informationen, welche den Übergang von einer gegebenen Lösung S zu einer Lösung S' in der Nachbarschaft $N(S)$ von S "codieren". Sofern, wie bei dem Standardproblem der Tourenplanung mit Zeitfensterrestriktionen, zwei Optimierungskriterien zu berücksichtigen sind, können zusätzlich Informationen über die Suchrichtung in die Repräsentation der Individuen übernommen werden. Bei der Generierung einer neuen Lösung $S' \in N(S)$ liegt dann das Gewicht mehr auf dem ersten oder dem zweiten Optimierungskriterium.

Die auf der Ebene der lokalen Suche eingesetzte Heuristik umfaßt zwei konzeptionelle Elemente, einen Nachbarschaftsbegriff und eine Vorschrift zur Generierung von Nachbarschaftslösungen. Bei dem hier verwendeten Nachbarschaftsbegriff besteht die Nachbarschaft $N(S)$ der Lösung S aus der Menge der zulässigen Lösungen, die sich ausgehend von S mittels einer Vorschrift erzeugen lassen. Die Vorschrift zur Erzeugung von Nachbarschaftslösungen kann somit als ein "move-generation mechanism" gemäß Osman (1995) konzipiert werden. Osman führt hierzu folgendes aus:

"A move-generation mechanism generates the set of neighbours changing one attribute or a combination of attributes of a given instance S . A move-generation is a transition from a solution S to another solution $S' \in N(S)$ in one step (or iteration)".

Bewirkt ein so definierter Zug nur relativ geringe Änderungen einer gegebenen Lösung S , so kann er als Mutationsoperator begriffen werden. Auf der Hand liegt nun auch ein einfaches Konzept zur Variation der Mutationsschrittweite: Bei der Erzeugung einer neuen Lösung S' werden entweder nur ein oder mehrere Züge durchgeführt.

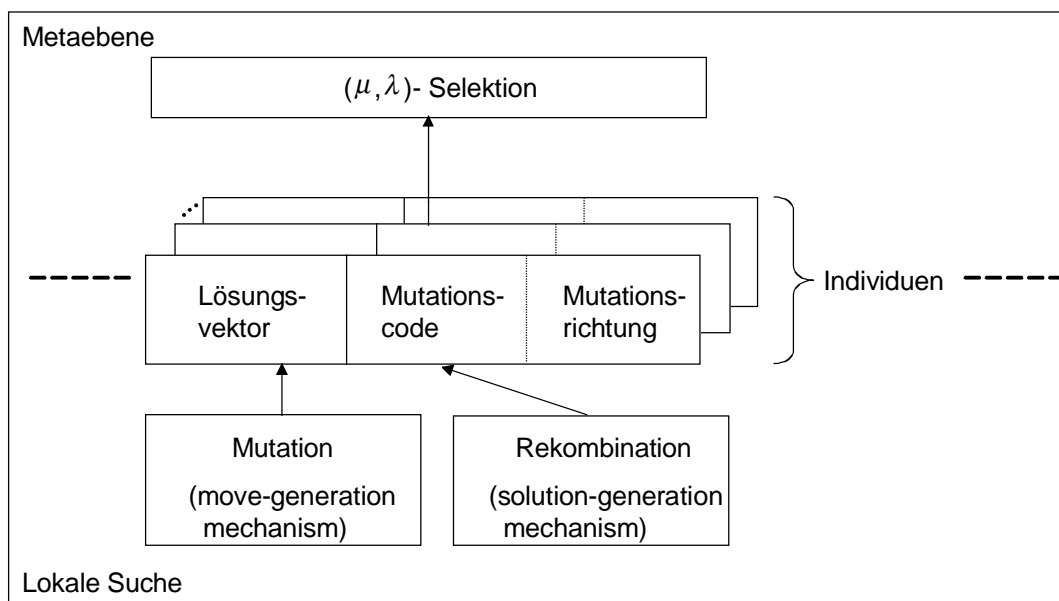


Abb. 2. Konzept einer evolutionären Metaheuristik.

Eine Modifikation des Nachbarschaftsbegriffs wäre erforderlich, wenn man zur Erzeugung einer neuen Lösung einen "solution-generation mechanism" nach Osman (1995) einsetzen würde; denn dieser Mechanismus setzt auf mehreren Lösungen auf:

"A solution-generation mechanism works on a set of solutions rather than a set of attributes for a single solution".

Bei dem hier vorgestellten Ansatz wird der "solution-generation mechanism" nicht zur Erzeugung neuer Lösungsvektoren eingesetzt, wohl aber zur Anpassung der Mutationsvorschrift bzw. des "move-generation mechanism". Dies ist stets dann möglich, wenn für einen Zug eine geeignete Repräsentationsform – beispielweise als Permutationsvektor – gefunden werden kann. In diesem Fall bieten sich zur Rekombination der Mutationsvorschriften zweier Individuen die verschiedenen in der Literatur beschriebenen Crossover-Operatoren an. So beispielsweise der Cycle Crossover von Oliver et al. (1987) oder der Uniform Order-Based Crossover von Davis (1991).

Insgesamt ergibt sich damit das in Abb. 2 veranschaulichte Konzept einer evolutionären Metaheuristik für kombinatorische Optimierungsprobleme.

Auf der Metaebene wird die Sequenz der erzeugten Populationen mittels der (μ, λ) -Selektion gesteuert. Wie bisher kann dabei der globale Charakter der Suche über den Quotienten μ/λ variiert werden. Ebenso wird ein zweigleisiger, auf die Verbesserung der Lösungen abzielender Lernprozeß ermöglicht: Einerseits führt die Mutation von Individuen in Verbindung mit der (μ, λ) -Selektion tendenziell zu besseren Lösungen. Andererseits führt die Rekombination des Mutationscodes von Individuen, wiederum in Verbindung mit der (μ, λ) -Selektion, zumindest in der Tendenz zu geeigneteren Ausprägungen des Mutationscodes.

Eine vereinfachte Variante der skizzierten Metaheuristik ergibt sich, wenn man auf die Rekombination des Mutationscodes verzichtet. In diesem Fall findet nur ein einglisiger Lernprozeß statt. Andererseits sind auch komplexere Varianten denkbar. So könnte man z.B. den durch Rekombination erzeugten Mutationscode zusätzlich einer Mutation unterwerfen. Nachfolgend werden lediglich zwei Varianten behandelt, die in Abb. 2 dargestellte Version und die vereinfachte Variante.

3 Anpassung der allgemeinen evolutionären Metaheuristik an das VRPTW

In den beiden folgenden Kapiteln werden die beiden Varianten der allgemeinen evolutionären Metaheuristik an das Standardproblem der Tourenplanung mit Zeitfensterrestriktionen angepaßt; in Kapitel 3.1 die vereinfachte Variante und in Kapitel 3.2 die in Abb. 2 dargestellte Version. Im Vordergrund stehen dabei die Konkretisierung der Repräsentation von Individuen und die Ausgestaltung der Mutationsoperatoren und des Rekombinationsoperators. Die Verfahrensbestandteile, die die Bewertung von Individuen und eine Verfeinerung der (μ, λ) -Selektion betreffen, sind in beiden Fällen gleich. Sie werden daher separat in dem Kapitel 3.3 behandelt. Schließlich wird in Kapitel 3.4 auf die Konfigurierung der beiden Evolutionsstrategien eingegangen.

3.1 Evolutionsstrategie ES1

Bei der Evolutionsstrategie ES1 wird bekanntlich auf die Rekombination des Mutationscodes verzichtet. Für die Erzeugung eines neuen Individuums ergibt sich damit die in Abb. 3 veranschaulichte Vorgehensweise.

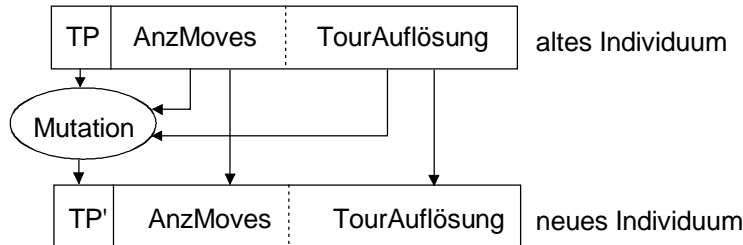
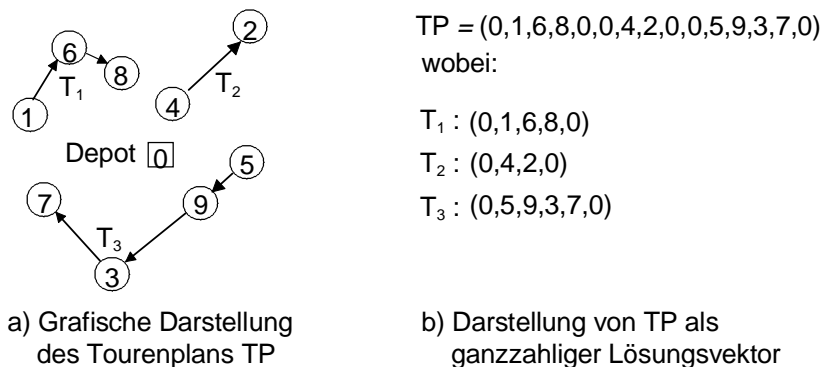


Abb. 3. Erzeugung eines neuen Individuums im Rahmen der Evolutionsstrategie ES1.

Nachfolgend wird zuerst die Repräsentation eines Individuums und danach die Mutationsvorschrift präzisiert.

In Abb. 3 bezeichnet TP eine zulässige Lösung des Standardproblems der Tourenplanung mit Zeitfensterrestriktionen, also einen zulässigen Tourenplan. Wie die Abb. 4 an einem einfachen Beispiel verdeutlicht, kann ein Tourenplan in Form eines ganzzahligen Lösungsvektors repräsentiert werden. Aus Gründen der Übersichtlichkeit wurden in Abb. 4 a) die Verbindungen der Anfangs- und Endkunden der einzelnen Touren mit dem Depot nicht dargestellt.



a) Grafische Darstellung des Tourenplans TP

b) Darstellung von TP als ganzzahliger Lösungsvektor

Abb. 4. Beispiel für die Repräsentation eines Tourenplans.

Für die beiden strategieorientierten Parameter $AnzMoves$ und $TourAuflösung$ gilt:

- $AnzMoves$ gibt an, wie oft der verwendete Move-Operator im Rahmen einer Mutation ausgeführt wird.
- Mit dem zweiwertigen Parameter $TourAuflösung \in \{0, 1\}$ wird die Suchrichtung verschlüsselt: Der Wert 0 bezeichnet die reine Entfernungsminimierung und der Wert 1 ein zweiphasiges Vorgehen, bei dem im Anschluß an eine Entfernungsminimierung der Versuch unternommen wird, die Anzahl der Fahrzeuge zu minimieren.

Mit dem Parameter $AnzMoves$ wird also nur die Mutationsschrittweite vorgegeben. Eine weitergehende Einflußnahme auf den Ablauf einer Mutation erfolgt nicht. Die Werte von $AnzMoves$ und $TourAuflösung$ werden, wie die Abb. 3 erkennen läßt, unverändert von einem Individuum auf ein neues Individuum vererbt.

Zum Zweck der Fortpflanzung wird der zulässige Tourenplan TP eines Individuums mittels einer Mutationsvorschrift variiert. Das Ergebnis ist ein zulässiger und zufällig aus der durch die Restriktionen des Standardproblems der Tourenplanung mit Zeitfensterrestriktionen einge-

schränkten Nachbarschaft $N(TP)$ ausgewählter Tourenplan $TP' \in N(TP)$. Was die Wahl einer geeigneten Nachbarschaftsstruktur betrifft, wird hier das komplexere Konzept des "Compound-Move" verwendet. Compound-Moves setzen sich aus verschiedenen einfachen Move-Operatoren zusammen. Nach Glover (1991) und Osman (1993) stellen sie ein besonders effektives Konzept dar. Im gegebenen Fall werden zwei einfache Moves in einen Compound-Move einbezogen:

- Ein aus der Menge $MoveSet = \{Or-opt-Move, 2-opt^*-Move, 1-Interchange-Move\}$ zufällig ausgewählter Move.
- Ein modifizierter Or-opt-Move, der dem Ziel der Verringerung der Fahrzeuganzahl durch die Auflösung einer Tour dient.

Dieser Compound-Move wird allerdings nur bei bestimmten Individuen angewandt; im Fall der übrigen Individuen reduziert sich der Compound-Move auf einen einfachen Move. Einzelheiten sind der Abb. 5 zu entnehmen, in der die gesamte Mutationsvorschrift beschrieben wird.

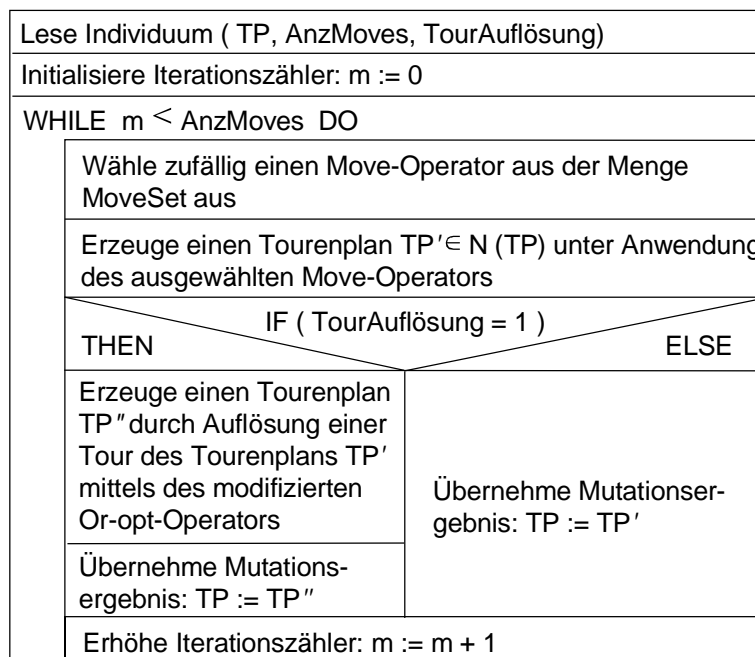


Abb. 5. Mutationsvorschrift der Evolutionsstrategie ES1.

Wie die Abb. 5 zeigt, wird ein Compound-Move nur ausgeführt, falls das mutierte Individuum dies explizit verlangt. Nur wenn es auch tatsächlich gelingt, die fragliche Tour aufzulösen, führt ein Compound-Move zum Erfolg. Andernfalls ändert sich die Anzahl der Touren nicht. Die Beschränkung des Compound-Move auf eine Teilmenge von Individuen ist sinnvoll, da die Auflösung einer Tour oftmals zu einer Vergrößerung der Gesamtdistanz führt.

Zur weiteren Präzisierung der Mutationsvorschrift sind einige Anmerkungen zu den Move-Operatoren erforderlich.

Die Operatoren der Menge $MoveSet$ gehen auf folgende Autoren zurück: Dem Or-opt-Move liegt ein Tauschkonzept von Or (1976) zugrunde. Weitere Move-Konzepte führten Osman (1993) mit dem 1-Interchange-Move sowie Potvin und Rousseau (1995) mit dem 2-opt*-Move ein. Exemplarisch sei nachfolgend der Or-opt-Move kurz beschrieben.

Ein Or-opt-Move faßt eine Ausfüge- und eine Einfügeoperation zusammen. Mittels dieser Operationen wird ein Kunde oder eine Sequenz von mehreren Kunden aus einer Tour T eines Tourenplans TP ausgefügt und anschließend an einer anderen Stelle von TP wieder eingefügt. Im Falle nur eines Kunden gilt:

- Ein Kunde k wird aus der ihn bedienenden Tour $T = (0, \dots, k-, k, k+, \dots, 0)$ ausgefügt, indem die verbleibenden Kunden der Tour T unter Ausschluß des Kunden k in der ursprünglichen Reihenfolge bedient werden. Die um den Kunden k reduzierte Tour T_1 lautet dann: $T_1 = (0, \dots, k-, k+, \dots, 0)$. Handelt es sich bei der Tour T um eine Pendeltour, d.h. das Fahrzeug bedient ausschließlich den Kunden k , so wird der Kunde k aus der Tour T ausgefügt, indem die Tour T aufgelöst wird.
- Ein ausgefügter Kunde k wird in eine Tour T_2 des Tourenplans eingefügt, indem die Tour T_2 um die Bedienung des Kunden k erweitert wird. Wird die Einfügestelle durch zwei aufeinanderfolgende Orte $(i, i+)$ bezeichnet, so gilt: $T_2 = (0, \dots, i, i+, \dots, 0)$. Durch das Einfügen des Kunden k ergibt sich die Tour $T_3 = (0, \dots, i, k, i+, \dots, 0)$. Grundsätzlich ist es nicht ausgeschlossen, daß ein Kunde k wieder in die Tour eingefügt wird, aus der er ausgefügt wurde. Allerdings kann sich dann die Einfügestelle von der Ausfügestelle unterscheiden.

Gemäß Abb. 5 zielt der modifizierte Or-opt-Operator auf die Berechnung eines Tourenplans mit einer geringeren Fahrzeuganzahl ab. Zu diesem Zweck wird versucht, diejenige Tour $T \in TP'$ des aktuellen Tourenplans TP' aufzulösen, welche die geringste Anzahl von Kunden aufweist. Die Tour mit der geringsten Anzahl von Kunden wird nachfolgend als "kleinste" Tour bezeichnet. Das Einfügen von Kunden wird wie folgt präzisiert: Die Kunden der Tour T werden entsprechend ihrer Bedienungsreihenfolge nacheinander in je eine Tour $T' \neq T$, $T' \in TP'$, eingefügt, sofern die Restriktionen des Standardproblems der Tourenplanung mit Zeitfensterrestriktionen nicht verletzt werden. Existieren für einen Kunden $k \in T$ alternative Einfügestellen $(i, i+)$ in anderen Touren, so sind zwei Fälle zu unterscheiden:

- Falls ein Index $i+$ einen Kunden bezeichnet, wird die durch das Einfügen bewirkte Verschiebung der frühest möglichen Abfahrtszeit bei dem Kunden $i+$ als Entscheidungskriterium herangezogen.
- Falls ein Index $i+$ das Depot bezeichnet, dient dagegen die durch das Einfügen bewirkte Verschiebung der frühest möglichen Ankunftszeit am Depot $i+$ als Entscheidungskriterium.

Ausgewählt wird also die Einfügestelle mit der geringsten Verschiebezeit $PT_k(i, i+)$, wobei sich diese wie folgt berechnet:

$$PT_k(i, i+) = \begin{cases} \max\{\max\{\delta_i + d'_{i,k}, e_k\} + s_k + d'_{k,i+}, e_{i+}\} + s_{i+} - \delta_{i+} & \text{wenn } i+ > 0, \\ \max\{\delta_i + d'_{i,k}, e_k\} + s_k + d'_{k,0} - (\delta_i + d'_{i,0}) & \text{sonst.} \end{cases} \quad (7)$$

Existieren mehrere zulässige Einfügestellen mit gleicher minimaler Verschiebezeit, so wird diejenige Einfügestelle gewählt, für die die zusätzlich zu fahrende Strecke $PD_k(i, i+)$ ein Minimum annimmt. Für $PD_k(i, i+)$ gilt:

$$PD_k(i, i+) = d_{i,k} + d_{k,i+} - d_{i,i+}. \quad (8)$$

Sofern kein Kunde der Tour T wegen drohender Verletzung der Restriktionen in eine andere Tour $T' \neq T$, $T' \in TP'$, eingefügt werden kann, wird keine Veränderung an TP' vorgenommen.

Durch Variation der Strategieparameter *AnzMoves* und *TourAuflösung* kann das Suchverhalten der Evolutionsstrategie ES1 verändert werden. So begünstigen größere Werte der Mutationsschrittweite das Verlassen lokaler Optima, während kleinere Werte eine Intensivierung der lokalen Suche zur Folge haben. Eine derartige Intensivierung ist beispielsweise angezeigt, wenn – bei fixierter Fahrzeuganzahl – die Gesamtentfernung minimiert werden soll. Größere Werte von *AnzMoves* wird man daher mit dem Ziel der Minimierung der Fahrzeuganzahl (*TourAuflösung* = 1) verbinden, und kleinere Werte mit dem Ziel der Entfernungsminimierung (*TourAuflösung* = 0).

3.2 Evolutionsstrategie ES2

Der Evolutionsstrategie ES2 liegt das in Abb. 2 dargestellte Konzept einer evolutionären Metaheuristik zugrunde. Dieses bereits erläuterte Konzept wird nachfolgend an das Standardproblem der Tourenplanung mit Zeitfensterrestriktionen angepaßt. Dabei stehen das Zusammenspiel von Rekombination und Mutation, sowie die Gestaltung des Rekombinationsprozesses und der Mutationsvorschrift im Vordergrund.

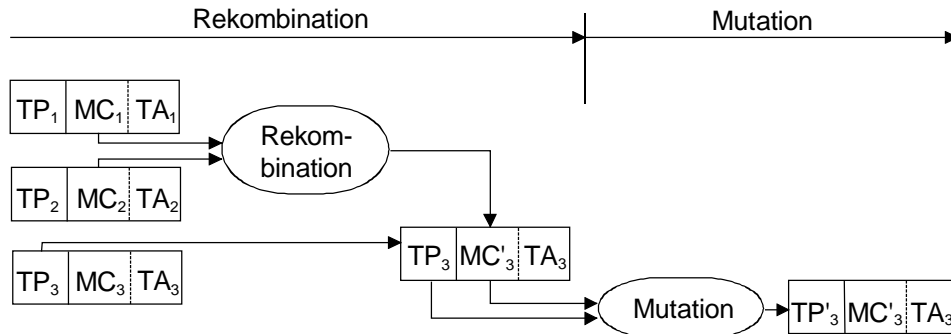


Abb. 6. Erzeugung eines neuen Individuums im Rahmen der Evolutionsstrategie ES2.

Die Abb. 6 veranschaulicht, wie Rekombination und Mutation ineinandergreifen. Ausgehend von drei Eltern wird demnach ein Nachkomme in zwei aufeinanderfolgenden Schritten erzeugt. Der nach dem ersten Schritt vorliegende vorläufige Nachkomme stellt die Verbindung zwischen beiden Schritten her.

Im ersten Schritt wird ein vorläufiger Nachkomme erzeugt, indem

- die Mutationscodes zweier Eltern rekombiniert werden und der resultierende Mutationscode MC'_3 in den vorläufigen Nachkommen übernommen wird;
- der Lösungsvektor TP_3 und die Mutationsrichtung TA_3 eines dritten Elternteils in den vorläufigen Nachkommen kopiert werden.

Bei der Erzeugung des Mutationscodes MC'_3 des vorläufigen Nachkommen wird folglich das Prinzip der sexuellen Fortpflanzung nachgeahmt. Dagegen basiert der beschriebene Kopiervorgang auf dem Prinzip der biologischen Replikation, d.h. der Verdoppelung genetischen Materials. Nach Abschluß der Rekombination ist noch kein Fortschritt im Lösungsraum erzielt worden. Der erzeugte vorläufige Nachkomme repräsentiert dieselbe Lösung wie der dritte Elternteil.

Zu dem Zweck der Rekombination zweier Mutationscodes kann auf bewährte Crossover-Operatoren aus dem Bereich der Genetischen Algorithmen zurückgegriffen werden. Hier wird der auf Davis (1991) zurückgehende Uniform Order-Based Crossover verwendet. Er wurde bereits mehrfach bei der erfolgreichen Lösung von kombinatorischen Optimierungsproblemen mit Genetischen Algorithmen eingesetzt.

Im zweiten Schritt wird der vorläufige Nachkomme in den endgültigen Nachkommen überführt, indem der ererbte Tourenplan TP_3 einer Mutation unterworfen wird. Eine Mutation umfaßt hier eine Menge von Ausfüge- und Einfügeoperationen, deren Reihenfolge über den Mutationscode gesteuert wird. Anders als bei der Evolutionsstrategie ES1 besteht der Mutationscode hier aus einem Permutationsvektor der Länge $2n$. Die Elemente des Permutationsvektors stellen Kundennummern dar. Jede Kundennummer tritt in einem Permutationsvektor genau zweimal auf. Das erste Auftreten einer Kundennummer identifiziert eine Ausfügeoperation und das zweite Auftreten eine Einfügeoperation. Für das Aus- und Einfügen eines Kunden gilt:

- Das Ausfügen eines Kunden erfolgt ausschließlich mit der Ausfügeoperation des Or-opt-Operators.

- Das Einfügen eines Kunden basiert auf folgender Einfügeheuristik: Jede Tour $T \in SV_3$ des Tourenplans SV_3 wird zunächst daraufhin überprüft, ob eine Erweiterung der Tour T um die Bedienung des einzufügenden Kunden k zulässig ist. Sofern eine Erweiterung einer oder mehrerer Touren um den Kunden k zulässig ist, wird die Einfügestelle so ausgewählt, daß die zusätzlich zu fahrende Strecke gemäß Gleichung (8) minimal ist. Scheitert jedoch das Einfügen des Kunden k in eine der Touren an den gegebenen Restriktionen, so wird dem Tourenplan eine den Kunden k bedienende Pendeltour hinzugefügt.

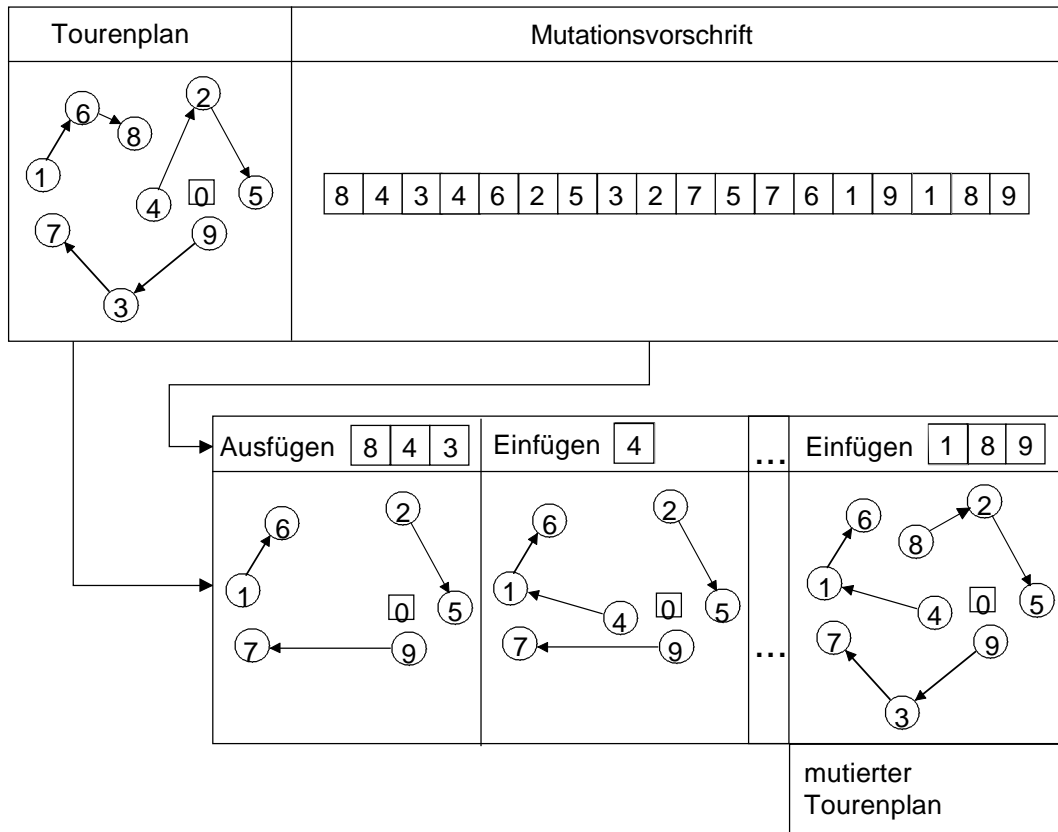


Abb. 7. Vereinfachte Darstellung der Mutation eines Tourenplans.

An die Ausführung sämtlicher mit dem Permutationsvektor gesteuerter Ausfüge- und Einfügeoperationen schließt sich gegebenenfalls noch eine weitere Operation an:

- Falls der Parameter *TourAuflösung* des erzeugten Nachkommen den Wert 1 aufweist, wird ein modifizierter Or-opt-Move in gleicher Weise wie bei der Evolutionsstrategie ES1 ausgeführt. Auch hier ist es das Ziel, die Fahrzeuganzahl durch die Auflösung einer Tour zu verringern.

Abb. 7 verdeutlicht an einem einfachen Beispiel die Steuerung von Ausfüge- und Einfügeoperationen mit einem Permutationsvektor. Der Vektor wird von links nach rechts abgearbeitet. Dabei wird geprüft ob eine Kundennummer zum ersten Mal oder bereits zum zweiten Mal auftritt. Im ersten Fall wird eine Ausfügeoperation und im zweiten Fall eine Einfügeoperation ausgeführt. So führt bei dem in Abb. 7 gezeigten Beispiel das erste Auftreten der Kundennummern 8, 4 und 3 zu dem Ausfügen dieser Kunden aus dem Tourenplan. Danach bewirkt das zweite Auftreten der Kundennummern 4 das Einfügen dieses Kunden in den Tourenplan. Zum Schluß werden die Kunden 1, 8 und 9 in den Tourenplan eingefügt.

Die im Zuge der Abarbeitung eines Permutationsrings ausgeführten Ausfüge- und Einfügeoperationen können sich – in unterschiedlichem Maße – in ihrer Wirkung aufheben. Je nach dem Umfang dieser Kompensation ergeben sich – zufällig – unterschiedlich große Änderungen des

Tourenplans bzw. Mutationsschrittweiten. Im Verfahrensablauf kommt es allerdings zu einer Konvergenz des Mutationscodes und damit zu einer Abnahme der Mutationsschrittweiten. Anfänglich werden also größere Mutationsschrittweiten und gegen Ende der Suche kleinere Schrittweiten realisiert.

3.3 Bewertungsfunktion und Selektion

Was die Bewertung und Selektion von Individuen im Fortpflanzungsprozeß betrifft, ergeben die bisherigen Ausführungen folgendes Konzept: Von λ , $\lambda > 0$, erzeugten Individuen gehen die μ , $\mu < \lambda$, Individuen mit den höchsten Fitnesswerten $F^*(SVektor)$ in die jeweiligen Folgepopulation ein. Hierbei stellt $F^*(SVektor)$ eine lexikografische Ordnungsrelation dar, welche die Fahrzeuganzahl mit erster Priorität und die Gesamtentfernung mit zweiter Priorität berücksichtigt. Ein Test dieses Konzepts hat allerdings zu unbefriedigenden Ergebnissen geführt und zwar aus zwei Gründen:

- Wird bei gegebener Fahrzeuganzahl die weitere Suchrichtung ausschließlich durch das sekundäre Kriterium – die Minimierung der Gesamtentfernung – bestimmt, so kann die Erreichung des primären Kriteriums – die Minimierung der Fahrzeuganzahl – durchaus erschwert werden. So zeigt Retzko (1995), daß in der Regel die von allen Fahrzeugen zurückgelegte Gesamtentfernung steigt, falls die Anzahl der Fahrzeuge unter einen bestimmten Wert fällt. Umgekehrt erscheint es daher plausibel, daß die Minimierung der Gesamtentfernung nicht zwangsläufig zu einer Verringerung der Fahrzeuganzahl führt.
- Die gegebene Nachbarschaftsstruktur scheint die Erreichung des primären Optimierungskriteriums nicht zu begünstigen. Einerseits werden neue Lösungen erzeugt, indem vorhandene Individuen relativ geringen Variationen unterworfen werden. Andererseits bedient eine Tour in der Regel eine größere Anzahl von Kunden. Bei der Erzeugung eines neuen Tourenplans wird es daher nur in den seltensten Fällen zu einer Verringerung der Fahrzeuganzahl kommen. Dies gilt umso mehr, als die Zeitfensterrestriktionen eine Verteilung der zu bedienenden Kunden auf eine geringere Anzahl von Touren zumindest tendenziell erschweren.

Es stellt sich daher die Frage, wie die Suche in die Richtung der Verringerung der Fahrzeuganzahl gelenkt werden kann. Zu diesem Zweck wurde ein drittes Bewertungskriterium eingeführt, welches aus zwei Kennzahlen besteht. Beide Kennzahlen beziehen sich auf die kleinste Tour T eines Tourenplans TP und schätzen auf unterschiedliche Art ab, wie leicht die Tour T im Zuge der weiteren Suche aufgelöst werden kann.

Als erste, ganz einfache Kennzahl dient die Anzahl Kd_T der Kunden, die in der Tour T bedient werden. Unterstellt wird also, daß ein kleiner Wert Kd_T die Auflösung der Tour T begünstigt.

Als zweite Kennzahl wird die hier so bezeichnete "minimale Verspätung" D_T für alle Kunden der Tour T verwendet. Die Größe D_T ergibt sich durch die Addition entsprechender kundenbezogener Werte D_k über alle Kunden k , $k \in T$, der Tour T :

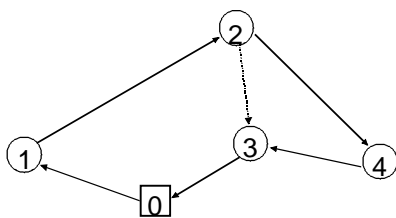
$$D_T = \sum_{k \in T} D_k . \quad (9)$$

Die Größe D_k bezeichnet hierbei die durch das Einfügen des Kunden $k \in T$ in eine andere Tour T' , $T' \neq T$, des Tourenplans bewirkte "minimale Verspätung".

Bei dem Einfügen eines Kunden k zwischen zwei aufeinanderfolgenden Orten i und $i+$ einer Tour T' sind drei Fälle zu unterscheiden:

- (1) Es existieren mehrere andere Touren, die um die Bedienung des Kunden k erweitert werden können, ohne daß es zu einer Verletzung der zu berücksichtigenden Restriktionen kommt. In diesem Fall tritt keine Verspätung auf und es gilt: $D_k = 0$.
- (2) Es existiert keine andere Tour, die um die Bedienung des Kunden k erweitert werden kann, ohne daß es zu einer Verletzung der entsprechenden Fahrzeugkapazitätsrestriktion kommt. In diesem Fall ist das Einfügen des Kunden k in eine andere Tour unmöglich und D_k wird auf unendlich gesetzt: $D_k = \infty$.
- (3) Es existieren eine oder mehrere Touren, die – unter ausschließlicher Verletzung von Zeitfensterrestriktionen – um die Bedienung des Kunden k erweitert werden können. In diesem Fall wird für alle Paare von aufeinanderfolgenden Orte ($i, i+$), zwischen die der Kunde k eingefügt werden kann, die sogenannte "Verspätung" $D_k(i, i+)$ berechnet. Die Größe $D_k(i, i+)$ setzt sich additiv aus den beiden Verschiebungszeiten $V1_k(i, i+)$ und $V2_k(i, i+)$ zusammen.

Geographische Darstellung einer Tour:



Legende:

- $V1_k(i, i+)$
- $V2_k(i, i+)$
- Zeitfenster
- Bedienungszeit
- Wartezeit
- Bedienungsreihenfolge

Zeitliche Darstellung einer Tour:

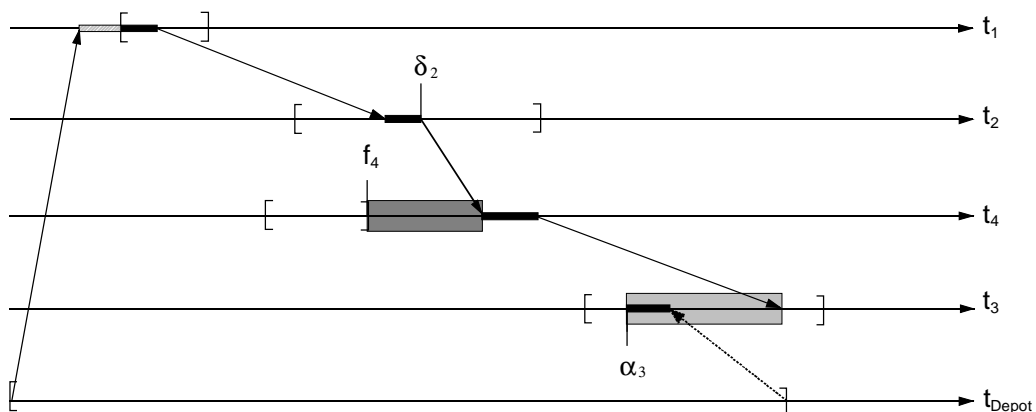


Abb. 8. Berechnung von $V1_k(i, i+)$ und $V2_k(i, i+)$.

$V1_k(i, i+)$ gibt an, um wieviele Zeiteinheiten der spätest zulässige Bedienungspunkt des eingefügten Kunden k gegebenenfalls überschritten wird. Und $V2_k(i, i+)$ gibt an, um wieviele Zeiteinheiten

- die spätest zulässige Ankunftszeit α_i des Kunden $i+$, sofern $i+$ einen Kunden bezeichnet, oder
- die spätest zulässige Ankunftszeit α_0 am Depot, sofern $i+$ das Depot bezeichnet, durch das Einfügen des Kunden k in die Tour T verzögert wird.

Im einzelnen gilt:

$$D_k(i, i+) = V1_k(i, i+) + V2_k(i, i+), \text{ wobei:} \quad (10)$$

$$V1_k(i, i+) = \max \{ \delta_i + d'_{i,k} - f_k, 0 \}, \quad (11)$$

$$V2_k(i, i+) = \max \{ \max \{ \delta_i + d'_{i,k}, e_k \} + s_k + d'_{k,i+} - \alpha_i, 0 \}. \quad (12)$$

Die Formeln zur Berechnung von $V1_k(i, i+)$ und $V2_k(i, i+)$ seien durch das in Abb. 8 dargestellte Beispiel verdeutlicht. Bei diesem Beispiel wurde der Kunde $k = 4$ zwischen die Orte $i = 2$ und $i+ = 3$ eingefügt.

Nachdem die Kunden 1 und 2 innerhalb ihres Zeitfensters bedient wurden, erreicht das Fahrzeug den eingefügten Kunden $k = 4$ außerhalb des Bedienungszeitfensters. Die Bedienung des Kunden wird – in Bezug auf den spätest zulässigen Beginn der Bedienung – um die dargestellte Verschiebungszeit $V1_4(2, 3)$ verzögert. Durch Rückwärtsrechnung ausgehend vom Depot läßt sich die späteste zulässige Ankunftszeit α_3 bei dem Kunden $i+ = 3$ für den Fall berechnen, daß das Fahrzeug gerade noch rechtzeitig zu dem Depot zurückkehrt. Kommt das Fahrzeug jedoch, wie in Abb. 8 angenommen, erst nach dem Zeitpunkt α_3 an, so ergibt sich eine positive Zeitdifferenz – die in der Abb. 8 ebenfalls dargestellte Verschiebezeit $V2_4(2, 3)$.

Unter Berücksichtigung dieser Fallunterscheidung kann nun die Vorschrift zur Ermittlung der "minimalen Verspätung" D_k für einen Kunden $k \in T$ angegeben werden. Bezeichne A_k die Menge aller Paare von Orten $(i, i+)$, die im Fall (3) zu betrachten sind, dann gilt:

$$D_k = \begin{cases} 0 & \text{für den Fall (1),} \\ \infty & \text{für den Fall (2),} \\ \min_{(i,i+) \in A_k} D_k(i, i+) & \text{für den Fall (3).} \end{cases} \quad (13)$$

Die gemäß den Beziehungen (9) bis (13) zu ermittelnde Kennzahl D_T stellt in Bezug auf die Frage, wie leicht die Tour T aufgelöst werden kann, nur einen sehr groben Schätzer dar. Einerseits werden zeitliche Interdependenzen, wie sie sich bei dem gleichzeitigen Einfügen aller Kunden der Tour T in andere Touren ergeben können, nicht berücksichtigt. Andererseits liegt der Ermittlung der Größe D_k gemäß Beziehung (13) ein optimistisches Konzept zugrunde. Plausibel wäre aber auch ein pessimistisches Konzept oder ein Erwartungswertkonzept.

In das erwähnte dritte Bewertungskriterium, im folgenden als "Auflösbarkeit der kleinsten Tour" bezeichnet, gehen die beiden Kennzahlen Kd_T und D_T im Sinne einer lexikografischen Ordnungsrelation ein. Die Bewertung eines Individuums fällt umso höher aus, je kleiner die Anzahl der Kunden Kd_T ist; bei gleicher Kundenanzahl wird ein Individuum umso höher bewertet, je geringer der Wert D_T ist.

Für die Einbeziehung des dritten Bewertungskriteriums in die – zu erweiternde – Bewertungsvorschrift $F^*(SVektor)$ bieten sich unterschiedliche Alternativen an. So kann die bisherige lexikografische Ordnungsrelation erweitert werden, indem z.B. die Auflösbarkeit der kleinsten Tour mit dritter Priorität berücksichtigt wird. Mehr Erfolg verspricht jedoch ein Vorgehen in zwei Phasen mit unterschiedlichen Suchrichtungen. So berichtet z.B. Kursawe (1991) über die erfolgreiche Anwendung einer mehrphasigen (μ, λ) -Selektion bei der Mehrzieloptimierung mit Evolutionsstrategien. In Anlehnung an Kursawe wurde ein zweiphasiges Konzept der Bewertung und Selektion realisiert:

- In der ersten Phase werden die erzeugten λ Individuen einer Generation in eine lexikografische Ordnung überführt, welche die Fahrzeuganzahl mit erster und die Auflösbarkeit der kleinsten Tour mit zweiter Priorität berücksichtigt. Die κ , $\kappa < \mu$, Individuen mit bester Bewertung werden in die Folgepopulation übernommen.
- In der zweiten Phase werden wiederum sämtliche λ Individuen einer Generation in eine lexikografische Ordnung überführt. Nach der Fahrzeuganzahl wird jetzt allerdings die Gesamtentfernung mit zweiter Priorität berücksichtigt. Die $\mu - \kappa$ Individuen mit bester Bewertung werden in die Folgepopulation übernommen.

Je nach Wahl des Quotienten κ/μ wird bei der Suche die Minimierung der Fahrzeuganzahl im Vergleich zur Entfernungsminimierung mehr oder weniger betont.

3.4 Konfigurierung

Unter Konfigurierung wird hier die Konkretisierung von Verfahrenselementen, wie der Initialisierung, der (μ, λ) -Selektion und der Terminationsbedingung verstanden. Im einzelnen wurden folgende Konkretisierungen vorgenommen:

- (1) Initialisierung der Ausgangspopulation: Sowohl für ES1, als auch für ES2 wurden die Individuen einer Ausgangspopulation mit Hilfe eines stochastischen Ansatzes erzeugt, der auf dem Savingsverfahren von Clarke und Wright (1964) basiert. Das stochastische Element besteht bei diesem Ansatz in der zufälligen Auswahl von Savingselementen aus der Savingsliste. Bei beiden Evolutionsstrategien, ES1 und ES2, wurden die Werte des Parameters *TourAuflösung* zufällig bestimmt; ebenso wurden im Fall der Strategie ES2 die Mutationscodes der Individuen der Ausgangspopulation als zufällige Permutationsvektoren bestimmt. Schließlich wurde bei der Evolutionsstrategie ES1 der Mutationscode, also die Mutations-schrittweite, zufällig auf einen Wert aus dem Intervall $[1, \dots, 10]$ festgelegt. Wie Tests ergeben haben, reagiert die Evolutionsstrategie ES1 relativ robust auf Änderungen der oberen Intervallgrenze. Jedoch führten zu klein gewählte Werte für *AnzMoves* häufiger zu einer frühzeitigen Konvergenz gegen lokale Optima.
- (2) (μ, λ) -Selektion und Populationsgröße: Für λ , μ und κ wurden die in der Tab. 1 angegebenen Werte gewählt. Dabei wurde ein Vorschlag von Schwefel (1987) berücksichtigt, der für den Quotienten μ/λ einen Wert von $\mu/\lambda \approx 1/7$ empfiehlt. Beide Evolutionsstrategien reagierten bei Tests relativ robust auf Parameteränderungen. Die Wahl relativ großer Parameterwerte bei dem Verfahren ES2 hat folgenden Grund: Bei zu klein gewählten Populationsgrößen ergibt sich eine frühzeitige Konvergenz der Permutationsvektoren und die Suche endet rasch in lokalen Optima.

Parameter	Evolutionsstrategien	
	ES1	ES2
λ	50	450
μ	8	45
κ	4	20

Tab. 1. Werte der Parameter λ , μ und κ

- (3) Terminationsbedingung: Sämtliche Berechnungen wurden nach Erreichen einer vorgegebenen Zeitschranke TL abgebrochen. In der Regel lagen jedoch Lösungen, die hinsichtlich der

Fahrzeuganzahl und der Gesamtentfernung nicht mehr verbessert werden konnten, bereits wesentlich früher vor.

4. Evaluierung

4.1 Problembeispiele

Um das Lösungsverhalten der Evolutionsstrategien für unterschiedliche Problemgrößen beurteilen zu können, wurden zwei Gruppen von Problembeispielen ausgewählt. Die Gruppen unterscheiden sich in der Anzahl von Kunden pro Problembeispiel.

Die erste Gruppe besteht aus den 56 von Solomon (1987) beschriebenen Problembeispielen, die in der Literatur üblicherweise zur Verfahrensbewertung herangezogen werden. Sämtliche Beispiele dieser Gruppe umfassen 100 Kunden. Die Kundenorte und das Depot befinden sich in einem kartesischen Koordinatensystem. Die Ortskoordinaten sind ganzzahlige Werte zwischen 0 und 100.

Die zweite Gruppe umfaßt die beiden von Russell (1995) angegebenen Beispiele D-417 und E-417, die der Praxis entnommen sind. Beide Beispiele umfassen 417 Kunden. Die Kundenorte und das Depot liegen ebenfalls in einem kartesischen Koordinatensystem. Für die Ortskoordinaten wurden ganzzahlige Werte angenommen.

Entsprechend der in der Literatur üblichen Vorgehensweise wurden die euklidischen Entfernungen d_{ij} und Fahrzeiten d'_{ij} auf zwei Nachkommastellen genau berechnet. Zur Verifikation der Zulässigkeit und der Genauigkeit wurden Endergebnisse allerdings mit der durch den Rechner festgelegten größtmöglichen Genauigkeit nachberechnet.

4.2 Verfahrensvergleich

Optimale Lösungen sind nur für einige einfache Problembeispiele bekannt. Es bietet sich daher eine Verfahrensbewertung auf der Grundlage der besten bekannten Lösungen an. Den mit den Verfahren ES1 und ES2 ermittelten Ergebnissen werden in der Literatur bzw. im Internet veröffentlichte Ergebnisse gegenübergestellt. Abschließend werden ES1 und ES2 einem separaten Vergleich unterzogen. Alle Berechnungen wurden mit einem Personalcomputer (Pentium-Prozessor, 200 MHz Taktfrequenz) durchgeführt.

Für die 56 Problembeispiele von Solomon (1987) wurde ein Vergleich mit verschiedenen, in der Literatur angegebenen Verfahren vorgenommen. Dem Vergleich liegen jeweils die besten, in multiplen Experimenten gewonnenen Lösungen zugrunde. Die Vergleichsergebnisse lassen sich wie folgt zusammenfassen:

- Sowohl mit ES1, als auch mit ES2 wurden für die fünf Problembeispiele R104, R110, R112, R207 und RC202 Lösungen ermittelt, die mit einem Fahrzeug weniger als die bisher besten bekannten Lösungen auskommen. Erkauft wurden diese Verbesserungen jedoch durch größere Gesamtentfernungen. In der Tab. 2 sind die entsprechenden Werte für die Fahrzeuganzahl NV und die Gesamtentfernung ND angegeben. Für ES1 und ES2 werden außerdem zwei Rechenzeiten angegeben: Einerseits die Zeit t_{NV} , die bis zu der Ermittlung einer Lösung mit minimaler Fahrzeuganzahl vergeht, und andererseits die Zeit t_{ND} , die bis zu der Ermittlung einer Lösung mit minimaler Gesamtentfernung verstreicht. Diese Zeiten liegen erheblich unter der Zeitschranke TL , die für die Beispiele von Solomon (1987) auf dem Wert $TL = 35$ Minuten festgelegt wurde.

Problembeispiel Solomon (1987)	Referenzverfahren	Referenzlösung		Evolutionsstrategie ES1				Evolutionsstrategie ES2			
		NV	ND	NV	ND	t_{NV}	t_{ND}	NV	ND	t_{NV}	t_{ND}
R104	Rochat und Taillard (1995)	10	982,01	9	1013,32	340	351	9	1030,78	372	1717
R110	Rochat und Taillard (1995)	11	1080,36	10	1182,49	38	88	10	1188,44	292	426
R112	Rochat und Taillard (1995)	10	953,63	9	1003,73	163	222	9	1009,04	406	517
R207	Rochat und Taillard (1995)	3	814,78	2	980,57	58	965	2	971,34	108	513
RC202	Taillard u.a. (1996)	4	1164,25	3	1398,28	19	990	3	1665,56	42	533

Tab. 2. Ergebnisvergleich für fünf Problembeispiele von Solomon (1987).

- Mit Ausnahme des Beispiels R101 wurden für die übrigen 51 Problembeispiele Lösungen erzielt, die den besten veröffentlichten Lösungen entsprechen. Eine Kumulation von Einzelergebnissen enthält die Tab. 3. Für ES1, ES2 und weitere Verfahren werden jeweils die kumulierte Fahrzeuganzahl KNV und die kumulierte Gesamtentfernung KND ausgewiesen; die Kumulation erstreckt sich hierbei über die besten Lösungen für alle 56 Problembeispiele. Angemerkt sei, daß jedes der in Tab. 3 genannten Verfahren die Minimierung der Fahrzeuganzahl als primäres Optimierungskriterium verwendet. Als sekundäres Optimierungskriterium dient bei Russell (1995), Chiang und Russell (1996), Potvin et al. (1996) sowie Potvin und Bengio (1996) die Minimierung der gesamten Fahrzeit; die Minimierung der Gesamtentfernung wird bei diesen Verfahren dagegen mit dritter Priorität gewichtet. Die übrigen Verfahren verwenden – wie ES1 und ES2 – die Minimierung der Gesamtentfernung als sekundäres Zielkriterium. Die Durchschnittswerte für die Rechenzeiten t_{NV} und t_{ND} betragen für ES1 49 und 780 Sekunden sowie für ES2 96 und 1135 Sekunden.

Referenzverfahren	Kumulierte Fahrzeuganzahl KNV	Kumulierte Gesamtentfernung KND
Homberger und Gehring ES1	406	57876
Homberger und Gehring ES2	406	58921
Taillard u.a. (1996)	410	57523
Bachem u.a. (1997)	414	61523
Rochat und Taillard (1995)	415	57231
Thangiah u.a. (1995)	418	58951
Potvin u.a. (1996)	422	62571
Chiang und Russell (1996)	422	65175
Russell (1995)	424	65827
Potvin und Bengio (1996)	426	63530
Kontoravdis und Bard (1995)	428	64196

Tab. 3. Vergleich kumulierter Ergebnisse für die 56 Problembeispiele von Solomon (1987).

Entsprechende Rechenergebnisse für die beiden Beispiele D-417 und E-417 von Russell (1995) enthält die Tab. 4. Wie die Tab. 4 zeigt, sind nur die mit dem Verfahren ES1 erzeugten besten Lösungen etwas besser, als die bislang besten bekannten Lösungen.

Referenzverfahren	Problembeispiel D-417		Problembeispiel E-417	
	NV	ND	NV	ND
Homberger und Gehring ES1	54	4703	54	4732
Rochat und Taillard (1995)	54	6265	54	7212
Thangiah u.a. (1991)	54	4866	55	4149
Homberger und Gehring ES2	54	9708	55	5174
Taillard u.a. (1996)	55	3440	55	3707
Chiang und Russell (1996)	55	4232	55	4397
Kontoravdis und Bard (1995)	55	4273	55	4986
Russell (1995)	55	4964	55	6092

Tab. 4. Ergebnisvergleich für zwei Problembeispiele von Russell (1995).

Zur Substantiierung der Qualitätsvorteile, die das Verfahren ES1 gegenüber dem Verfahren ES2 aufweist, wurde noch eine spezielle Testserie durchgeführt. Die 56 Problembeispiele von Solomon wurden mit ES1 und ES2 fünfmal berechnet; jede Problemberechnung startete dabei mit einer unterschiedlichen, stochastisch initialisierten Ausgangspopulation. In Tab. 5 werden für jedes der beiden Verfahren kumulierte Werte für die Fahrzeuganzahl und die Gesamtentfernung angegeben; die minimalen, die durchschnittlichen und die maximalen Werte von KNV und der entsprechenden Werte KND , die durch fünfmalige Berechnung der 56 Problembeispiele ermittelt wurden, werden mit KNV_{min} , KND_{min} usw. bezeichnet. Die Ergebnisse in Tab. 5 erhärten die bereits festgestellten Qualitätsvorteile des Verfahrens ES1. Die Qualitätsvorteile von ES1 sind damit zu erklären, daß keiner der verwendeten Move-Operatoren die Fahrzeuganzahl vergrößern kann. Hingegen ist es bei ES2 durchaus möglich, daß aufgrund der verwendeten Einfügeheuristik die Fahrzeuganzahl während der Suche zunimmt.

Evolutionsstrategie	Kumulierte Minimalwerte		Kumulierte Durchschnittswerte		Kumulierte Maximalwerte	
	KNV_{min}	KND_{min}	KNV_{durch}	KND_{durch}	KNV_{max}	KND_{max}
ES1	410	59704	411	58906	412	58055
ES2	413	58176	415	57949	416	57933

Tab. 5. Vergleich der Evolutionsstrategien ES1 und ES2.

Zusammenfassend kann festgestellt werden, daß die entwickelten Evolutionsstrategien anderen Heuristiken zur Lösung des Standardproblem der Tourenplanung mit Zeitfensterrestriktionen durchaus ebenbürtig sind. Vermutlich können auch andere kombinatorische Optimierungsprobleme mit Evolutionsstrategien erfolgreich bearbeitet werden.

Literaturverzeichnis

- Ablay, P. (1979) Optimieren mit Evolutionsstrategien. Doktorarbeit, Fachbereich Wirtschafts- und Sozialwissenschaften, Universität Heidelberg, Heidelberg.
- Bachem, A., Bodmann, M., Bolz, G., Emden-Weinert, T., Erdmann, A., Kiahaschemi, M., Monien, B., Prömel, H. J., Schepers, J., Schrader, R., Schulze, J. and Tschöke, S. (1997) Verbundprojekt PARALOR: Parallele Algorithmen zur Wegeoptimierung in Flugplanung und Logistik. Report No. 97-269, Beitrag zur Statustagung des BMBF HPSC 97, Angewandte Mathematik und Informatik, Universität Köln, Köln.

- Chiang, W.-C. and Russell, R. A. (1996) Simulated annealing metaheuristics for the vehicle routing problem with time windows. *Annals of Operations Research*, **63**, 3-27.
- Clarke, G. and Wright, J. W. (1964) Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research*, **12**, 568-581.
- Davis, L. (1991) *Handbook of Genetic Algorithms*. Van Nostrand Reinhold, New York.
- Domschke, W. (1990) *Logistik: Rundreisen und Touren*. Oldenbourg, München.
- Gehring, H. and Schütz, G. (1994) Zwei genetische Algorithmen zur Lösung des Bandabgleichproblems. In *Operations Research: Reflexionen aus Theorie und Praxis*, Hrsg. B. Werners and R. Gabriel, S. 85-128. Springer, Berlin.
- Glover, F. (1991) Multilevel tabu search and embedded search neighborhoods for the traveling salesman problem. Working paper, Graduate School of Business and Administration, University of Colorado, Boulder, Colorado.
- Hoffmeister, F. and Bäck, T. (1992) Genetic algorithms and evolution strategies: similarities and differences. Technical report SYS-1/92, Fachbereich Informatik, Universität Dortmund, Dortmund.
- Kontoravdis, G. and Bard, J. F. (1995) A GRASP for the vehicle routing problem with time windows. *ORSA Journal on Computing*, **7**, 10-23.
- Kursawe, F. (1991) A variant of evolution strategies for vector optimization. In *Parallel Problem Solving from Nature*, Hrsg. H.-P. Schwefel and R. Männer, S. 193-197. Springer, Berlin.
- Lenstra, J. and Rinnooy Kan, A. (1981) Complexity of vehicle routing and scheduling problems. *Networks*, **11**, 221-227.
- Nissen, V. (1994) *Evolutionäre Algorithmen*. Deutscher Universitäts-Verlag, Wiesbaden.
- Oliver, I. M., Smith, D. J. and Holland, J. R. C. (1987) A study of permutation crossover operators on the traveling salesman problem. In *Genetic Algorithms and their Applications*, Hrsg. J. J. Grefenstette, S. 224-225. Lawrence Erlbaum Associates, Hillsdale/NJ.
- Or, I. (1976) Traveling salesman-type combinatorial problems and their relation to the logistics of blood banking. Ph.D. thesis, Department of Industrial Engineering and Management Science, Northwestern University, Evanston, IL.
- Osman, I. H. (1993) Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. *Annals of Operations Research*, **41**, 421-451.
- Osman, I. H. (1995) An introduction to meta-heuristics. In *Operational Research Tutorial Papers*, Hrsg. M. Lawrence and C. Wilson, S. 92-122. Operational Research Society Press, Birmingham.
- Potvin, J.-Y. and Rousseau, J. M. (1995) An exchange heuristic for routing problems with time windows. *Journal of the Operational Research Society*, **46**, 1433-1446.
- Potvin, J.-Y., Kervahut, T., Garcia, B.-L. and Rousseau, J.-M. (1996) The vehicle routing problem with time windows - part I: tabu search. *INFORMS Journal on Computing*, **8**, 158-164.
- Potvin, J.-Y. and Bengio, S. (1996) The vehicle routing problem with time windows - part II: genetic search. *INFORMS Journal on Computing*, **8**, 165-172.
- Rechenberg, I. (1973) *Evolutionsstrategie*. Fromman-Holzboog, Stuttgart.
- Retzko, R. (1995) *Flexible Tourenplanung mit selbstorganisierenden Neuronalen Netzen*. Unitext, Göttingen.

- Rochat, Y. and Taillard, E. D. (1995) Probabilistic diversification and intensification in local search for vehicle routing. Technical report CRT-95-13, Centre de recherche sur les transports, Université de Montréal, Montréal.
- Russell, R. A. (1995) Hybrid heuristics for the vehicle routing problem with time windows. *Transportation Science*, **29**, 156-166.
- Schwefel, H.-P. (1977) *Numerische Optimierung von Computer-Modellen mittels der Evolutionsstrategie*. Birkhäuser, Basel.
- Schwefel, H.-P. (1987) Collective phenomena in evolutionary systems. Interne Berichte und Skripten Nr. 233, Fachbereich Informatik, Universität Dortmund, Dortmund.
- Solomon, M. M. (1987) Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research*, **35**, 254-265.
- Solomon, M. M., Baker, E. K. and Schaffer, J. R. (1988) Vehicle routing and scheduling problems with time window constraints: efficient implementations of solution improvement procedures. In *Vehicle Routing: Methods and Studies*, Hrsg. B. L. Golden and A. A. Assad, S. 85-105. North-Holland, Amsterdam.
- Taillard, E., Badeau, P., Gendreau, M., Guertin, F. and Potvin, J.-Y. (1996) A tabu search heuristic for the vehicle routing problem with soft time windows. Technical report CRT-95-66, Centre de recherche sur les transports, Université de Montréal, Montréal.
- Thangiah, S. R., Nygard, K. E. and Juell, P. L. (1991) GIDEON: a genetic algorithm system for vehicle routing with time windows. In *Proceedings of the 7th Conference on Artificial Intelligence for Applications*, S. 322-328. IEEE Press, Miami, FL.