



Lehrstuhl für Verkehrstechnik
Technische Universität München
Univ.-Prof. Dr.-Ing. Fritz Busch

Schriftenreihe Heft 10

Robert Braun

Ein echtzeitfähiger Evolutionärer Algorithmus zur
netzweiten Optimierung der Lichtsignalsteuerung

München 2008

Die Deutsche Bibliothek – CIP Einheitsaufnahme

Braun, Robert:

Ein echtzeitfähiger Evolutionärer Algorithmus zur
netzweiten Optimierung der Lichtsignalsteuerung

Hrsg.: Fritz Busch, München, 2009

Schriftenreihe des Lehrstuhls für Verkehrstechnik
der Technischen Universität München, Heft 10

Zugl.: München, Techn. Univ., Diss., 2008

ISBN 978-3-937631-10-3

Copyright ©

Lehrstuhl für Verkehrstechnik der Technischen Universität München 2009

Alle Rechte, auch das des auszugsweisen Nachdruckes, der auszugsweisen oder
vollständigen Wiedergabe, der Speicherung in Datenverarbeitungsanlagen und der
Übersetzung, vorbehalten.

Druck: TypeSet GmbH, Ismaning

ISBN 978-3-937631-10-3

ISSN 1612-9431

Lehrstuhl für Verkehrstechnik · Institut für Verkehrswesen

Technische Universität München · 80333 München

Telefon: 089 / 289 – 22438 · Telefax: 089 / 289 – 22333 · E-Mail: info@vt.bv.tum.de

www.vt.bv.tum.de

Bei der vorliegenden Veröffentlichung handelt es sich um die Dissertation von
Herrn Dr.-Ing. Robert Braun

Vorsitzender: Univ.-Prof. Dr.-Ing. Stephan Freudenstein, TU München

Prüfer der Dissertation:

1. Univ.-Prof. Dr.-Ing. Fritz Busch, TU München
2. Univ.-Prof. Dr.-Ing. Bernhard Friedrich,
TU Braunschweig

Danksagung

Diese Dissertation entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Lehrstuhl für Verkehrstechnik der Technischen Universität München.

Herzlich bedanke ich mich bei meinem Doktorvater Herrn Professor Fritz Busch für die Möglichkeit zur Promotion und die wertvollen fachlichen Anregungen.

Ein herzlicher Dank gilt auch Herrn Professor Bernhard Friedrich für die Übernahme des Koreferats.

Wesentliche Teile der Arbeit entstanden im Rahmen des Forschungsprojekts TRAVOLUTION. Bei allen Beteiligten bedanke ich mich für die hervorragende Zusammenarbeit. Ein besonderer Dank gilt dabei Herrn Carsten Kemper von der GEVAS software GmbH, der den Evolutionären Algorithmus umgesetzt hat. Dem Freistaat Bayern danke ich für die Förderung des Forschungsprojekts.

Nicht zuletzt bedanke ich mich bei meinen Kollegen vom Lehrstuhl für Verkehrstechnik für die großartige TUM-VT-Atmosphäre, die Unterstützung und den wertvollen fachlichen Austausch.

Robert Braun

Kurzfassung

In der vorliegenden Arbeit wird mit GALOP ein echtzeitfähiger Evolutionärer Algorithmus zur netzweiten Optimierung der Lichtsignalsteuerung entwickelt und bewertet, der zur Verbesserung des Verkehrsablaufs des motorisierten Verkehrs beiträgt. *Echtzeitfähig* heißt in diesem Zusammenhang, dass das Optimierungsverfahren in der Lage ist, zeitnah, d.h. innerhalb einer vorgegebenen Zeit von z.B. fünf Minuten, ein möglichst gutes Ergebnis zu liefern. GALOP optimiert die Signalpläne aller Lichtsignalanlagen (LSA) des jeweiligen Netzes, wobei alle Steuerungsparameter gleichzeitig optimiert werden können. Daraus können Rahmensignalpläne generiert werden, die dann einer lokalen verkehrsabhängigen Steuerung zur Verfügung gestellt werden können.

Grundannahme ist eine einheitliche Umlaufzeit für das zu optimierende Netz. Es handelt sich also um einen klassischen *zentralen* Ansatz. Vorausgesetzt werden vorhandene Signalpläne mit vorab geplanten möglichen Phasenfolgen (phasenorientierter Ansatz) für die einzelnen Knotenpunkte.

Das Verfahren wird so konzipiert, dass es tatsächlich *auf die Straße* gebracht werden kann, weil die Randbedingungen der in Deutschland weit verbreiteten lokalen verkehrsabhängigen Steuerung berücksichtigt werden.

Das Optimierungsverfahren greift auf das mesoskopische BALANCE-Verkehrsmodell zurück, das als Wirkungskenngrößen für jede Signalgruppe Wartezeiten, Zahl der Halte und Rückstaulängen für den Kraftfahrzeugverkehr liefert. Diese können in einer Zielfunktion zusammengefasst und beliebig gewichtet werden. D.h. die Gewichte können auch so gewählt werden, dass der Zielfunktionswert Kraftstoffverbrauch oder Emissionen widerspiegelt.

Die Umsetzung von GALOP fand im Rahmen des Forschungsprojekts TRAVOLUTION statt. Der bisher in der adaptiven Netzsteuerung BALANCE verwendete Hill-Climbing Algorithmus wurde durch GALOP ersetzt. Der Hill-Climbing Algorithmus wurde als eine Referenz für die *Bewertung* des umgesetzten Evolutionären Algorithmus verwendet. Zur Bewertung wurden insgesamt zwölf Fälle aus drei Netzen und vier Verkehrsnachfrageszenarien herangezogen.

Je größer und komplexer das Netz wird, desto mehr kommen die Vorteile eines Evolutionären Algorithmus wie insbesondere die gleichzeitige Optimierung aller Steuerungsparameter zum Tragen. Stärke und Ausprägung der Verkehrsnachfrage scheinen dabei keine Rolle für die Algorithmen zu spielen. Für das mit 27 LSA größte der drei untersuchten Netze, ist der Gütewert der durch GALOP gefundenen Lösungen im Vergleich zu den Lösungen des Hill-Climbing Algorithmus je nach Nachfrageszenario um 23% bis 36% besser.

Die Sensitivität des Algorithmus gegenüber den Einstellungen seiner Parameter ist vor allem bei den kleineren Netzen sehr gering. Beim größten Netz verhält sich GALOP etwas sensiti-

ver, liefert allerdings selbst bei ungünstigen Einstellungen ein besseres Ergebnis als der Hill-Climbing Algorithmus. Der Evolutionäre Algorithmus sollte deshalb insbesondere bei komplexeren Netzen ausreichend kalibriert werden, um möglichst gute Lösungen zu finden. Durch das Einlesen guter Lösungen aus früheren Optimierungsläufen kann der Algorithmus noch robuster gemacht werden, weil von Anfang an relativ gutes Genmaterial vorhanden ist.

Im Rahmen des Projekts TRAVOLUTION wurde GALOP in Ingolstadt implementiert. Bei den durchgeführten Feldversuchen wurden drei Steuerungsvarianten miteinander verglichen: das Basisszenario als Vorher-Fall ohne adaptive Netzsteuerung, BALANCE mit Hill-Climbing Algorithmus und BALANCE mit GALOP. In allen drei Fällen besteht eine lokale verkehrabhängige Steuerung, in die eine ÖPNV-Priorisierung integriert ist. Außerdem sind einige reine Fußgängerschutzanlagen im Testgebiet, die nur auf Anforderung per Taster reagieren. Die Qualität der Steuerungen wurde durch Befahrungen und zum Teil durch Reisezeitmessung mittels Fahrzeugwiedererkennung bewertet. Die Auswertungen ergaben, dass die Verlustzeiten durch BALANCE mit GALOP im Vergleich zum Basisszenario im Tagesdurchschnitt um insgesamt 21% reduziert wurden. Die Verbesserung durch GALOP im Vergleich zum Hill-Climbing Algorithmus liegt bei 10%.

Inhaltsverzeichnis

1	Einleitung.....	1
1.1	Kontext.....	1
1.2	Zielsetzung und Eingrenzung.....	1
1.3	Inhalt und Aufbau.....	4
2	Stand der Technik	7
2.1	Lichtsignalsteuerung.....	7
2.1.1	Grundlagen der Lichtsignalsteuerung.....	7
2.1.2	Verkehrsabhängige Steuerung.....	8
2.2	Optimierung	10
2.2.1	Begriffe.....	10
2.2.2	Bekannte Optimierungsverfahren	11
2.2.3	Eignung der Verfahren für die Problemstellung	16
2.3	Bekannte modellbasierte Online-Netzsteuerungsverfahren	17
2.4	Evolutionäre Algorithmen in der netzweiten LSA-Steuerung.....	21
2.5	Zwischenfazit	23
3	Gesamtsystem	25
3.1	Regelkreis und Ablauf der Optimierung.....	25
3.2	Verkehrsmodell	26
3.3	Zielfunktion.....	27
3.4	Optimierungsverfahren	29
4	Evolutionärer Algorithmus GALOP	31
4.1	Allgemeines zu Evolutionären Algorithmen.....	31
4.2	Entwurfsmethodik.....	33
4.3	Kodierung der Steuerungsparameter.....	35
4.3.1	Problemspezifische Randbedingungen	35
4.3.2	Repräsentation der Steuerungsparameter im Individuum	37
4.3.3	Umlaufzeit	38
4.3.4	Versatzzeit.....	38
4.3.5	Phasenfolge	39
4.3.6	Phasendauern bzw. Startzeitpunkte der Phasenübergänge.....	40
4.4	Startpopulation	44
4.5	Operatoren	45
4.5.1	Selektion	45
4.5.2	Rekombination.....	47
4.5.3	Mutation.....	48
4.5.4	Parameteradaption	49
4.6	Verjüngung	50

4.7	Abbruchkriterien	50
4.8	Zwischenfazit	51
5	Bewertung des Optimierungsverfahrens GALOP	53
5.1	Umsetzung des Algorithmus	53
5.2	Bewertungsmethodik	54
5.2.1	Bewertungskriterien	55
5.2.2	Bewertungsmetriken	56
5.3	Versuchsaufbau und Vorgehensweise	57
5.4	Güteverteilung im Suchraum.....	59
5.5	Einfluss der Startzufallszahl (Konsistenz)	60
5.6	Auswahl einer Grundvariante	62
5.7	Sensitivitätsanalyse	65
5.7.1	Rekombinationsvarianten.....	66
5.7.2	Rekombinations- und Mutationswahrscheinlichkeit	68
5.7.3	Anzahl der Nachkommen	70
5.7.4	Startmutationsschrittweite	74
5.7.5	Fazit der Sensitivitätsanalyse	76
5.8	Bewertung der Qualität und Effizienz	76
5.9	Fazit der Bewertung	78
6	Anwendung des Verfahrens	81
6.1	Untersuchungsgebiet	81
6.2	Verkehrsstärkeganglinien	82
6.3	Messmethodik.....	85
6.4	Ergebnisse.....	85
6.4.1	Reisezeitmessung durch Fahrzeugwiedererkennung	86
6.4.2	Befahrungen.....	88
6.4.3	Kombinierte Ergebnisse für Verlustzeit	95
6.5	Fazit aus den Feldversuchsergebnissen	98
7	Zusammenfassung und Ausblick	99
7.1	Zusammenfassung der Arbeit	99
7.2	Erweiterungsmöglichkeiten und Ausblick	101
	Literatur.....	105
	Glossar.....	111
	Abbildungsverzeichnis.....	113
	Tabellenverzeichnis	115

1 Einleitung

1.1 Kontext

Der Verkehr in städtischen Gebieten ist meist durch eine große Zahl von Verkehrsteilnehmern auf engem Raum gekennzeichnet. Dabei stoßen unterschiedliche Modi aufeinander. Neben dem wegen seines Platzbedarfs und seiner Geschwindigkeit häufig dominanten motorisierten Individualverkehr (Pkw, Lkw, Krad) und dem ebenfalls motorisierten öffentlichen Personennahverkehr (Straßenbahn, Bus) wird der Straßenraum auch vom nichtmotorisierten Individualverkehr (Radfahrer, Fußgänger usw.) benutzt.

Der städtische Verkehr wird geprägt durch die zahlreichen Knotenpunkte, die Steuerung erfolgt meist durch Lichtsignalanlagen. Die Lichtsignalsteuerung hat insbesondere auf den Verkehrsablauf des motorisierten Verkehrs sehr großen Einfluss, da die Wartezeiten an Lichtsignalanlagen einen verhältnismäßig großen Anteil der Gesamtreisezeit ausmachen.

Die einzelnen Verkehrsteilnehmer haben meist die gleichen Ziele. Jeder möchte seinen Weg mit möglichst geringen Verlustzeiten und einer möglichst kleinen Anzahl an Halten zurücklegen können (Nutzeroptimum). Im Sinn eines Systemoptimums, das durch die Lichtsignalsteuerung erreicht werden soll, treten jedoch Zielkonflikte auf. Um die Wartezeiten beispielsweise für Fußgänger möglichst gering zu halten, müssen die Umlaufzeiten möglichst kurz sein. Demgegenüber sollten die Umlaufzeiten bei hoher Verkehrsbelastung möglichst lang sein, um die Verluste durch die Zwischenzeiten möglichst gering zu halten. Verfolgt man das Ziel, Schadstoff- und Lärmemissionen zu verringern, ohne gleichzeitig Zufahrtsbeschränkungen zu beschließen, so läuft das auf eine Reduktion der Zahl der Halte des motorisierten Verkehrs hinaus.

1.2 Zielsetzung und Eingrenzung

Ziel dieser Arbeit ist die Entwicklung eines echtzeitfähigen Optimierungsverfahrens für die netzweite Lichtsignalsteuerung, das zur Verbesserung des Verkehrsablaufs des motorisierten Verkehrs beiträgt. *Echtzeitfähig* heißt in diesem Zusammenhang, dass das Optimierungsverfahren in der Lage sein muss, zeitnah, d.h. innerhalb einer vorgegebenen Zeit von z.B. fünf Minuten, ein möglichst gutes Ergebnis zu liefern.

Die netzweite Optimierung der Lichtsignalsteuerung stellt ein komplexes Problem mit einer sehr hohen Anzahl von Freiheitsgraden in Form von Steuerungsparametern dar. MERTZ [2001] hat gezeigt, dass es zur Klasse der NP-vollständigen Probleme gehört. Da keine geschlossene mathematische Formulierung möglich ist, ist es analytisch nicht zu lösen.

Zur Verdeutlichung der Komplexität des gegebenen Problems zeigt Abb. 1.1 zwei Dimensionen eines 20-dimensionalen Lösungsraums. Das zu Grunde liegende Netz besteht aus 5 Knotenpunkten mit Lichtsignalanlage (LSA). Dargestellt ist das Ergebnis der Zielfunktion (Wartezeitensumme) in Abhängigkeit der möglichen Werte der Versatzzeiten (Var 1 und Var 2) von zwei der fünf Knotenpunkte. Die restlichen Variablen (Umlaufzeit, Phasenfolgen und Phasendauern) sind konstant gehalten.

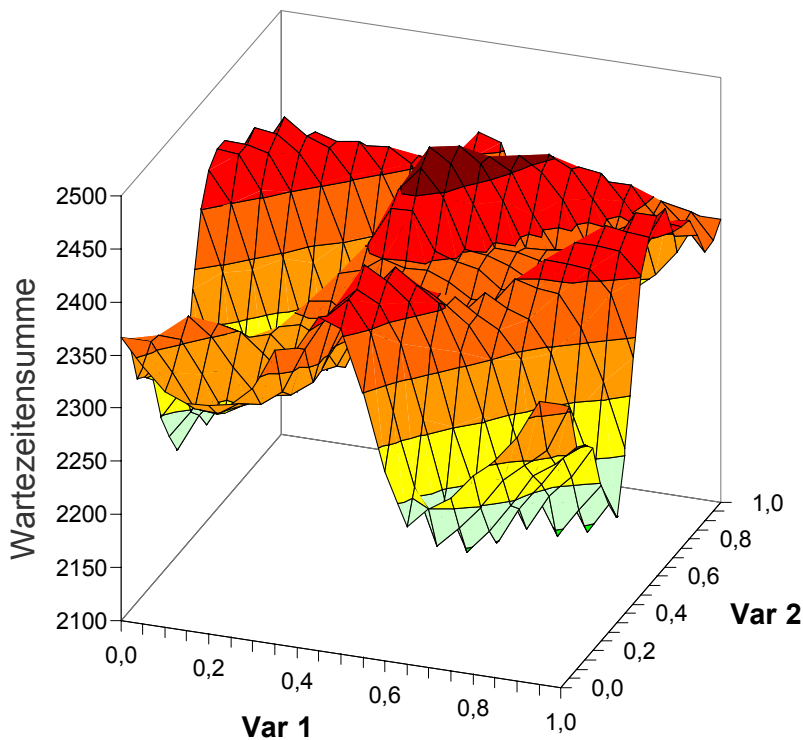


Abb. 1.1 Zwei Dimensionen eines 20-dimensionalen Lösungsraums

Bereits anhand der dargestellten zwei Dimensionen des 20-dimensionalen Lösungsraums lassen sich die zahlreichen lokalen Minima erahnen. Die Größe des Lösungsraums nimmt mit der Anzahl der LSA exponentiell zu, was folgende Überlegungen zeigen.

Für eine LSA ergeben sich pro Phasenfolge m mögliche Freigabezeitverteilungen:

$$m = \prod_{i=1}^{(k_{\sigma}-1)} \left(\frac{t_{F,v} + k_{\sigma} - i}{k_{\sigma} - i} \right)$$

mit

$t_{F,v}$ für die Optimierung verfügbare Freigabezeit

k_{σ} Anzahl der Phasen in Phasenfolge σ

Das größte der in dieser Arbeit verwendeten Netze hat 27 Knotenpunkte mit LSA. Es gibt durchschnittlich 2 mögliche Phasenfolgen pro LSA mit durchschnittlich 4 Phasen und einer durchschnittlich für die Optimierung verfügbaren Freigabezeit von 45 s. Bei einer Umlaufzeit von 90 s ergeben sich daraus über 17 000 mögliche Freigabezeitverteilungen. Setzt man für die Versatzzeit 10 mögliche Werte an (beschränkte Optimierung der Versatzzeit), dann ergibt sich für die Größe des Suchraums folgende Tabelle:

LSA	Lösungen	Rechenzeit [a]
1	345 920	0,0001
2	$1 \cdot 10^{11}$	37,9
3	$4 \cdot 10^{16}$	$1 \cdot 10^{07}$
4	$1 \cdot 10^{22}$	$5 \cdot 10^{12}$
5	$5 \cdot 10^{27}$	$2 \cdot 10^{18}$
10	$2 \cdot 10^{55}$	$8 \cdot 10^{45}$
20	$6 \cdot 10^{110}$	$2 \cdot 10^{101}$
27	$4 \cdot 10^{149}$	$1 \cdot 10^{140}$
30	$1 \cdot 10^{166}$	$5 \cdot 10^{156}$

Tab. 1.1 Größe des Lösungsraums

Unter der für heutige Rechner realistischen Annahme, dass die Bewertung einer Lösung 0,01 s benötigt, würde die Berechnung aller möglichen Lösungen bereits bei 2 LSA fast 38 Jahre dauern. Für 27 LSA existieren demnach $4 \cdot 10^{149}$ mögliche Lösungen. Zum Vergleich: die Anzahl der Atome im sichtbaren Universum wird auf etwa 10^{80} geschätzt.

Die Zielsetzung besteht darin, ein geeignetes Optimierungsverfahren zu entwickeln, das in der Lage ist, eine möglichst gute Lösung in relativ kurzer Zeit zu finden. Grundannahme ist eine einheitliche Umlaufzeit für das zu optimierende Netz. Es handelt sich also um einen klassischen *zentralen* Ansatz und keine *dezentrale* Optimierung wie sie beispielsweise von LÄMMER [2007] vorgeschlagen wird.

Vorausgesetzt werden bereits vorhandene Signalpläne mit vorab geplanten möglichen Phasenfolgen (*phasenorientierter Ansatz*) für die einzelnen Knotenpunkte. Es sollen also keine Signalpläne für einzelne Knotenpunkte gebildet, sondern vorhandene Signalpläne optimiert werden.

Es wird ein echtzeitfähiger Evolutionärer Algorithmus entwickelt, der die Signalpläne aller Lichtsignalanlagen des Netzes optimiert. Ein wesentlicher Vorteil dieser Optimierungsmethode besteht darin, dass alle Steuerungsparameter gleichzeitig optimiert werden können, was mit einigen anderen Optimierungsverfahren, wie beispielsweise dem häufig verwendeten Hill-Climbing Algorithmus, nicht möglich ist. Dadurch ist die Gefahr, in einem lokalen

Optimum hängen zu bleiben, deutlich geringer. Aus den optimierten Signalplänen können Rahmensignalpläne generiert werden, die beispielsweise alle fünf Minuten der lokalen verkehrsabhängigen Steuerung zur Verfügung gestellt werden können.

Das Optimierungsverfahren greift auf ein mesoskopisches Verkehrsmodell zurück, das als Wirkungskenngröße für jede Signalgruppe Wartezeiten, Zahl der Halte und Rückstaulängen für den Kraftfahrzeugverkehr liefert. Diese können in einer Zielfunktion zusammengefasst und beliebig gewichtet werden.

Die direkte Integration der Ziele der Fußgänger und Radfahrer sowie des Öffentlichen Personennahverkehrs (ÖPNV) in die Optimierung wird in dieser Arbeit nicht vorgenommen. Der ÖPNV kann unabhängig von den zentralen Vorgaben des Optimierungsverfahrens durch eine lokale verkehrsabhängige Steuerung sekundengenau priorisiert werden, wie es dem Stand der Technik entspricht. Die Wartezeiten für Fußgänger und Radfahrer können durch die Vorgabe einer (maximalen) Umlaufzeit im Rahmen gehalten werden.

Das Verfahren wird so konzipiert, dass es tatsächlich *auf die Straße* gebracht werden kann, weil die Randbedingungen der in Deutschland weit verbreiteten lokalen verkehrsabhängigen Steuerung berücksichtigt werden.

1.3 Inhalt und Aufbau

Die Arbeit ist in sieben Kapitel gegliedert, beginnend mit dieser Einleitung.

In *Kapitel 2* folgen Beschreibung und Diskussion des Stands der Technik im Bereich der Lichtsignalsteuerung und der Optimierung. Als geeignete Optimierungsmethode für die gegebene Problemstellung werden Evolutionäre Algorithmen ausgewählt. Außerdem finden sich in Kapitel 2 Beschreibungen bekannter modellbasierter Online-Netzsteuerungsverfahren und Beispiele, die den Einsatz Evolutionärer Algorithmen zur Optimierung der netzweiten Lichtsignalsteuerung schildern.

In *Kapitel 3* wird der Gesamttablauf der Optimierung als Regelkreis erläutert. Die wesentlichen Komponenten der Optimierung sind ein Verkehrsmodell, eine Zielfunktion und das in dieser Arbeit zu entwickelnde Optimierungsverfahren.

Beim Entwurf eines Evolutionären Algorithmus handelt es sich um einen iterativen Prozess. Bereits während der Entwicklung des Algorithmus ist eine ständige Bewertung erforderlich. Entwicklung und Bewertung werden in den Kapiteln 4 und 5 abgehandelt.

In *Kapitel 4* folgt nach einem einführenden Abschnitt zu den Grundlagen Evolutionärer Algorithmen zunächst die verwendete Entwurfsmethodik für die Entwicklung des Algorithmus. In den weiteren Abschnitten wird der echtzeitfähige Genetische Algorithmus zur netzweiten

Optimierung der Lichtsignalsteuerung GALOP dargestellt, der im Rahmen dieser Arbeit entwickelt wird. Anschließend wird der entscheidende Teil der Entwicklung erläutert: die Kodierung der Steuerungsparameter. In den weiteren Abschnitten von Kapitel 4 werden die weiteren Bestandteile des Algorithmus, u.a. die evolutionären Operatoren des Algorithmus, beschrieben, gefolgt von einem Zwischenfazit.

In *Kapitel 5* erfolgt die Bewertung des Optimierungsverfahrens. Zunächst wird die Umsetzung des Algorithmus erläutert. Anschließend werden die verwendete Bewertungsmethodik sowie der Versuchsaufbau und die Vorgehensweise bei der Bewertung dargestellt. Vor der eigentlichen Bewertung wird die Güteverteilung im Suchraum untersucht. Zum Abschluss von Kapitel 5 wird ein Fazit der Bewertung gezogen.

GALOP kommt in Ingolstadt zum Einsatz, die Feldversuchsergebnisse werden in *Kapitel 6* vorgestellt. Dabei wird zunächst das Untersuchungsgebiet dargestellt, anschließend werden die Verkehrsstärkeganglinien der Messtage verglichen. Nach der Erläuterung der verwendeten Messmethodik werden die Ergebnisse der Feldversuche dargestellt.

In *Kapitel 7* wird die vorliegende Arbeit zusammengefasst. Zum Abschluss der Arbeit werden einige Erweiterungsmöglichkeiten und Lösungsansätze diskutiert. Dazu werden im Verlauf der Arbeit nicht umgesetzte oder betrachtete Teilaspekte und festgestellte offene Punkte oder Problemstellungen nochmals aufgegriffen.

2 Stand der Technik

Zu Beginn der Arbeit wird der Stand der Technik im Bereich der Lichtsignalsteuerung (Abschnitt 2.1) und der Optimierung (Abschnitt 2.2) dargestellt. Nach der Beschreibung bekannter Optimierungsverfahren wird deren Eignung für die gegebene Problemstellung diskutiert und evolutionäre Algorithmen als geeignete Methode ausgewählt. In Abschnitt 2.3 werden bekannte modellbasierte Online-Netzsteuerungsverfahren beschrieben. Evolutionäre Algorithmen kamen bereits häufiger zur Optimierung der netzweiten Lichtsignalsteuerung zum Einsatz, in der Literatur gefundene Beispiele werden in Abschnitt 2.4 dargestellt.

2.1 Lichtsignalsteuerung

2.1.1 Grundlagen der Lichtsignalsteuerung

Die grundlegenden verkehrstechnischen Bestimmungen für die Einrichtung und den Betrieb von Lichtsignalanlagen in Deutschland sind in den *Richtlinien für Lichtsignalanlagen RiLSA* [FGSV, 1998] und im *Handbuch für die Bemessung von Straßenverkehrsanlagen HBS* [FGSV, 2005] enthalten. Die Forschungsgesellschaft für Straßen- und Verkehrswesen (FGSV) arbeitet derzeit an einer Neufassung der RiLSA, die Veröffentlichung steht noch aus.

Die Lichtsignalsteuerung lässt sich zum einen nach dem *räumlichen Einsatzbereich* gliedern in:

- Punktsteuerung (einzelner Knotenpunkt mit LSA)
- Liniensteuerung (Grüne Welle)
- Netzsteuerung (Netzkoordinierung von LSA).

Zum anderen gliedert sie sich nach dem Grad der Verkehrsabhängigkeit:

- Festzeitsteuerung, meist tageszeitabhängig
- Verkehrsabhängige Steuerung

Grundlage aller Varianten ist die Steuerung des einzelnen Knotenpunkts.

Planung eines Signalplans

Die Planung eines Festzeit-Signalplans für einen einzelnen Knoten kann auf zwei Arten erfolgen:

- Sperrgruppenorientiert
- Phasenorientiert

Der *sperrgruppenorientierte* Ansatz geht von den einzelnen Signalgruppen aus. Eine Signalgruppe enthält alle Signalgeber, die zu jeder Zeit das gleiche Signalbild zeigen. Eine Sperrgruppe enthält alle Verkehrsströme, die in einem Umlauf separat freigegeben werden müssen, da sie zueinander jeweils unverträglich sind. Die Zusammenstellung der Sperrgruppen erfolgt beim Verfahren *Addition kritischer Fahrzeugstrombelastungen* (AKF-Schema) nach einer bestimmten Systematik: die maßgebende Sperrgruppe ergibt sich durch die Addition der maßgebenden Belastung der zugehörigen Verkehrsströme. Mit der maßgebenden Sperrgruppe lässt sich schließlich ein Signalplan erstellen.

Der *phasenorientierte* Ansatz geht von festzulegenden Phasen aus, innerhalb derer einer oder mehrere Verkehrsströme freigegeben werden. Innerhalb einer Phase bleiben alle Signalbilder konstant. Die einzelnen Phasen werden zu einem Signalplan zusammengestellt, wobei zwischen den Phasen die Phasenübergänge dafür sorgen, dass Zwischenzeiten und ggf. Mindestgrünzeiten eingehalten werden.

Mit beiden Varianten erhält man am Ende einen Signalplan, der sich in Phasen und Phasenübergänge unterteilen lässt. Mit den vorgegebenen Phasen lässt sich ein Signalplan über folgende Steuerungsparameter eindeutig beschreiben:

- Umlaufzeit
- Versatzzeit
- Phasenfolge
- Phasendauern

Innerhalb der *Umlaufzeit* läuft ein Signalplan einmal komplett ab. Die *Versatzzeit* ist der zeitliche Versatz des Referenzpunkts des lokalen Signalplans zur globalen Zeitskala. Sie ist eine wesentliche Stellgröße für die Koordinierungsstruktur einer Strecke bzw. eines Netzes. Die *Phasenfolge* gibt an, welche Phasen in welcher Reihenfolge geschaltet werden. Ein *Signalprogramm* enthält häufig mehrere Signalpläne mit unterschiedlichen Phasenfolgen. Die *Phasendauern* geben die Dauer einer Phase an. Dadurch sind gleichzeitig auch die Freigabezeiten festgelegt.

2.1.2 Verkehrsabhängige Steuerung

In der *Teilfortschreibung der RiLSA* [FGSV, 2003] werden verkehrsabhängige Steuerungsverfahren nach zwei unterschiedlichen Ansätzen unterschieden:

- Messwertbasiert
- Modellbasiert

Bei einem *messwertbasierten* Steuerungsverfahren werden Steuerungsentscheidungen in einer Ablauflogik durch die Abfrage bestimmter logischer und zeitlicher Bedingungen unmittelbar abgeleitet. Es findet online keine Bewertung und Optimierung statt.

Bei einem *modellbasierten* Steuerungsverfahren werden die Freiheitsgrade durch ein Steuerungsmodell beschrieben, die Modellierung der Verkehrssituation und der daraus resultierenden Wirkungen geschieht online. Die Entscheidung über die Steuerung wird nach einer Zielgrößenoptimierung getroffen.

Verkehrsabhängige Knotenpunktsteuerung

Am einzelnen Knotenpunkt können gemäß RiLSA [FGSV, 1998] folgende Anpassungen in Abhängigkeit von der Verkehrssituation vorgenommen werden:

- Freigabezeitanpassung
- Phasentausch
- Bedarfsphasenanforderung
- Signalprogrammabbildung

Häufigstes messwertbasiertes Verfahren zur *Freigabezeitanpassung* ist die sog. Zeitlückensteuerung. Dabei werden durch einen Detektor die zeitlichen Abstände (Nettozeitlücken) aufeinanderfolgender Fahrzeuge eines Verkehrsstroms gemessen. Die Freigabezeit wird nach Ablauf einer vorgegebenen Mindestdauer so lange verlängert, bis die gemessene Zeitlücke einen festgelegten Grenzwert überschreitet. Eine maximale Freigabezeit stellt sicher, dass bei starkem Verkehr auch die Querrichtung wieder zum Zug kommt. Es gibt also einen zeitlichen Rahmen, innerhalb dessen eine Phase vorzeitig abgebrochen werden darf (siehe auch Abschnitt 4.3.1).

Zur Minimierung der Verlustzeiten des öffentlichen Verkehrs werden in steigendem Umfang messwertbasierte Vorrangschaltungen an LSA eingesetzt (ÖPNV-Priorisierung), d.h. ein ankommendes ÖPNV-Fahrzeug fordert eine Freigabe an. Diese lässt sich neben der Freigabezeitanpassung durch *Phasentausch* (Veränderung der Reihenfolge der Phasen) oder durch eine *Bedarfsphasenanforderung* (Einschieben einer Phase) realisieren.

Bei der meist modellbasierten verkehrsabhängigen *Signalprogrammabbildung* liegen keine festen Größen für Umlaufzeit und Freigabezeiten vor. Diese werden ständig mit Hilfe einer Zielfunktion, die bestimmte Qualitätskriterien enthält, optimiert. Als Bewertungsgrößen können dafür beispielsweise die Rückstaulänge, die Verkehrsstärke, der Belegungsgrad oder die Zeitlückenverteilung verwendet werden, sofern sie durch Detektion direkt oder indirekt gemessen werden können.

Verkehrsabhängige Linien- und Netzsteuerung

Zusätzlich zu den oben genannten Anpassungen, kommen bei der Linien- und Netzsteuerung als weitere Anpassungsmöglichkeit die Versatzzeiten hinzu. Die verkehrsabhängige Linien- und Netzsteuerung ist in Deutschland meist modellbasiert. Modellbasierte Verfahren bestehen aus einem Verkehrsmodell zur Berechnung von Wirkungskenngrößen, einer Ziel-

funktion zur Berechnung der Güte einer Lösung und einem Optimierungsverfahren zum Finden möglichst guter Lösungen. Stand der Technik sind modellbasierte Verfahren zur Netzsteuerung wie z.B. SCOOT, MOTION und BALANCE, die in Abschnitt 2.3 näher beschrieben werden.

2.2 Optimierung

2.2.1 Begriffe

Der Begriff *Optimum* kommt aus dem Lateinischen und bedeutet *das Beste* oder *das Bestmögliche* bzw. *das Wirksamste*. *Optimierung* lässt sich folglich als *Verbesserung* übersetzen, wobei festgelegt werden muss, was unter *besser* verstanden wird. Um das zu entscheiden, ist eine Gütefunktion erforderlich, die eine problemspezifische Zielsetzung widerspiegelt und dadurch eine Bewertung potentieller Lösungen bzw. Entscheidungsmöglichkeiten ermöglicht.

PAPAGEORGIU [1996] beschreibt das Finden einer optimalen Entscheidung wie folgt: "*Unter Berücksichtigung der Entscheidungsrestriktionen und –auswirkungen bestimme diejenige Entscheidung, die die spezifizierte Zielsetzung am ehesten erfüllt.*"

Ein *Optimierungsproblem* bezeichnet demnach ein Problem, das nach einer bestmöglichen Lösung (= Entscheidung) aus einer Menge von potentiellen Lösungen fragt. Die *Qualität* oder *Güte* jeder potentiellen Lösung wird dabei durch ein Maß bewertet, das der Lösung eine oder mehrere (meist reelle) Zahl(en) zuordnet. Diese wird allgemein als *Güteindex* bezeichnet.

Mathematisch ausgedrückt ist ein *Optimierungsproblem* (Ω, f) definiert durch einen *Suchraum* oder *Lösungsraum* Ω und eine *Bewertungsfunktion* $f : \Omega \rightarrow \mathbb{R}$, die jeder möglichen Lösung x' einen Gütewert $f(x')$ zuweist.

Je nachdem, ob es sich um ein Minimierungs- oder Maximierungsproblem handelt, ist eine Lösung mit minimalem bzw. maximalem Güteindexwert gesucht. Die beiden Probleme sind äquivalent, da gilt

$$f(x') \rightarrow \min! \Leftrightarrow -f(x') \rightarrow \max!$$

$$f(x') \rightarrow \max! \Leftrightarrow -f(x') \rightarrow \min!$$

Im weiteren Text wird deshalb der Einfachheit halber immer davon ausgegangen, dass ein Minimum gesucht wird.

Die Menge der globalen *Minima* $X \subseteq \Omega$ ist dann

$$X = \{x \in \Omega \mid \forall x' \in \Omega : f(x) \leq f(x')\}$$

Ziel eines Optimierungsverfahrens ist es, eine möglichst gute Lösung im Suchraum zu finden, idealerweise ein oder das globale Minimum.

2.2.2 Bekannte Optimierungsverfahren

In der Fachliteratur (vgl. z.B. PAPAGEORGIOU [1996], GERDES et al. [2004], HILDEBRAND [2005], WEICKER [2007]) sind zahlreiche Optimierungsverfahren beschrieben. Einige wesentliche Verfahren werden in diesem Abschnitt kurz dargestellt.

Optimierungsverfahren lassen sich nach den unterschiedlichsten Kriterien in verschiedene Gruppen einordnen. Eine Möglichkeit ist die Einteilung in folgende zwei Gruppen, wobei nicht immer eine eindeutige Zuordnung eines Verfahrens in eine der Gruppen möglich ist:

- Mathematische Verfahren
- Heuristische Verfahren

Der Vorteil mathematischer Verfahren besteht darin, dass man ermitteln kann, wie eine Lösung relativ zum globalen Optimum liegt. Die mathematischen Verfahren lassen sich wiederum unterteilen in *mathematisch exakte Verfahren* und *mathematische Näherungsverfahren*.

Mathematisch exakte Verfahren

Die mathematisch exakten Verfahren können wie folgt unterschieden werden:

- Analytische Optimierungsverfahren
- Optimierungsverfahren mit endlichem Rechenprozess
 - Vollständige Enumeration
 - Lineare Programmierung
 - Dynamische Programmierung

Analytische Optimierungsverfahren

Analytische Optimierungsverfahren berechnen die analytische Lösung direkt. Voraussetzung dafür ist eine stetig differenzierbare Zielfunktion. Meist ist dies allerdings nicht gegeben.

Trotzdem lässt sich in einigen Fällen das Optimum durch einen *endlichen Rechenprozess* numerisch exakt bestimmen.

Optimierungsverfahren mit endlichem Rechenprozess

Sofern der Suchraum klein genug ist, ist es möglich das globale Optimum durch das Auswerten sämtlicher Lösungen zu ermitteln (*vollständige Enumeration*).

Bei der *linearen Programmierung* handelt es sich um relativ häufig eingesetzte Verfahren, mit denen viele Optimierungsaufgaben gelöst werden können. Zur linearen Programmierung zählt beispielsweise das *Simplex-Verfahren*, das in den meisten Fällen exakt zu mindestens einem globalen Optimum führt.

Beim *Simplex-Verfahren* müssen die Randbedingungen und die zu minimierende Zielfunktion des Optimierungsproblems als Linearkombination formuliert werden können. Ist das nicht möglich, kann das Verfahren nicht angewendet werden.

Im Gegensatz zur linearen Programmierung, mit der das Optimierungsproblem unter gleichzeitiger Berücksichtigung aller Variablen und Randbedingungen gelöst werden soll, wird bei der *dynamischen Programmierung* das Problem in Teilprobleme zerlegt und sequenziell gelöst.

Wie bei der vollständigen Enumeration wächst auch bei der dynamischen Programmierung der erforderliche Rechenaufwand mit der Zahl der Eingangs- und Steuergrößen exponentiell.

Mathematische Näherungsverfahren

Mathematische Näherungsverfahren zeichnen sich dadurch aus, dass sie die Lösungen durch einen unendlichen Iterationsprozess gewinnen, der nach endlich vielen Schritten abgebrochen werden muss.

Folgende Optimierungsmethoden zählen unter anderem zu den mathematischen Näherungsverfahren:

- Auf- bzw. Abstiegsverfahren (z.B. Hill-Climbing, Gradientenverfahren)
- Methoden der linearen und dynamischen Programmierung (z.B. Simplex) bei Zielfunktionen, die keine exakte Lösung erlauben

Hill-Climbing Algorithmus

Das einfachste Auf- bzw. Abstiegsverfahren ist der *Hill-Climbing Algorithmus*. Dabei wird ausgehend von einem (meist zufällig) gewählten Startpunkt (Anfangslösung) ein zufälliger Schritt zu einer Nachbarlösung gemacht. Sofern die neue Lösung besser ist, wird sie behalten und als neuer Startpunkt gewählt. Andernfalls wird von der alten Lösung aus weitergesucht. Da jede Verbesserung sofort akzeptiert wird, läuft er meistens in das nächstgelegene lokale Optimum, das er dann nicht mehr verlassen kann.

Gradientenverfahren

Nach einem ähnlichen Prinzip funktioniert das *Gradientenverfahren*. Es verfolgt den Grundgedanken, sich vom Startpunkt aus in diejenige Richtung fortzubewegen, in der die Zielfunktion am stärksten abfällt (bei der Suche des Minimums). Anschließend geht man so lang in diese Richtung, bis sich der Zielfunktionswert nicht mehr verkleinert. Dieser Punkt ist dann der neue Startpunkt.

Um das Gradientenverfahren anwenden zu können, muss allerdings die Zielfunktion im Suchraum differenzierbar sein, da zur Ermittlung der Gradienten die partiellen Ableitungen berechnet werden müssen.

Das Gradientenverfahren steuert ebenfalls direkt auf das nächste lokale Optimum zu. Deshalb ist es empfehlenswert, den Algorithmus mehrmals mit verschiedenen Startwerten auszuführen.

Um das Problem lokaler Optima zu verringern, können die Auf- bzw. Abstiegsverfahren um eine *Tabu-Suche* erweitert werden. Dabei werden auch Verschlechterungen in Kauf genommen. Bei der Tabu-Suche sollen keine Schritte ausgeführt werden, die die Wirkung vorangegangener Schritte wieder rückgängig machen. Dazu werden die komplementären Schritte der erfolgten Schritte in eine Tabu-Liste eingetragen. Die dort eingetragenen Schritte sind dann für eine bestimmte Anzahl von Optimierungsschritten verboten.

Die prinzipiell mathematisch exakten *Methoden der linearen und dynamischen Programmierung* (z.B. Simplex) werden bei Zielfunktionen, die keine exakte Lösung erlauben, ebenfalls zu den mathematischen Näherungsverfahren gezählt.

Mathematische Verfahren stoßen insbesondere dann an ihre Grenzen, wenn der Suchraum viele lokale Optima enthält und wenn die Notwendigkeit besteht, ein Problem in seiner Gesamtheit zu betrachten, weil Ergebnisse isolierter Betrachtungen von Teilproblemen keine Aussage über das Gesamtproblem ermöglichen.

In diesen Fällen kommen häufig heuristische Optimierungsverfahren zum Einsatz, die zwar meist einen höheren rechentechnischen Aufwand bedeuten, aber auch bessere Chancen bieten, globale Optima oder zumindest ausreichend gute lokale Optima zu finden.

Heuristiken

Der Begriff Heuristik (griechisch *heuriskein = finden, auffinden, entdecken*) steht für die Kunst, neue Erkenntnisse oder wahre Aussagen zu finden. Heuristische Verfahren werden auf die Lösung spezieller Problemstrukturen zugeschnitten, wobei meist Expertenwissen, sowie Annahmen und Erkenntnisse aus anderen Lösungsverfahren einfließen.

Im Bereich der Optimierung zählen folgende Methoden zu den heute bekannten und häufig verwendeten heuristischen Verfahren:

- Complex-Methode
- Simulierte Abkühlung
- Ameisenkolonieoptimierung
- Partikelschwärme
- Evolutionäre Algorithmen
- Memetische Algorithmen

Complex-Methode

Bei der *Complex-Methode* wird ein Netz von n Punkten über die (meist mehrdimensionale) Gütefunktionsoberfläche gespannt. Durch iteratives Austauschen der Punkte wird versucht, das Netz auf ein höheres Gütepotenzial zu heben und sich dadurch an das Optimum anzunähern. Sobald das Netz nicht mehr auf ein höheres Potenzial gehoben werden kann, wird der Iterationsprozess abgebrochen.

Simulierte Abkühlung

Das Optimierungsverfahren *Simulierte Abkühlung (Simulated Annealing)* basiert auf der Analogie zum Verhalten einzelner Teilchen bei der Metallschmelze. Durch langsames Abkühlen wird verhindert, dass die Atome im Metallgitter in energetisch ungünstigen Positionen stehen bleiben. Die Teilchen einer Flüssigkeit nehmen bei bestimmten Energieniveaus nicht immer die energetisch günstigste Position in der Umgebung ein, sondern können in Abhängigkeit der Umgebungstemperatur auch schlechtere Positionen einnehmen. Je geringer die Umgebungstemperatur wird, desto weniger wahrscheinlich wird eine ungünstigere Position angenommen.

Analog dazu wird in jedem Optimierungsschritt die zufällig gewählte Nachbarlösung mit einer bestimmten Wahrscheinlichkeit auch dann akzeptiert, wenn sie schlechter als die aktuelle Lösung ist. Die Wahrscheinlichkeit sinkt mit jedem Optimierungsschritt nach und nach (meist exponentiell) auf null. Im Gegensatz zur Complex-Methode kann beim simulierten Abkühlen ein Steckenbleiben in lokalen Optima zumindest in der Anfangsphase des Algorithmus vermieden werden.

Ameisenkolonieoptimierung

Die *Ameisenkolonieoptimierung* basiert auf dem Prinzip, wie Ameisen kürzeste Wege finden. Ameisen orientieren sich an den Duftstoffen (Pheromone) anderer Ameisen. Nimmt eine Ameise keine Pheromone wahr, wählt sie (mehr oder weniger) einen Zufallspfad. Nimmt sie aber eine Pheromonspur wahr, wird sie dieser mit hoher Wahrscheinlichkeit folgen. Da sich

die Pheromone im Lauf der Zeit kontinuierlich verflüchtigen, ist die Pheromonkonzentration auf einem kürzeren Pfad bei gleicher Frequentierung höher als auf einem längeren Pfad. Dadurch wird der kürzere Pfad nach und nach immer mehr Ameisen anlocken.

Die Ameisenkolonieoptimierung lässt sich für alle Optimierungsprobleme anwenden, deren Lösung als ein Weg in einem Graphen dargestellt werden kann. Die virtuelle Ameise nutzt ihr Erinnerungsvermögen, um keine Kanten doppelt zu gehen, sowie das Pheromon, das von anderen Ameisen auf den Kanten verteilt wurde. Sobald eine Ameise einen kompletten Lösungskandidaten erzeugt hat, wird dieser bewertet und in Abhängigkeit seiner Güte entsprechend viel oder wenig Pheromon auf die zugehörigen Kanten verteilt.

Bei der Ameisenkolonieoptimierung findet somit ein Informationsaustausch über gute und schlechte Lösungen statt. Dadurch werden mehr Ressourcen (Ameisen) auf vielversprechende Teile des Lösungsraums verwendet.

Partikelschwärme

Partikelschwärme sind eine Optimierungsmethode, die auf der Modellierung sozialer Interaktion beruht. Verbesserungen werden durch Nachahmung benachbarter Individuen erreicht. Die grundlegende Idee ist die Übertragung des Schwarmverhaltens von Vögeln oder Fischen hinsichtlich optimaler Futterplätze. Dabei steht auf der einen Seite das Bestreben eines Individuums, zu seinen Erfolgen (= guten Lösungen im Suchraum) zurückzukehren (Erfahrungswissen), auf der anderen Seite die Orientierung des Individuums an den besten Erfolgen (= gefundene Lösungen) der Nachbarn (Gruppenwissen).

Die Partikel bewegen sich durch den Suchraum basierend auf ihrer individuellen Geschwindigkeit, wobei die Geschwindigkeiten zufällig initialisiert werden. Die Optimierung mittels Partikelschwärmen basiert also auf einer Menge von Partikeln, die mit einer partikelindividuellen Geschwindigkeit, beeinflusst durch weitere Partikel, ihre Position im Suchraum verändern. Dabei kennt jedes Partikel seine beste Position und deren Güte, sowie die besten Positionen seiner Nachbarn und deren Güte.

Evolutionäre Algorithmen

Evolutionäre Algorithmen sind heuristische Suchverfahren, die den biologischen Evolutionsprozess nachahmen, d.h. sie modellieren verschiedene natürliche Prozesse wie Selektion, Reproduktion, Rekombination, Mutation, Migration, Lokalität, Nachbarschaft, Parallelität oder Konkurrenz.

Optimale Lösungen entwickeln sich nach einer Reihe von Iterationen und Veränderungen als Folge einer besseren *Fitness* gegenüber anderen Lösungen. Eine bessere Fitness (Güte) entspricht einem besseren Zielfunktionswert (Güteindex).

Evolutionäre Algorithmen als Optimierungsmethode bieten einige positive Eigenschaften:

- Es können alle Formen von Bewertungsmodellen eingesetzt werden
- Die Algorithmen können schnell sehr große unregelmäßige Lösungsräume durchsuchen
- Alle zu optimierenden Parameter können gleichzeitig optimiert werden
- Ungerichtete und gerichtete Suchen können kombiniert werden
- Es können Lösungen ermittelt werden, die typischerweise durch eine graduelle Veränderung einzelner Parameterwerte nicht gefunden werden können, das Risiko eines vorzeitigen Abbruchs der Suche bei Erreichen eines lokalen Optimums ist verringert

Evolutionäre Algorithmen benötigen als externe Inputs lediglich einen skalaren Fitnesswert, der die Qualität einer Gesamtlösung angibt. Sie werden häufig bei kombinatorischen Problemen angewendet, deren Lösungsraum schwierig zu durchsuchen ist, d.h. bei hochdimensionalen, multimodalen, un stetigen oder verrauschten Funktionen der Steuerungsgrößen [GREFENSTETTE, 1986].

Memetische Algorithmen

Memetische Algorithmen kombinieren Evolutionäre Algorithmen mit lokalen Suchstrategien. Alle durch den Evolutionären Algorithmus erzeugten Individuen werden lokal optimiert und anschließend wieder in die Population des EA aufgenommen. Zur lokalen Optimierung wird z.B. ein Gradientenverfahren oder Simulierte Abkühlung verwendet. Der Vorteil besteht darin, dass die Individuen im Extremfall (wenn lange genug lokal optimiert wird) alle einem lokalen Optimum entsprechen, wodurch unterschiedliche Eigenschaften des Suchraums sehr schnell zu neuen Individuen kombiniert werden. Nachteilig ist, dass der Bewegungsspielraum der Lösungskandidaten zu stark eingeschränkt werden kann.

2.2.3 Eignung der Verfahren für die Problemstellung

Als kleiner Exkurs soll an dieser Stelle bereits die Eignung der unterschiedlichen Optimierungsverfahren für die Problemstellung diskutiert werden.

WEICKER [2007] schreibt: "*Für jeden Algorithmus gibt es eine Nische im Raum aller Probleme, für die er besonders gut geeignet ist.*"

Wie in der Einleitung bereits erläutert, ist das gegebene Optimierungsproblem, also die netzweite Optimierung der Lichtsignalsteuerung, NP-vollständig. D.h. es ist analytisch nicht lösbar, weil keine geschlossene mathematische Formulierung möglich ist, sondern die Güte nur durch eine Simulation (= Verkehrsmodell) ermittelt werden kann. Folglich sind analytische Optimierungsverfahren nicht geeignet.

Aus der Problemstellung ergeben sich einige *Anforderungen* an das Optimierungsverfahren. Es muss in der Lage sein, mit großen Lösungsräumen, die zahlreiche lokale Minima enthalten, zurechtzukommen, und dabei möglichst schnell gute Lösungen finden. Wünschenswert sind zudem die Möglichkeit der Integration von Nebenbedingungen, eine gleichzeitige Optimierung der Steuerungsparameter und die Möglichkeit zur Parallelisierung des Optimierungsprozesses.

Die Optimierungsverfahren mit endlichem Rechenprozess sind nur für sehr kleine Lösungsräume geeignet, da die erforderliche Rechenzeit mit der Größe des Lösungsraums exponentiell steigt.

Für die Problemstellung werden häufig Abstiegsverfahren wie der Hill-Climbing Algorithmus verwendet. Diese haben allerdings den Nachteil, dass das Optimierungsergebnis sowohl von der verwendeten Startlösung als auch von der Reihenfolge abhängt, in der die zu optimierenden Parameter abgearbeitet werden. Zudem bleiben diese Verfahren sehr leicht in einem lokalen Optimum hängen, weshalb die Optimierung mehrmals mit zufällig ausgewählten Startpunkten wiederholt werden sollte, um die Chance, das globale Optimum zu finden, zu erhöhen.

Auf Grund der oben genannten Anforderungen wird im Rahmen dieser Arbeit auf Evolutionäre Algorithmen zurückgegriffen. Sie kommen mit großen Suchräumen und ungleichmäßigen Fitnessfunktionen zurecht, optimieren alle zu optimierenden Parameter gleichzeitig und sind parallelisierbar. Durch die Integration globaler und lokaler Suchstrategien sind sie in der Lage, relativ schnell gute Lösungen zu finden, was für eine Online-Anwendung von großer Bedeutung ist. Der Einsatz von Evolutionären Algorithmen ist zudem dann gut geeignet, wenn sich die Güte einer Lösung nur über eine Simulation oder ein Modell ermitteln lässt.

Nachteil Evolutionärer Algorithmen ist der im Vergleich zu anderen Methoden relativ hohe Rechenzeitbedarf. Die heutzutage zur Verfügung stehenden Rechner sind jedoch inzwischen leistungsfähig genug, um eine Echtzeitanwendung realisieren zu können.

2.3 Bekannte modellbasierte Online-Netzsteuerungsverfahren

Die international bekanntesten Netzsteuerungsverfahren sind sicherlich SCOOT, SCATS, UTOPIA/SPOT, OPAC und PRODYN. Sie wurden bereits in den 1970er und 1980er Jahren entwickelt. Weitere relativ bekannte Verfahren sind RHODES und TUC. In Deutschland sind bisher MOTION und BALANCE im Einsatz.

SCATS, PRODYN, OPAC, RHODES und UTOPIA/SPOT

Das australische SCATS [LOWRIE, 1982] ist zwar nicht modellbasiert [MARTIN et al., 2003], da es allerdings eines der bekanntesten Netzsteuerungsverfahren ist, wird es hier trotzdem

kurz erwähnt. In SCATS werden Freigabezeitanteile und Umlaufzeit über den Auslastungsgrad bestimmt, die Versatzzeiten werden zentral über eine regelbasierte Signalplanauswahl ermittelt.

Die modellbasierten Verfahren PROLYN [HENRY et al., 1983], OPAC [GARTNER, 1983] und RHODES [SEN und HEAD, 1997] arbeiten *azyklisch*, d.h. sie arbeiten mit einem gleitenden Horizont und setzen keine feste Umlaufzeit voraus.

Bei PROLYN fallen wesentliche Steuerungsentscheidungen auf lokaler Ebene, weshalb es eher als dezentrales System zu bezeichnen ist. OPAC arbeitet sowohl zentral als auch dezentral. Über die verwendeten Optimierungsmethoden sind in der Literatur widersprüchliche Angaben zu finden [vgl. z.B. HOUNSELL und McDONALD, 2001 bzw. PAPAGEORGIU et al., 2003]. Sowohl OPAC als auch PROLYN wird entweder eine vollständige Enumeration (was nur durch eine extreme Einschränkung des Suchraums möglich wird) oder eine dynamische Programmierung zugeschrieben. RHODES verwendet ebenfalls eine dynamische Programmierung zur Optimierung.

Gemäß FRIEDRICH [2002A] ist auch UTOPIA/SPOT [DONATI et al., 1984] zu den azyklischen Verfahren zu zählen, da sich die Steuerung lediglich auf Phasendauern bezieht. Das Verfahren enthält zwei Steuerungsebenen. Die zentrale Ebene (UTOPIA) enthält ein makroskopisches Verkehrsmodell zur groben Wirkungsanalyse. Auf lokaler Ebene optimiert SPOT auf Basis aktueller Messwerte. ÖPNV-Fahrzeuge werden direkt mit berücksichtigt, ihre Abbildung erfolgt über ein eigenes Modell. Eine absolute Priorisierung ist damit allerdings nicht möglich. Als Optimierungsmethode zur Festlegung der Phasendauern kommt ein semi-heuristisches Verfahren zum Einsatz.

TUC

Ein relativ neues Netzsteuerungsverfahren ist TUC [DIAKAKI et al., 2003]. Es wurde als Regelungssystem (sog. Teilstreckenverfahren, store-and-forward) konzipiert, wobei die Modellierung nur auf die erfassten Daten des letzten Umlaufs aufbaut. Optimiert werden primär die Freigabezeiten. Die Versatzzeiten und die Umlaufzeit sind in separaten Modulen optimierbar, wobei für die Versatzzeitenoptimierung Hauptstrecken vorzugeben sind. Versatzzeiten und Umlaufzeit dienen dann als Vorgabe für die Freigabezeitoptimierung. Die Anzahl der Phasen ist fix vorzugeben. Über ein einfaches Modell wird die Anzahl gestauter Fahrzeuge aus dem gemessenen Belegungsgrad der Detektoren ermittelt. Die Steuerung wird als Steuerungsmatrix dargestellt, die offline durch eine linear-quadratische Optimierung bestimmt wurde. Unter Berücksichtigung der getroffenen Annahmen (Zeitkontinuität des Verkehrsablaufs und Rückkopplungscharakteristik des Problems) und der Einschränkungen (feste Umlaufzeit und Versatzzeiten) wird durch die Optimierung die beste Lösung für die Freigabezeitaufteilung gefunden. [vgl. FRIEDRICH und SHAHIN, 2002]

SCOOT

Die wohl bekannteste modellbasierte zentrale Netzsteuerung SCOOT (Split Cycle and Offset Optimisation Technique) wurde am britischen Transport and Road Research Laboratory (TRRL) entwickelt [HUNT et al., 1981]. Die aktuelle Version SCOOT MC3 wird vertrieben von Peek Traffic Limited, Siemens Traffic Controls and TRL Limited [SCOOT, 2008]. SCOOT ist ein rein zentrales System, das keine separate lokale verkehrsabhängige Steuerung berücksichtigt.

SCOOT basiert auf dem in TRANSYT [ROBERTSON, 1969] verwendeten mesoskopischen Verkehrsmodell. Die Verkehrssituation wird mit Induktivschleifen am Beginn einer Zufahrt (Kante), also normalerweise am Ausgang der benachbarten Knotenpunkte, erfasst. Unter Berücksichtigung der Daten vorangegangener Umläufe werden im Verkehrsmodell sog. Verkehrsflussprofile erstellt. Daraus werden in Abhängigkeit der erwarteten Fahrgeschwindigkeit, der Pulkauflösung und des Signalisierungszustandes Staulängen, Wartezeiten und Halte berechnet, welche neben der Auslastung der Zufahrten als Bewertungsgrößen dienen. Für jeden Knotenpunkt müssen Phasenanzahl und -folge vorgegeben werden. Freigabezeitaufteilung (Split), Versatz (Offset) und Umlaufzeit (Cycle Length) werden sukzessive getrennt optimiert.

Die Umlaufzeit wird alle 5 Minuten optimiert. Kriterium für die Festlegung ist die Auslastung des höchstbelasteten Knotens, welcher zu 90% ausgelastet sein soll. Weicht die günstigste Umlaufzeit von der bestehenden ab, wird diese in Schritten von maximal 16 s angepasst. Der ÖPNV kann durch die Änderung der Phasenfolge berücksichtigt werden.

Maßgebend für die Freigabezeitaufteilung ist das Auslastungskriterium. Wenige Sekunden vor dem Ende jeder Freigabezeit wird geprüft, ob die größte am Knoten auftretende Auslastung einer Zufahrt durch eine Verlängerung oder den Abbruch der Grünzeit verringert werden kann. Ist das der Fall, wird diese um maximal 4 s verkürzt oder verlängert.

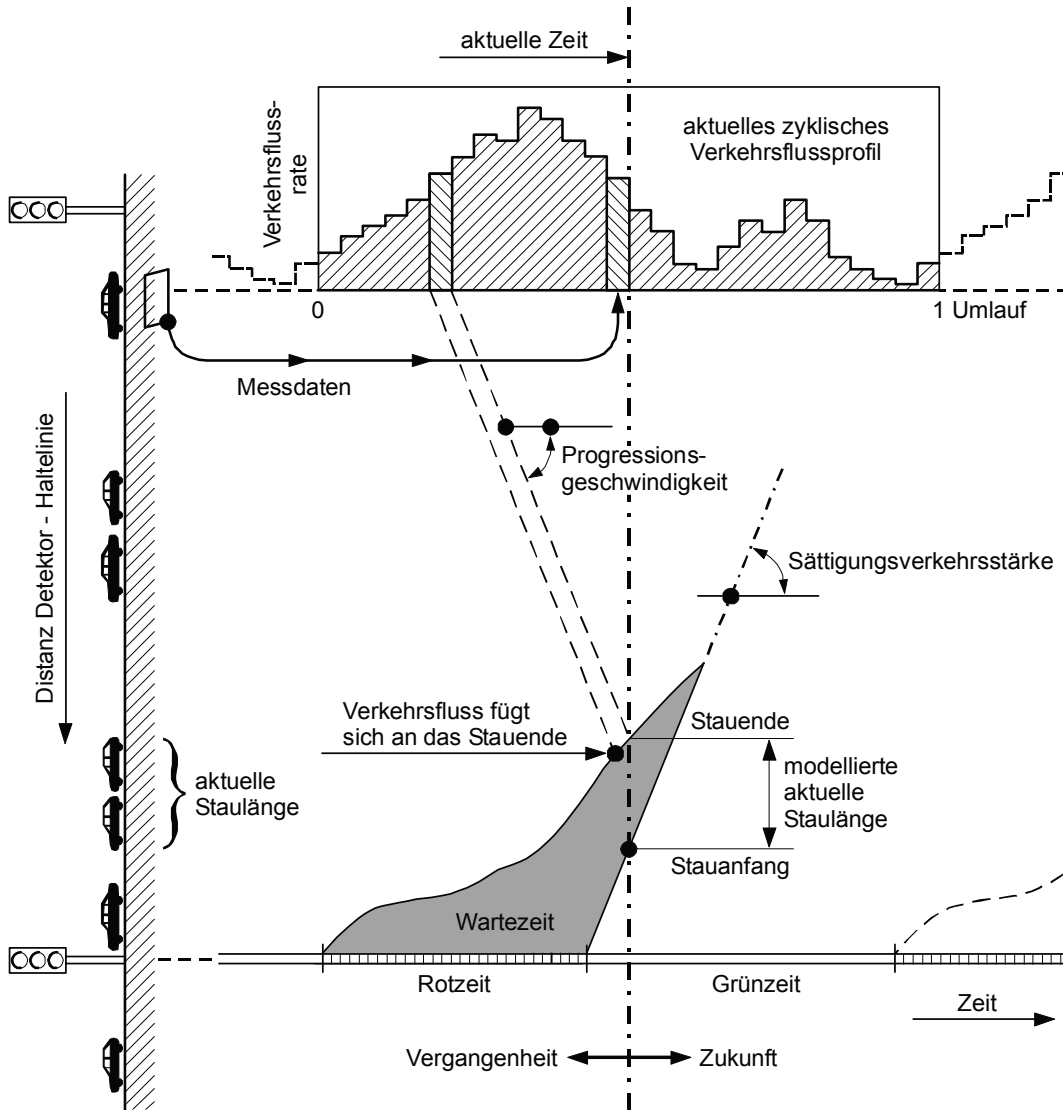


Abb. 2.1 SCOOT-Verkehrsmodell (aus FRIEDRICH und KEMPER [2006] in Anlehnung an HUNT et al. [1981])

Die optimalen Versatzzeiten werden einmal pro Umlauf mit Hilfe eines aus den Verkehrsflussprofilen (Abb. 2.1) für alle zu- und abfließenden Ströme abgeleiteten Güteindex (eine Linearkombination aus Wartezeiten und Halten) für jeden Knoten gegenüber den benachbarten Knoten ermittelt und gegebenenfalls gegenüber ihrer Ausgangslage um wenige Sekunden verändert. Das Verkehrsmodell kommt also ausschließlich bei der Optimierung der Versatzzeiten zum Einsatz, Freigabezeit und Umlaufzeit werden nur über den Auslastungsgrad bestimmt. [vgl. FRIEDRICH, 2002B]

MOTION

Das Netzsteuerungsverfahren MOTION [BUSCH und KRUSE, 1993] hat seinen Ursprung u.a. am ehemaligen Fachgebiet für Verkehrstechnik und Verkehrsplanung der TU München, wo

im Auftrag der Siemens AG eine Projektstudie zur Entwicklung eines verkehrsabhängigen Steuerungsverfahrens MOTION [BIELEFELDT et al., 1986] begonnen wurde.

In der aktuellen Version SITRAFFIC Motion MX 4.0 [MÜCK, 2008] werden Umlaufzeit und Freigabezeitanteile über den Auslastungsgrad bestimmt und dienen als Vorgaben für die anschließende modellbasierte Optimierung der Versatzzeiten und Phasenfolgen. In die Zielfunktion der modellbasierten Optimierung gehen Wartezeiten und Zahl der Halte ein. MOTION liefert als Ergebnis Rahmensignalpläne, die der lokalen verkehrsabhängigen Steuerung zur Verfügung gestellt werden.

Als Optimierungsmethoden stehen zwei Verfahren zur Verfügung. Neben einem Genetischen Algorithmus wurde ein bisher nicht veröffentlichtes deterministisches Optimierungsverfahren entwickelt. [MÜCK, 2008]

BALANCE

Das modellbasierte Netzsteuerungsverfahren BALANCE [FRIEDRICH, 1999] hat seinen Ursprung ebenfalls am ehemaligen Fachgebiet für Verkehrstechnik und Verkehrsplanung der TU München. Die ersten Entwicklungen fanden im Rahmen der Forschungsprojekte Munich COMFORT [MERTZ, 1996] und TABASCO [FRIEDRICH, 1999] statt.

Das Verkehrsmodell ist zweistufig. Ein makroskopisches Verkehrsmodell bestimmt näherungsweise die Quelle-Ziel-Beziehungen im Teilnetz und führt eine Verkehrsumlegung zur Zuordnung der Verkehrsströme zu den einzelnen Kanten im Netz durch. Ein mesoskopisches Verkehrsflussmodell erzeugt dann aus den makroskopischen Verkehrskenngrößen zyklische Verkehrsflussprofile. Dabei werden die Einflüsse der Lichtsignale, der Reisezeit und der Pulkauflösung berücksichtigt. Als Wirkungskenngrößen für die Zielfunktion werden Wartezeiten, Zahl der Halte und Rückstaulängen berechnet. [MERTZ et al., 2000]

BALANCE liefert alle 5 Minuten für alle theoretisch vorgesehenen Phasenfolgen Rahmensignalpläne für eine lokale verkehrsabhängige Steuerung. Optimiert werden dadurch die Freigabezeiten und implizit in eingeschränktem Maß auch die Versatzzeiten. Die Optimierung der Umlaufzeit ist ebenfalls möglich. Als Optimierungsmethode wird ein Hill-Climbing Algorithmus verwendet.

2.4 Evolutionäre Algorithmen in der netzweiten LSA-Steuerung

Evolutionäre Algorithmen kommen immer häufiger zur Lösung komplexer Optimierungsprobleme zum Einsatz, da sie große Lösungsräume effizient nach einem Optimum durchsuchen. Wie bereits erwähnt, stellt die Optimierung der Lichtsignalsteuerung in Netzen ein komplexes (NP-vollständiges) Optimierungsproblem dar, das analytisch nicht gelöst werden kann (vgl. z.B. MERTZ [2001]). Deshalb wurden auch auf diesem Gebiet in den letzten Jah-

ren immer wieder Evolutionäre Algorithmen entwickelt und eingesetzt. Aufgrund der hohen Rechenzeit werden diese jedoch bisher hauptsächlich im Planungsbereich (offline) angewendet.

Evolutionäre Algorithmen in der LSA-Planung

FOY, BENEKOHAL und GOLDBERG [1992] entwarfen eine einfache GA-Optimierung für ein Rasternetz mit vier Knotenpunkten. Für dieses Netz werden Umlaufzeit und Freigabezeitverteilung einfacher Zweiphasensteuerungen optimiert, nicht jedoch die für eine Koordinierung wesentliche Versatzzeit. Zwei hybride GA wurden von HADI und WALLACE [1993] entworfen, die in Verbindung mit TRANSYT-7F Umlaufzeit, Freigabezeiten, Versatzzeiten und Phasenfolgen gleichzeitig optimieren. Von PARK, MESSER und URBANIK [1999] wurde eine GA-Optimierung im Hinblick auf übersättigte Verkehrszustände entwickelt, die auch die Versatzzeit zwischen zwei benachbarten Knotenpunkten mitoptimiert.

DÜRR [2001] entwickelte das Steuerungsverfahren DARVIN, das für vorgegebene Nachfrageszenarien mittels eines Genetischen Algorithmus offline optimale Steuerungslösungen ermittelt. Dabei werden motorisierter Individualverkehr und ÖPNV integriert betrachtet.

Das US-amerikanische Programm TRANSYT-7F [MCTRANS, 2008], das seit 1981 auf Basis der Version 7 des britischen TRANSYT [ROBERTSON, 1969] separat entwickelt wird [HALE, 2005], enthält seit 2001 (Release 9) zusätzlich zum bisherigen Hill-Climbing Algorithmus einen Genetischen Algorithmus als Optimierungsmethode, der sämtliche Steuerungsparameter einer Festzeitsteuerung (Umlaufzeit, Phasenfolge, Freigabezeitaufteilung, Versatzzeiten) optimiert.

CEYLAN und BELL [2004 und 2005] gehen in ihrem Ansatz davon aus, dass die Steuerung Auswirkungen auf die Routenwahlentscheidung hat. Deshalb führen sie die Optimierung auf zwei Ebenen durch: die obere Ebene optimiert die Steuerung mittels eines GA, die untere Ebene optimiert die Verkehrsumlegung. Die Rückkopplung der Signalsteuerung zur Routenwahl ist jedoch ein längerfristiges Phänomen und spielt für die Online-Steuerung keine Rolle.

CEYLAN [2006] schlägt einen kombinierten Einsatz eines GA und eines Hill-Climbing Algorithmus zur optimalen Planung einer netzweiten Festzeitsteuerung vor.

Im Rahmen des Forschungsprojekts RATISBONA-opt wurde ein Genetischer Algorithmus zur netzweiten Offline-Optimierung der Lichtsignal-Koordinierung entwickelt [BRAUN und WEICHENMEIER, 2005]. Optimiert werden sämtliche Steuerungsparameter einer Festzeitsteuerung (Umlaufzeit, Phasenfolge, Freigabezeitaufteilung, Versatzzeiten). Das GALOP-Offline genannte Verfahren wurde in Regensburg eingesetzt, wo die LSA-Steuerung eines

Netzes mit sechs Knotenpunkten optimiert wurde [BRAUN et al., 2005]. Im Anschluss an das Projekt kam es noch in weiteren Teilnetzen in Regensburg zum Einsatz.

ALMASRI [2006] nutzt einen Genetischen Algorithmus zur Optimierung der Versatzzeiten. Das Verfahren ist auf Grund seiner Schnelligkeit theoretisch auch zur Online-Anwendung geeignet, wird im Rahmen der Arbeit allerdings nur zur Offline-Optimierung verwendet. Offen bleibt die Fragestellung, ob Versatzzeiten in einer Online-Anwendung ohne Einschränkung optimiert werden können, oder ob es durch zu große Änderungen der Koordinierungsstruktur von einem zum nächsten Optimierungsintervall zu negativen Effekten im Verkehrsablauf kommt.

YUN und PARK [2006] haben einen binär kodierten GA entwickelt, um für alle LSA eines Netzes die Parameter einer in den USA verbreiteten verkehrsabhängigen Steuerung optimal zu planen. Die zu optimierenden Parameter sind z.B. die zu Grunde liegende Umlaufzeit und die Versatzzeiten, minimale und maximale Freigabedauern, oder unterschiedliche Konfigurationsvarianten für die Detektoren. Zur Bewertung wurde eine mikroskopische Verkehrssimulation (CORSIM) verwendet. Für ein Beispielnetz mit vier Knotenpunkten benötigte die Optimierung drei Stunden und 40 Minuten.

STEVANOVIC et al. [2007] verwenden ebenfalls einen binär kodierten GA zur netzweiten Planung optimaler Signalpläne. Das VISGAOST genannte Verfahren verwendet zur Lösungsbewertung das mikroskopische Verkehrssimulationsprogramm VISSIM. Für ein Netz mit zwölf signalisierten Knotenpunkten wurde die beste Lösung nach 3 200 Bewertungen erreicht, was trotz paralleler Optimierung auf neun Rechnern ca. 90 Stunden dauerte.

Evolutionäre Algorithmen zur Online-Optimierung

Trotz zahlreicher Ansätze mit Evolutionären Algorithmen zur Optimierung der Lichtsignalsteuerung im Planungsbereich oder für einzelne Knotenpunkte gibt es nach Kenntnis des Autors mit MOTION erst eine Anwendung zur *netzweiten Online-Optimierung*. Wie oben beschrieben werden in MOTION für alle LSA des Netzes die Versatzzeiten und die Phasenfolgen optimiert. Der Algorithmus hat jedoch die in ihn gesetzten Erwartungen bisher nicht erfüllt. [MÜCK, 2008]

2.5 Zwischenfazit

In Deutschland spielte sich die verkehrsabhängige LSA-Steuerung in der Vergangenheit im Wesentlichen auf lokaler Ebene, d.h. an den einzelnen Knotenpunkten ab. Sehr häufig kommt dabei die oben beschriebene Zeitlückensteuerung zum Einsatz. Die netzweite Koordination wird üblicherweise vorab geplant und gibt den Rahmen für die lokale verkehrsabhängige Steuerung vor. Für die geplante Koordination wird eine einheitliche Umlaufzeit

vorausgesetzt. Azyklische Ansätze wie in PRODYN, die ohne Umlaufzeit arbeiten, sind in Deutschland derzeit kaum umsetzbar.

Nicht zuletzt deshalb sind im Bereich der verkehrsabhängigen Netzsteuerung in Deutschland die modellbasierten Verfahren MOTION und BALANCE Stand der Technik, die an die hier vorherrschenden Gegebenheiten angepasst sind. Beide setzen eine einheitliche Umlaufzeit für das zu optimierende Netz voraus und geben als Optimierungsergebnis Rahmensignalpläne an die lokalen Steuergeräte weiter.

Ein wesentliches Ziel dieser Arbeit ist, das Optimierungsverfahren so zu entwickeln, dass es umgesetzt werden kann. Deshalb wird es so konzipiert, dass es in BALANCE implementiert werden kann und dort als Alternative zum vorhandenen Hill-Climbing Algorithmus zur Verfügung steht. Die entsprechenden Bedingungen wie eine einheitliche Umlaufzeit und die Randbedingungen der lokalen verkehrsabhängigen Steuerung werden als gegeben vorausgesetzt.

Als geeignet zur Lösung des gegebenen Optimierungsproblems werden Evolutionäre Algorithmen betrachtet (vgl. Abschnitt 2.2.3). Sie wurden zwar auf dem Gebiet der netzweiten LSA-Steuerung bisher auf Grund des relativ hohen Rechenzeitbedarfs hauptsächlich zur Planung eingesetzt, die heutzutage zur Verfügung stehenden Rechner sind jedoch inzwischen leistungsfähig genug, um eine Online-Anwendung realisieren zu können.

3 Gesamtsystem

In diesem Kapitel wird das Gesamtsystem dargestellt. Der Ablauf der Optimierung orientiert sich am Standardregelkreis und wird in Abschnitt 3.1 erläutert. Die wesentlichen Komponenten der Optimierung sind ein Verkehrsmodell (Abschnitt 3.2), eine Zielfunktion (3.3) und das in dieser Arbeit zu entwickelnde Optimierungsverfahren (3.4).

3.1 Regelkreis und Ablauf der Optimierung

Der Standardregelkreis ist in Abb. 3.1 dargestellt. Er besteht aus einer Regelstrecke, einem Regler und zeichnet sich durch die Rückkopplung der Regelgröße aus.

"Es ist die Aufgabe der Regelung, auf Grundlage der Messung den Eingang so zu bestimmen, dass der Ausgang trotz der Einwirkung der Störung Werte annimmt, die dem vorgegebenen Ziel möglichst nahe kommen. (...). Auch hier jedoch stellt die sinnvolle Übersetzung eines Regelungsproblems in ein geeignetes Optimierungsproblem eine nichttriviale Ingenieurleistung dar." [PAPAGEORGIU, 1996]

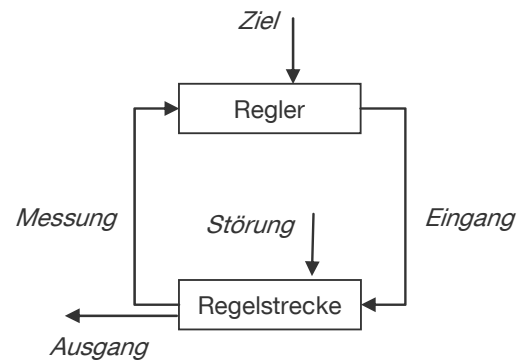


Abb. 3.1 Allgemeiner Regelkreis

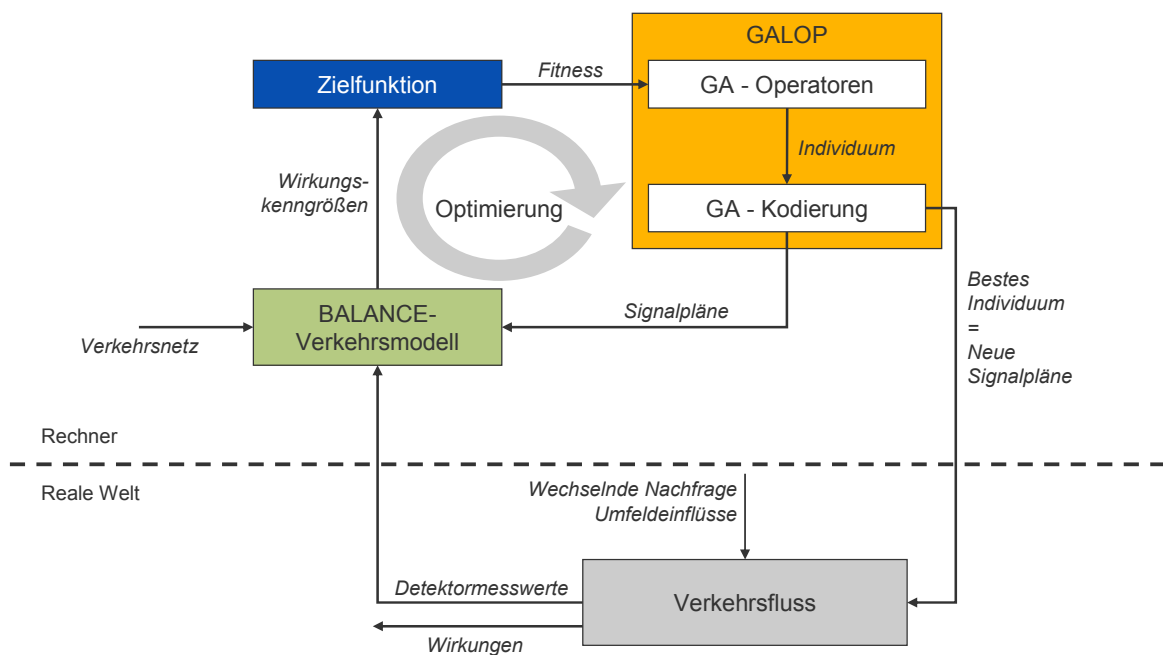


Abb. 3.2 Ablauf der Optimierung

In der vorliegenden Problemstellung entspricht die Regelstrecke dem tatsächlichen Verkehrsfluss. Der Regler enthält alle Komponenten der Optimierung (Abb. 3.2):

- Verkehrsmodell (inkl. Wirkungsmodell)
- Zielfunktion
- Optimierungsverfahren (GALOP)

Eingang für den (realen) Verkehrsfluss sind die optimierten Signalpläne, Ausgang sind die tatsächlichen Wirkungen wie Zahl der Halte und Wartezeiten. Die Detektormesswerte sind die Eingangswerte für den Regler. Störungen im Sinn des Regelkreises sind die wechselnde Verkehrsnachfrage und Umwelteinflüsse, die das Fahrverhalten beeinflussen. Das Ziel der Optimierung wird durch die Zielfunktion abgebildet.

Die eigentliche Optimierung findet im inneren Regelkreis zwischen Verkehrsmodell, Zielfunktion und Optimierungsverfahren statt. Am Ende der Optimierung stehen verbesserte Signalpläne. Die Optimierung kann in bestimmten Abständen, z.B. alle 5 Minuten stattfinden.

3.2 Verkehrsmodell

Im Ablauf der Optimierung ist die Bewertung der einzelnen Lösungen erforderlich. Dazu sind Wirkungskenngrößen zu ermitteln, die sich in Abhängigkeit von der Verkehrsnachfrage für die unterschiedlichen Signalpläne ergeben.

Es wird auf das in BALANCE enthaltene *Verkehrsmodell* zurückgegriffen. Es bildet aus den an den Messquerschnitten gezählten Verkehrsstärken eine interne räumlich-zeitliche Repräsentation des aktuellen Verkehrszustands. Im Wesentlichen besteht es aus drei Funktionsblöcken, die in zwei Stufen angeordnet sind:

- Die erste Stufe ist ein makroskopisches Verkehrsmodell, das einmal pro Bearbeitungsschritt aufgerufen wird. Es besteht aus
 - einer Schätzung der Quelle-Ziel-Beziehungen im Teilnetz
 - einer Verkehrsumlegung zur Zuordnung der Verkehrsströme zu den einzelnen Kanten im Netz
- Die zweite Stufe ist ein mesoskopisches Verkehrsflussmodell, welches aus den makroskopischen Verkehrskenngrößen der ersten Verkehrsmodell-Stufe kantenfeine zyklische Verkehrsfluss-Profile erzeugt.

Das auf dem Verkehrsmodell aufbauende *Wirkungsmodell* dient zur Ermittlung der *Wirkungskenngrößen*. Als relevante Wirkungskenngrößen werden Wartezeiten, Zahl der Halte und Rückstaulängen berechnet.

Die Wirkungskenngrößen werden durch zwei Teilmodelle gebildet: Ein im Sekundenraster relativ fein auflösendes (mesoskopisches) Modell berechnet aus den modellierten Verkehrs-

flussprofilen und den umlaufbezogenen Freigabezeiten den deterministischen Anteil der Wirkungskenngrößen. Es ist sehr ähnlich zu dem in SCOOT verwendeten Modell (vgl. Abb. 2.1 in Abschnitt 2.3). Stochastische Schwankungen und Überlastungen werden mit Hilfe des makroskopischen Warteschlangenmodells von KIMBER und HOLLIS [1979] abgebildet, welches als makroskopische Eingangsgröße nur den Auslastungsgrad einer Zufahrt benötigt. [MERTZ et al., 2000]

Durch die Verwendung des Verkehrsmodells ist sichergestellt, dass tatsächlich systemweit auf verkehrliche Wirkungskenngrößen hin optimiert wird. Das Verkehrsmodell wurde u.a. von FRIEDRICH [1999] ausführlich validiert. Es liefert sowohl für unter- als auch für übersättigte Verkehrszustände ausreichend gute Ergebnisse zur Bewertung einer Lösung. Allerdings weist er zu Recht darauf hin, dass "(...) die Güte der Schätzung in entscheidender Weise von der richtigen Einstellung der mittleren Sättigungsverkehrsstärke bzw. der entsprechenden Häufigkeitsverteilung abhängt." Dieses generelle Problem ist jedoch unabhängig vom Optimierungsverfahren, das im Vordergrund dieser Arbeit steht.

3.3 Zielfunktion

In der Zielfunktion werden die Optimierungsziele abgebildet. Sie ist prinzipiell unabhängig vom Optimierungsverfahren. Die Optimierungsziele werden durch Optimierungskriterien festgelegt. Für die Verbesserung des Verkehrsablaufs können dies z.B. sein:

- Wartezeiten
- Zahl der Halte
- Rückstaulängen

Diese Größen können für den motorisierten Verkehr durch das Verkehrsmodell berechnet werden. Für Fußgänger können durchschnittliche und maximale Wartezeiten direkt aus den Signalplänen ermittelt werden. Allerdings ist auf Grund fehlender Detektion meist nicht bekannt, ob und wie viele Fußgänger gerade betroffen sind.

Liegen mehrere Optimierungskriterien vor, spricht man auch von einer *Mehrkriterienbewertung*. Grundsätzlich gibt es bei der Mehrkriterienbewertung mit k Optimierungskriterien auch k Zielfunktionen $f_i(s)$. Es bestehen zwei prinzipielle Möglichkeiten zur Lösung von Mehrkriterienoptimierungsproblemen:

- Zusammenfassung der einzelnen Zielfunktionen zu einer Gesamtzielfunktion
- Suche nach pareto-optimalen Lösungen

Die *Zusammenfassung der einzelnen Zielfunktionen* zu einer Gesamtzielfunktion geschieht über die Bildung der gewichteten Summe über die Einzelkriterien:

$$f(s) = \sum_{i=1}^k w_i \cdot f_i(s)$$

Über die Gewichte lässt sich die Wichtigkeit der einzelnen Kriterien bestimmen. Bei der Wahl der Gewichte ist allerdings die jeweilige Schwankungsbreite der einzelnen Zielfunktionen zu berücksichtigen. Wenn eine Zielfunktion im Vergleich zu den anderen Zielfunktionen nur eine geringe Schwankungsbreite hat, wird auch eine hohe Gewichtung dieser Zielfunktion keinen großen Einfluss auf das Optimierungsergebnis haben.

Bei der *Suche nach pareto-optimalen Lösungen* wird keine Gesamtzielfunktion gebildet. Eine Lösung s_0 ist dann bezüglich der Zielfunktionen f_i pareto-optimal, wenn keine andere Lösung existiert, die für alle Zielfunktionen mindestens gleich gute Werte und mindestens für eine Zielfunktion einen besseren Wert besitzt. Aus den pareto-optimalen Lösungen muss dann noch entschieden werden, welche Lösung ausgewählt werden soll.

Die Ermittlung von einem Satz pareto-optimaler Lösungen ist vor allem dann sinnvoll, wenn sich die einzelnen Zielkriterien grundsätzlich widersprechen. Dies ist beim vorliegenden Problem nicht der Fall. Eine hohe Anzahl von Halten bedeutet meist auch eine hohe Wartezeit und eine große Rückstaulänge. Folglich ist es günstiger und deutlich weniger rechenzeitaufwändig, die Einzelkriterien in einer Zielfunktion zusammenzufassen und entsprechend ihrer Wichtigkeit zu gewichten, bzw. daraus ableitbare Optimierungskriterien wie beispielsweise den Kraftstoffverbrauch zu berechnen.

Die aus dem Verkehrsmodell zur Verfügung stehenden signalgruppenbezogenen Wirkungskenngrößen (Wartezeiten W_{sg} , Zahl der Halte H_{sg} , maximale Rückstaulängen L_{sg}) aller Signalgruppen sg können in der Zielfunktion zu einer gewichteten Summe kombiniert werden:

$$Fitness = \sum_{sg \in SG} (\alpha_{sg} \cdot W_{sg} + \beta_{sg} \cdot H_{sg} + \gamma_{sg} \cdot L_{sg})$$

Als Ergebnis erhält man den Zielfunktionswert, der die Fitness (Güte) der bewerteten Lösung angibt.

Durch Änderung der Gewichte $(\alpha_{sg}, \beta_{sg}, \gamma_{sg})$ können zusätzliches Expertenwissen sowie planerische und ggf. politische Vorgaben in den Optimierungsprozess eingebracht werden. Beispielsweise können Hauptstraßen, auf denen der Verkehr gebündelt werden soll, um sensiblere Bereiche zu entlasten, durch eine entsprechende Wahl der Gewichte in der Zielfunktion bevorzugt werden.

Wie bereits erwähnt, können die Gewichte für die Zahl der Halte und die Wartezeiten auch so kombiniert werden, dass der Zielfunktionswert dem Kraftstoffverbrauch entspricht. SCHNABEL und LOHSE [1997] geben beispielsweise für den Mehrverbrauch eines Abbrems-

Anfahrt Zyklus bei einer Geschwindigkeit von 50 km/h für einen PKW mit einem Gewicht von 1,3 t 20,5 ml Benzin an. Pro Stunde Wartezeit ergibt sich ohne Betrachtung des Abbrems-Anfahrt Zyklus ein Benzinverbrauch von durchschnittlich einem Liter. Neuere Zahlen liegen in einer ähnlichen Größenordnung. Deutlich wird, dass es aus Verbrauchssicht besser ist, die Zahl der Halte zu reduzieren und dafür ggf. eine längere Wartezeit in Kauf zu nehmen, als umgekehrt.

Die Zielfunktion kann auch um *Bestrafungsterme* zur Unterstützung des Optimierungsverfahrens erweitert werden. Beispielsweise können bei einem Evolutionären Algorithmus in Abhängigkeit der gewählten Kodierung ungültige Individuen entstehen (siehe Abschnitt 4.3.1). Durch einen Bestrafungsterm in der Zielfunktion kann dann erreicht werden, dass inkorrekte Lösungen deutlich schlechter bewertet werden, wodurch sich die Wahrscheinlichkeit, dass diese Lösungen im weiteren Optimierungsverlauf Bestand haben, verringert. Das Behalten inkorrektur Lösungen ist insbesondere dann sinnvoll, wenn der Lösungsraum durch die gegebenen Randbedingungen in mehrere Teilräume zerfällt. Ein sofortiges Löschen inkorrektur Lösungen könnte in diesen Fällen dazu führen, dass einige Teilräume im Lauf der Optimierung nicht erreicht werden, weil kein Individuum den Sprung von einem zum anderen Teilraum über den ungültigen Bereich hinweg schafft.

Eine weitere Möglichkeit ist, die Zielfunktion in Abhängigkeit vom Verkehrszustand zu verändern. Beispielsweise könnte es sinnvoll sein, bei untersättigtem Verkehrszustand die Zahl der Halte hoch zu gewichten, bei übersättigtem Verkehrszustand hingegen die Gewichte der Rückstaulängen hoch zu setzen.

3.4 Optimierungsverfahren

Bei einer Online-Optimierung handelt es sich um ein *zeitabhängiges* Optimierungsproblem, da sich die Verkehrsnachfrage ständig ändert. Für das Optimierungsmodell bedeutet das, dass gleiche Lösungen je nach Verkehrsnachfrage unterschiedliche Zielfunktionswerte haben, d.h. mit der Änderung der Verkehrsstärke ändert sich auch die Gütelandschaft und damit die Lage des globalen Optimums im Suchraum.

Ein zeitabhängiges Optimierungsproblem lässt sich jedoch als Folge statischer Optimierungsprobleme betrachten, wobei für jedes Zeitintervall eine optimale Lösung gefunden werden soll. Für jedes dieser Zeitintervalle wird die Verkehrsbelastung als konstant angesetzt.

Die Lösung für ein Zeitintervall (= Optimierungsintervall) sollte allerdings trotzdem nicht als komplett unabhängig von der Lösung des vorangegangenen Zeitintervalls – also der aktuellen Schaltung – gesehen werden. So ist zu vermuten, dass insbesondere die Versatzzeiten nur in relativ kleinen Schritten verändert werden sollten, um keine unerwünschten Effekte im

Verkehrsablauf durch eine zu plötzliche Änderung der Koordinierungsstruktur zu erhalten, die zudem nicht über das Verkehrsmodell abgebildet werden können, und deshalb auch in der Optimierung nicht erkannt werden können.

Das zu entwickelnde Optimierungsverfahren ist der Kern dieser Arbeit und wird im folgenden Kapitel ausführlich beschrieben.

4 Evolutionärer Algorithmus GALOP

In diesem Kapitel wird der echtzeitfähige Genetische Algorithmus zur netzweiten Optimierung der Lichtsignalsteuerung GALOP dargestellt, der im Rahmen dieser Arbeit entwickelt wird. Nach dem einführenden Abschnitt 4.1 zu den Grundlagen Evolutionärer Algorithmen folgt in Abschnitt 4.2 die verwendete Entwurfsmethodik für die Entwicklung des Algorithmus. Abschnitt 4.3 befasst sich mit dem entscheidenden Teil der Entwicklung, der Kodierung der Steuerungsparameter im Individuum. Die Abschnitte 4.4 bis 4.7 beschreiben die weiteren Bestandteile des Algorithmus, u.a. in Abschnitt 4.5 die evolutionären Operatoren des Algorithmus. Abschließend folgt ein Zwischenfazit (Abschnitt 4.8).

4.1 Allgemeines zu Evolutionären Algorithmen

Der Begriff Evolutionärer Algorithmus wird heute meist als Überbegriff für zahlreiche Varianten sehr ähnlich aufgebauter heuristischer Optimierungsmethoden verwendet, die Prinzipien der Evolution übernehmen. Ein Evolutionärer Algorithmus bildet verschiedene natürliche Prozesse wie z.B. Selektion, Reproduktion, Rekombination, und/oder Mutation nach. Alle möglichen Lösungen des Suchraums werden als Individuen aufgefasst. Optimale Lösungen entwickeln sich nach einer Reihe von Iterationen und Veränderungen als Folge einer besseren *Fitness* gegenüber anderen Lösungen. Eine bessere Fitness entspricht einem besseren Zielfunktionswert.

Die eigentliche Struktur des Suchraums Ω wird als *Phänotyp*, die Darstellung einer möglichen Lösung als *Genotyp* bezeichnet. Die Bewertungsfunktion (Zielfunktion) bezieht sich auf den Phänotyp, während sich die Evolutionären Operatoren wie Mutation und Rekombination auf den Genotyp beziehen. Um ein im Genotyp dargestelltes Individuum bewerten zu können, d.h. ihm eine Fitness zuweisen zu können, muss es in den phänotypischen Suchraum abgebildet werden. Die Übersetzung eines Individuums vom Phänotyp in den Genotyp wird als *Kodierung*, die Rückübersetzung entsprechend als *Dekodierung* bezeichnet.

Aus der Historie heraus kann man vier Grundrichtungen Evolutionärer Algorithmen unterscheiden:

- Genetische Algorithmen
- Evolutionsstrategien
- Evolutionäre Programmierung
- Genetische Programmierung

Die Zielrichtung der Evolutionären bzw. Genetischen Programmierung ist ursprünglich nicht das Lösen von Optimierungsproblemen, sondern die Erzeugung intelligenter Programme (Automaten) bzw. die Simulation intelligenten Verhaltens.

Genetische Algorithmen und Evolutionsstrategien wurden zunächst separat voneinander entwickelt. Die ersten Genetischen Algorithmen wurden ausschließlich binär kodiert. Bei den Evolutionsstrategien wurde zu Beginn nur mit Mutationen gearbeitet, andere Operatoren kamen zunächst nicht zum Einsatz.

Die Grenzen zwischen den einzelnen Formen sind inzwischen nicht mehr klar zu ziehen. Die Elemente der zunächst unterschiedlichen Ansätze können problemlos kombiniert werden, um einen Algorithmus zu entwerfen, der möglichst gut an das zu lösende Problem angepasst ist.

Obwohl es sich historisch gesehen bei Genetischen Algorithmen um eine Untermenge der Evolutionären Algorithmen handelt, werden die Begriffe Evolutionärer Algorithmus und Genetischer Algorithmus inzwischen häufig synonym verwendet. Ausführliche Informationen zur Historie Evolutionärer Algorithmen finden sich z.B. in WEICKER [2007].

Der allgemeine Ablauf eines Evolutionären Algorithmus ist in Abb. 4.1 dargestellt.

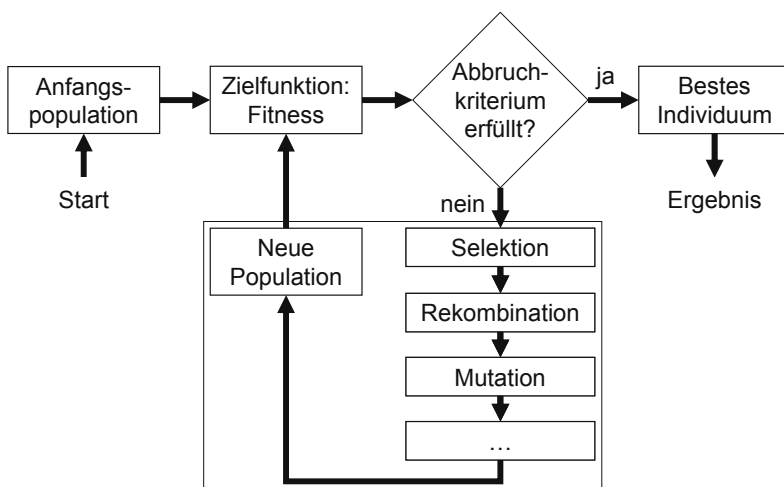


Abb. 4.1 Allgemeiner Ablauf eines Evolutionären Algorithmus

Zu Beginn einer Optimierung wird eine Anfangspopulation erzeugt. Die Individuen der Anfangspopulation werden üblicherweise zufällig generiert, es können aber auch bereits bekannte Individuen mit integriert werden. Jedes Individuum muss bewertet werden, d.h. seine Fitness wird ermittelt.

Mit Hilfe der Fitnesswerte findet eine Auswahl von Eltern statt (*Selektion*). Durch eine (meist paarweise) *Rekombination* von Elternindividuen werden neue Individuen erzeugt. Die einzelnen Gene der Nachkommen werden anschließend mit einer bestimmten Wahrscheinlich-

keit verändert (*Mutation*). Die Nachkommen ersetzen einen Großteil der alten Population und bilden zusammen mit dem oder den besten Individuen der Elterngeneration eine neue Population. Diese wird wiederum bewertet, d.h. es wird die Fitness der einzelnen Individuen als Grundlage für die Selektion der nächsten Eltern ermittelt. Abbruchkriterium ist meist eine maximale Anzahl von Generationen.

Neben den genannten Standardoperatoren Selektion, Rekombination und Mutation ist der Einsatz weiterer Operatoren wie Migration und Konkurrenz möglich, was den Algorithmus noch leistungsfähiger machen kann [vgl. z.B. POHLHEIM, 1999].

4.2 Entwurfsmethodik

"(...) das Anwenden des Basisalgorithmus ist zwar prinzipiell leicht möglich, die Ergebnisse sind aber häufig nicht von überzeugender Qualität. Dem entsprechend groß ist die Notwendigkeit, Operatoren und Parameter optimal an das betrachtete Problem anzupassen."
[GERDES et al., 2004]

Wenn man sich darauf beschränkt, unreflektiert einen Standardalgorithmus zu übernehmen und die Anpassung an das zu lösende Optimierungsproblem nur auf Parametereinstellungen zu beschränken, wird als Ergebnis meist ein Algorithmus stehen, der entweder mangelhafte Lösungskandidaten liefert oder hohe Rechenzeiten erfordert.

Als Alternative zur simplen Auswahl eines Standardalgorithmus steht die Wahl eines Entwurfsmusters, das gemäß anderer Literatur für ein ähnliches Problem besonders geeignet ist. Das Problem dabei ist jedoch die Charakterisierung der Problemklassen. Existierende Klassifikationen sind deutlich zu grob. *"Alle Versuche, feinere Klassifikationen zu entwickeln sind bisher gescheitert, da es keine funktionierende Menge an Metriken gibt, welche die Eigenarten und Schwierigkeiten eines Optimierungsproblems hinreichend beschreiben."*
[WEICKER, 2007]

Für die Entwicklung eines Evolutionären Algorithmus gibt es offensichtlich keine standardisierte Vorgehensweise. Von entscheidender Bedeutung für den Erfolg des Algorithmus ist ein gutes Verständnis des Problems. Das Einbringen von Problemwissen lässt sich jedoch nur schwer standardisieren.

In Anlehnung an WEICKER [2007] wird in dieser Arbeit die in Abb. 4.2 dargestellte Vorgehensweise für den Entwurf des Evolutionären Algorithmus gewählt.

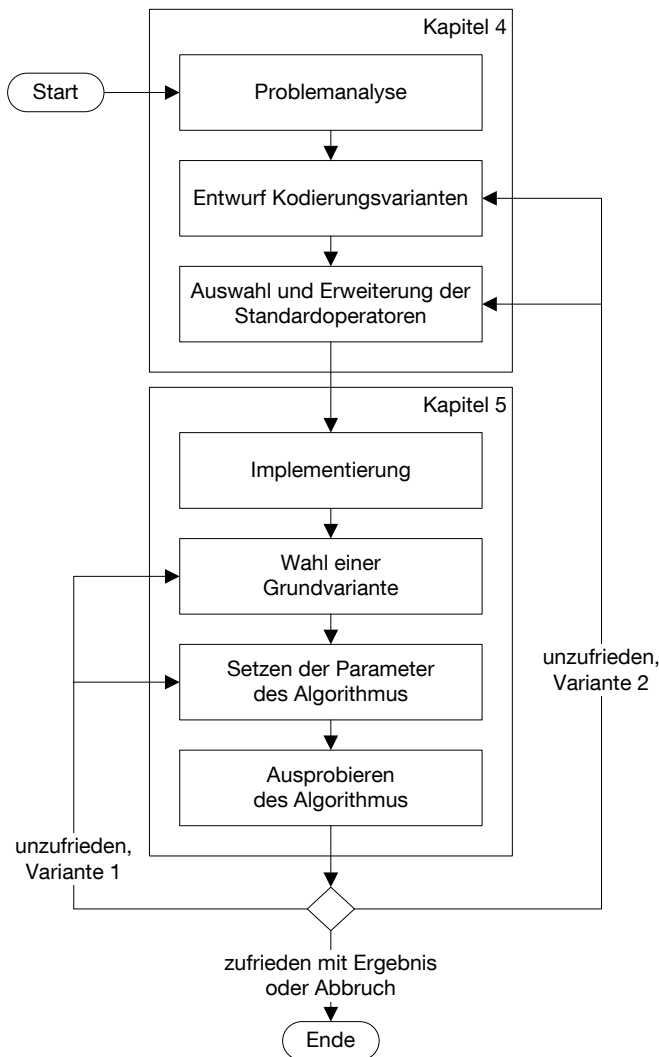


Abb. 4.2 Entwurfsmethodik (in Anlehnung an WEICKER [2007])

Begonnen wird der Entwurf mit einer *Problemanalyse*. Dazu zählen insbesondere die problemspezifischen Randbedingungen, die in Abschnitt 4.3.1 erläutert werden, da sie für den *Entwurf der Kodierungsvarianten* eine wesentliche Rolle spielen. Schließlich erfolgt die *Auswahl und Erweiterung der Standardoperatoren*.

Nach den ersten drei Schritten, deren Ergebnisse in den folgenden Abschnitten des Kapitels 4 dargestellt werden, folgt die Implementierung, die Wahl einer Grundvariante, das Setzen der Parameter und das Ausprobieren des Algorithmus. Wie Abb. 4.2 verdeutlicht, handelt es sich um einen iterativen Prozess. Die letzten drei Schritte sind gleichzeitig bereits Teil der Bewertung des Optimierungsverfahrens und deshalb Thema in Kapitel 5 dieser Arbeit.

4.3 Kodierung der Steuerungsparameter

Von entscheidender Bedeutung für die Qualität und Funktionsfähigkeit der Optimierung ist die *Kodierung der Steuerungsparameter*. Unter Kodierung versteht man bei einem Evolutionären Algorithmus die Übersetzung eines *Phänotyps* in einen *Genotyp*, d.h. in diesem Fall der Signalpläne in ein für einen Evolutionären Algorithmus verarbeitbares Individuum. Die Rückübersetzung eines Individuums in die Signalpläne wird dementsprechend als Dekodierung bezeichnet und durch die Dekodierungsfunktion definiert. Der Einfachheit halber wird *Kodierung* häufig als Überbegriff für beide Richtungen verwendet.

Im Idealfall stellt die Dekodierungsfunktion eine Bijektion dar, d.h. zu einem Phänotyp gibt es genau einen Genotyp und umgekehrt. Häufig ist das allerdings nicht möglich, was zur Folge hat, dass mindestens zwei Individuen existieren, die durch die Dekodierungsfunktion auf den selben Phänotyp abgebildet werden. Anzustreben ist in jedem Fall eine möglichst schlüssige und allgemeingültige Repräsentation der einzelnen Parameter als Gene im Individuum.

Die ersten Genetischen Algorithmen wurden *binär* kodiert. D.h. es wird eine Dekodierungsfunktion benötigt, die eine binäre Zeichenkette auf einen ganzzahligen oder reellen Wert abbildet. Der Unterschied zwischen Genotyp und Phänotyp ist dabei meist sehr groß. Phänotypisch aufeinanderfolgende Werte können wegen der Bit-Struktur durch zwei komplett unterschiedliche Bitmuster dargestellt werden. Ein wesentliches Problem der binären Kodierung ist deshalb, dass sie Nichtlinearitäten in das Optimierungsproblem bringt. Das hat zur Folge, dass phänotypisch benachbarte Werte nicht durch einfache Anwendung des Mutationsoperators ineinander überführt werden können. [WEICKER, 2007]

Demgegenüber haben *reellwertige Gene* den Vorteil, dass sie direkt als numerischer Wert interpretiert werden können. Außerdem verfügen reellwertige Algorithmen über die höchste Zahl an Freiheitsgraden bezüglich Rekombination und Mutation. [HILDEBRAND, 2005]

MICHALEWICZ [1996] zeigte in einem ausführlichen Vergleich zwischen binärer- und reeller Kodierung, dass die Kodierung mit reellen Zahlen (Fließkommazahlen) sowohl schneller und genauer, als auch stabiler in den erreichten Lösungen war.

4.3.1 Problemspezifische Randbedingungen

Die Schwierigkeit der Kodierung ergibt sich primär aus den *Randbedingungen* des Problems, die eine Einschränkung des Suchraums bedeuten. Randbedingungen, die zwingend erfüllt sein müssen, werden als harte Randbedingungen bezeichnet. Randbedingungen, deren Erfüllung wünschenswert ist, werden als weiche Randbedingungen bezeichnet. Lösungen, die eine harte Randbedingung nicht einhalten, nennt man ungültige Individuen.

Je nach gewählter Kodierung können mehr oder weniger viele unzulässige Lösungen auftreten, die die erforderlichen Randbedingungen nicht einhalten. Da diese Lösungen nicht verwendet werden können, ist ein wichtiges Ziel, die Zahl ungültiger Individuen möglichst gering zu halten. Dazu gibt es mehrere Möglichkeiten [vgl. SADEK et al., 1997]:

- Wahl einer relativen Kodierung, die die Randbedingungen bereits in der Art enthält, dass nur gültige Lösungen produziert werden
- Einführung einer Bestrafungsfunktion für Individuen, die Randbedingungen verletzen
- Entwurf geeigneter Dekodierungs- oder Reparaturalgorithmen
- Iterative Neuversuche bis ein gültiges Individuum erzeugt wird [DÜRR, 2001]

Für die gegebene Problemstellung sind die folgenden problemspezifischen Randbedingungen für die Kodierung relevant:

- Planerische Vorgaben (erlaubte Umlaufzeiten, erlaubte Phasenfolgen)
- Zwingende Randbedingungen (Zwischenzeiten, Mindestfreigabedauern)
- Randbedingungen der lokalen verkehrsabhängigen Steuerung

Welche Umlaufzeiten und Phasenfolgen erlaubt sind, muss vorgegeben werden und ergibt sich unter anderem aus den örtlichen Gegebenheiten.

Die zwingenden Randbedingungen, nämlich Zwischenzeiten und Mindestfreigabedauern, müssen immer eingehalten sein. Diese zeitlichen Anteile eines Signalplans stehen folglich nicht für die Optimierung zur Verfügung.

Für die Online-Optimierung sind zusätzlich die bereits vorhandenen lokalen (d.h. an einem einzelnen Knotenpunkt) verkehrsabhängigen Steuerungen und ÖPNV-Priorisierungen zu beachten. Die in Deutschland weit verbreitete lokale Verkehrsabhängigkeit führt zu einigen Einschränkungen. So ist bei der messwertbasierten Zeitlückensteuerung für den Abbruch einer Phase ein Rahmen vorgegeben, innerhalb dessen der Abbruch erfolgen kann. Der Abbruch einer Phase ist gleichbedeutend mit dem Start eines Phasenübergangs. Der Rahmen kann z.B. durch sogenannte T-Zeit-Grenzen (T_{IA}, T_{IB}) festgelegt werden (Abb. 4.3).

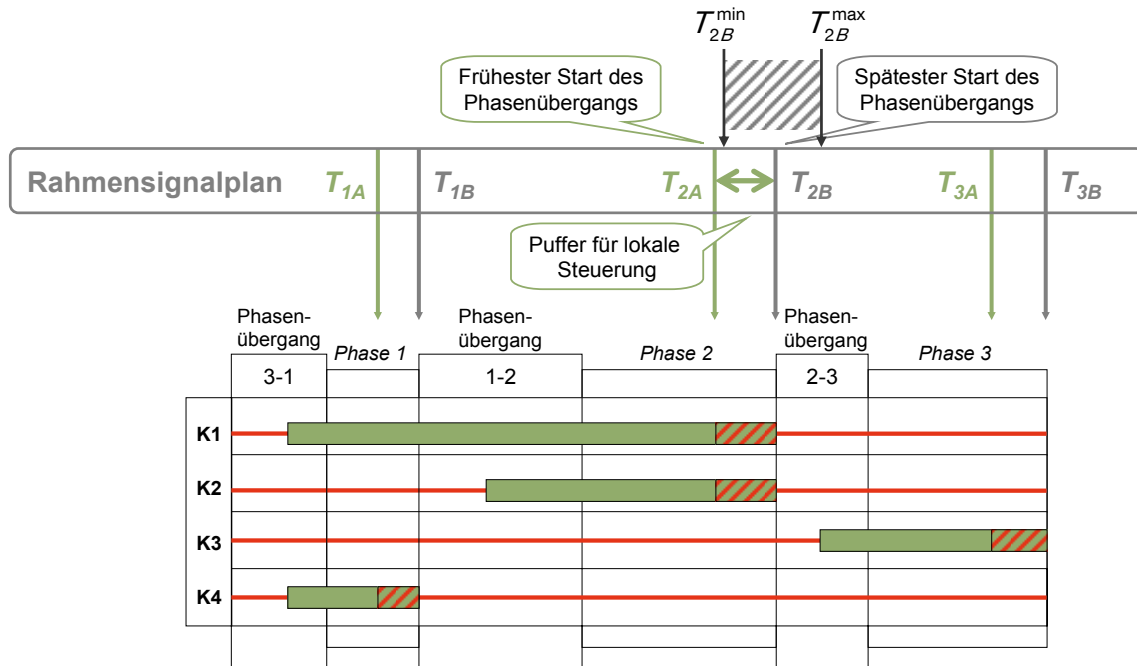


Abb. 4.3 T-Zeit-Grenzen für die lokale verkehrsabhängige Steuerung

Die Optimierungsaufgabe besteht darin, die spätesten Starts der Phasenübergänge T_{iB} für alle LSA im betrachteten Netz so zu bestimmen, dass der Zielfunktionswert einen möglichst geringen Wert erreicht. Die frühesten Starts eines Phasenübergangs T_{iA} berechnen sich direkt aus T_{iB} und der zugehörigen Freigabedauer t_{Fi} ($T_{iA} = T_{iB} - 0,3 \cdot t_{Fi}$). Sie sind folglich keine Variablen für die Optimierung.

Um die Funktionalität der lokalen verkehrsabhängigen Steuerung sicherzustellen, ist für die T-Zeit T_{iB} ein Intervall $[T_{iB}^{\min}; T_{iB}^{\max}]$ vorgegeben, innerhalb dessen Grenzen sich T_{iB} befinden muss (siehe Abb. 4.3).

Die Herausforderung besteht darin, die Kodierung so zu gestalten, dass die komplexen Randbedingungen möglichst allgemeingültig integriert sind. Neben den problemspezifischen Randbedingungen ist bei der Kodierung auch eine möglichst schlüssige Repräsentation der einzelnen Parameter als Gene im Individuum anzustreben.

4.3.2 Repräsentation der Steuerungsparameter im Individuum

Mit vorgegebenen Phasen kann der Signalplan eines Knotenpunkts durch folgende Steuerungsparameter festgelegt werden:

- Umlaufzeit
- Versatzzeit
- Phasenfolge
- Phasendauern bzw. Startzeitpunkte der Phasenübergänge

Ein *Individuum* repräsentiert alle Signalpläne des zu optimierenden Netzes und hat folgende Gestalt:

$$\{\varphi, (\sigma_1, \omega_1, o_1, \theta_{11}, \dots, \theta_{1m_1}), (\sigma_2, \omega_2, o_2, \theta_{21}, \dots, \theta_{2m_2}), \dots, (\sigma_n, \omega_n, o_n, \theta_{n1}, \dots, \theta_{nm_n})\}$$

Es enthält ein Gen φ für die gemeinsame Umlaufzeit sowie n sogenannte *Chromosomen* für n Knotenpunkte des zu optimierenden Netzes. Jedes Chromosom besteht aus einem Gen σ zur Festlegung der Phasenfolge, einem Gen ω für den globalen Versatz (das allerdings im Rahmen dieser Arbeit inaktiv bleibt), einem Gen o für den lokalen Versatz, sowie m Genen θ für die Freigabezeiten. Jedes Gen nimmt einen reellen Wert zwischen 0 und 1 an¹.

In den folgenden Abschnitten werden die Eigenschaften der einzelnen Steuerungsparameter diskutiert und die entwickelten Kodierungen dargestellt.

4.3.3 Umlaufzeit

Die *Umlaufzeit* wird für alle LSA des zu optimierenden Netzes gleich gewählt, da sonst keine koordinierte Steuerung möglich ist. Die erlaubten Umlaufzeiten müssen vorgegeben werden. Halbe oder Drittel Umlaufzeiten für wenig belastete Knotenpunkte können über die Festlegung von Phasenfolgen realisiert werden (siehe Abschnitt 4.3.5).

Die Umlaufzeit t_U errechnet sich zu

$$t_U = t_{U,\min} + \varphi \cdot (t_{U,\max} - t_{U,\min})$$

Das Ergebnis wird auf die nächstliegende erlaubte Umlaufzeit gerundet.

4.3.4 Versatzzeit

Die *globale Versatzzeit* gibt die Differenz des Referenzpunkts des jeweiligen lokalen Signalplans zur *globalen* (netzweit gültigen) Zeitskala an. Während bei der Planung optimaler Festzeitsignalpläne die globale Versatzzeit eines Knotenpunkts alle Werte zwischen 0 und t_U annehmen kann [vgl. BRAUN und WEICHENMEIER, 2005], gestaltet sich die Online-Optimierung der Versatzzeit deutlich komplizierter.

Die globalen Versatzzeiten sind der wesentliche Parameter für die Festlegung der Koordinierungsstruktur im Netz. Es ist deshalb zu vermuten, dass sich bei einer starken Änderung

¹ Eine ganzzahlige Kodierung wäre für das gegebene Problem nicht sinnvoll umsetzbar, weil die Phasendauern voneinander abhängig sind

der Versatzzeiten von einem zum nächsten Optimierungsintervall unerwünschte Effekte im Verkehrsablauf ergeben, da sich die Koordinierungsstruktur plötzlich wesentlich ändert.

Hierbei spielt v.a. die Tatsache eine Rolle, dass es je nach gegebener Netzstruktur und Nachfragestruktur sehr gute Lösungen mit gleichen Fitnesswerten geben kann, die jedoch eine komplett andere Lösungsform haben. Typisches Beispiel ist eine symmetrische Strecke mit gleicher Nachfrage in beiden Richtungen. Offensichtlich gibt es dafür zwei gleich gute Lösungen, nämlich eine Koordinierung in die eine oder in die andere Richtung. Schränkt man in diesem Fall die Versatzzeitenoptimierung nicht ein, wird es durch kleinste Änderungen der Verkehrsnachfrage zu einer kompletten Änderung der Koordinierungsrichtung von einem zum nächsten Optimierungsintervall kommen. Dass eine derartige Konstellation in der Realität vorkommt, konnte in BRAUN et al. [2005] gezeigt werden. Hier fand der Algorithmus GALOP-Offline für ein kleines Netz mit 6 Knotenpunkten zwei ähnlich gute Lösungen, die eine unterschiedliche Koordinierungsstruktur repräsentierten. Für eine derartige Änderung der Koordinierungsstruktur werden üblicherweise Umschaltprogramme verwendet.

Das verwendete Verkehrsmodell kann die *Änderung* der Signalpläne (inklusive evtl. erforderlicher Umschaltprogramme) und damit der Koordinierungsstruktur nicht abbilden, d.h. die sich daraus ergebenden Effekte können auch in der Optimierung nicht erkannt werden. Aus diesem Grund wird in dieser Arbeit darauf verzichtet, die globalen Versatzzeiten und damit die Koordinierungsstruktur mitzuoptimieren, die Kodierung lässt dies jedoch prinzipiell zu.

Allerdings stellt die Optimierung der Freigabezeiten bzw. T-Zeiten (also der Endzeitpunkte einer Phase) bis zu einem gewissen Grad eine *ingeschränkte Versatzzeitoptimierung* im Rahmen der vorgegebenen Intervalle $[T_{iB}^{\min}; T_{iB}^{\max}]$ dar. Die Intervalle sind üblicherweise zwischen 6 s und 20 s lang. Die Einschränkung muss kein grundsätzlicher Nachteil sein, sofern die zu Grunde liegende Koordinierungsstruktur zuvor gut geplant bzw. optimiert wurde. Zur Optimierung der Koordinierungsstruktur kann beispielsweise GALOP-Offline [BRAUN und WEICHENMEIER, 2005] eingesetzt werden.

Je nach Kodierung der Phasendauern bzw. Startzeitpunkte für die Phasenübergänge (siehe Abschnitt 4.3.6) ist eine *lokale Versatzzeit* erforderlich, die den lokalen Signalplan bezüglich der *lokalen* Zeitskala positioniert. Mit Hilfe der lokalen Versatzzeit können die Phasendauern in die Startzeitpunkte der Phasenübergänge (T-Zeiten) umgerechnet werden (lokale Zeitskala).

4.3.5 Phasenfolge

Die *Phasenfolge* definiert die Abfolge der Phasen. Üblicherweise stehen für jeden Knotenpunkt mehrere unterschiedliche Phasenfolgen zur Auswahl, die vorab festgelegt werden müssen. Alle Phasenfolgen sollten so definiert sein, dass sie sich in die geplante Koordinie-

rungsstruktur einfügen. Andernfalls besteht die Gefahr, dass eine Änderung der Phasenfolge an einem Knotenpunkt zu einer starken Änderung der Koordinierungsstruktur führt.

Die Phasenfolge wird über eine Umrechnung des Gens σ in eine Kennzahl ausgewählt, die einer der zuvor festzulegenden möglichen Phasenfolgen zugeordnet ist.

Bei den Phasenfolgen besteht das Problem, dass diese in vielen Fällen von der lokalen Steuerung bestimmt werden. Obwohl die Phasenfolge zentral mitoptimiert wird, kann es vorkommen, dass die lokale verkehrsabhängige Steuerung eine andere, netzweit betrachtet nicht optimale Phasenfolge auswählt. Um sicherzustellen, dass trotzdem für jede der möglichen Phasenfolgen aller Knotenpunkte ein guter Rahmensignalplan vorliegt, werden für alle Phasenfolgen die im Lauf der Optimierung jeweils besten Individuen gespeichert.

Um bei größeren Netzen trotz der Voraussetzung einer einheitlichen Umlaufzeit für alle LSA bei schwach belasteten Knoten keine unnötigen Wartezeiten zu erzeugen, könnte man halbe oder drittel Umlaufzeiten erlauben. Ohne die Kodierung zu verändern, können für die in Frage kommenden Knotenpunkte zusätzliche Phasenfolgen versorgt werden, die zwei- oder dreimal den gleichen Phasenablauf innerhalb der Umlaufzeit beinhalten.

4.3.6 Phasendauern bzw. Startzeitpunkte der Phasenübergänge

Die *Phasendauern* bzw. *Startzeitpunkte der Phasenübergänge* legen fest, wie lange die einzelnen Phasen und damit die Freigabezeiten der unterschiedlichen Verkehrsströme dauern.

Die Kodierung der Phasendauern bzw. der Startzeitpunkte der Phasenübergänge stellt sich am schwierigsten dar, da sich die genannten Randbedingungen primär darauf beziehen. Es wurden dafür mehrere Kodierungsmöglichkeiten entworfen, wovon sich zwei Kodierungsmöglichkeiten als gut geeignet erwiesen, die im Folgenden beschrieben werden: eine *relative* Kodierung und eine *sequentielle* Kodierung.

Relative Kodierung

Die relative Kodierung lässt die Randbedingungen der lokalen Steuerung zunächst unberücksichtigt. Danach berechnen sich die *Phasendauern* zu

$$P_i = \frac{\theta_i}{\sum_{l=1}^{k_\sigma} \theta_l} \cdot \left(t_U - \sum_{l=1}^{k_\sigma} Z_{\sigma,l} \right)$$

Z ist der minimale Abstand zwischen zwei benachbarten T-Zeiten, der sich aus der maßgebenden Summe von Zwischenzeit und Mindestfreigabezeit ergibt. Die Mindestphasendauern sind damit in dieser Formel bereits in Z enthalten, d.h. P_i ist nur der optimierbare

Anteil der Phase. k_σ ist die Anzahl der Phasen in der durch das Gen σ bestimmten Phasenfolge.

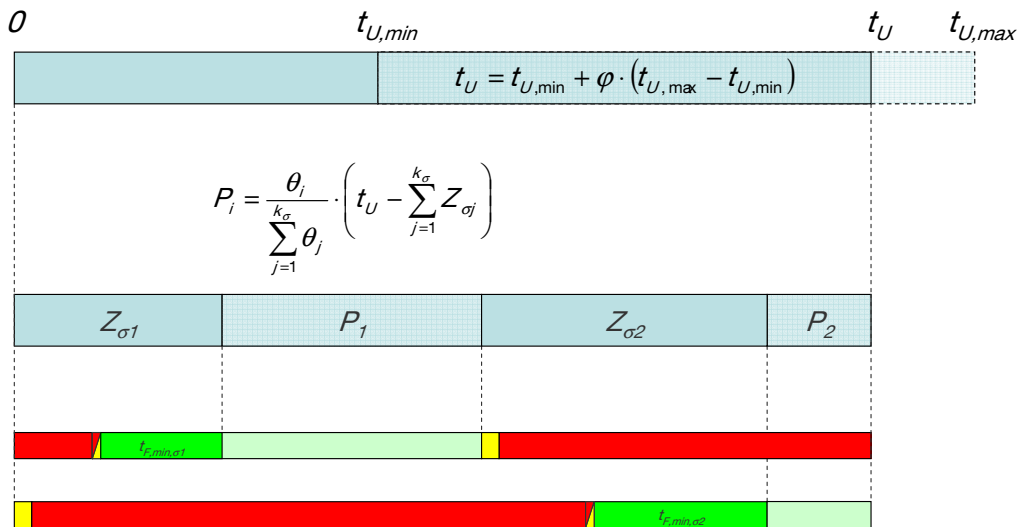


Abb. 4.4 Relative Kodierung von Umlaufzeit und Freigabezeiten

Für die Position des Signalplans bezüglich der lokalen Zeitskala – auf die sich auch T_{iB}^{\min} und T_{iB}^{\max} beziehen – ist ein zusätzliches Gen für den lokalen Versatz erforderlich:

$$T_{1B} = (T_{1B}^{\min} + o \cdot ((T_{1B}^{\max} - T_{1B}^{\min} + t_U) \bmod t_U)) \bmod t_U$$

Alle weiteren T-Zeiten T_{iB} berechnen sich dann in Abhängigkeit von T_{1B} und Z .

$$T_{iB} = \left(T_{1B} + \sum_{l=1}^{i-1} P_l + \sum_{l=1}^{i-1} Z_{\sigma,l} \right) \bmod t_U$$

Durch die relative Kodierung sind die zwingenden Randbedingungen immer eingehalten. Allerdings ist damit noch nicht sichergestellt, dass auch die Randbedingungen der lokalen verkehrsabhängigen Steuerung eingehalten werden. Deshalb werden die T-Zeiten für jedes Chromosom dahingehend überprüft. Für den Fall der Nichteinhaltung kommt ein Reparaturmechanismus zum Einsatz, der das Chromosom so verändert, dass die Randbedingungen der lokalen verkehrsabhängigen Steuerung eingehalten werden. Sofern die Reparatur nicht funktioniert, wird das Chromosom zufällig neu erzeugt.

Der *Reparaturmechanismus* funktioniert wie folgt: Für jede T-Zeit wird überprüft, ob sie sich innerhalb des Intervalls $[T_{iB}^{\min}; T_{iB}^{\max}]$ befindet. Ist das nicht der Fall, wird der gesamte Signalplan mit Hilfe des lokalen Versatzes o so verschoben, dass die ungültige T-Zeit der nächsten Grenze des Intervalls $[T_{iB}^{\min}; T_{iB}^{\max}]$ entspricht. Anschließend müssen *alle* T-Zeiten erneut überprüft werden. Sobald der Signalplan einmal verschoben wurde, darf er nur noch

in die gleiche Richtung weiter verschoben werden. Bei einem Widerspruch ist das Chromosom nicht zu reparieren und muss neu erzeugt werden. Der Signalplan darf immer nur komplett verschoben werden, da andernfalls nicht mehr sichergestellt wäre, dass die zwingenden Randbedingungen noch eingehalten sind.

Die Wahrscheinlichkeit einer erforderlichen Reparatur kann dadurch minimiert werden, dass für den lokalen Versatz statt T_{1B} diejenige T-Zeit herangezogen wird, für die das kleinste Intervall $[T_{iB}^{\min}; T_{iB}^{\max}]$ zur Verfügung steht.

Sequentielle Kodierung

Um ohne Reparatur auszukommen, bietet sich eine sequentielle Kodierung zur Umrechnung eines Individuums in einen Signalplan an, die die Randbedingungen der lokalen verkehrabhängigen Steuerung direkt mitberücksichtigt. Bei der sequentiellen Kodierung ist das Gen θ , also der lokale Versatz nicht notwendig, da nicht die Phasendauern, sondern die Startzeitpunkte der Phasenübergänge (T-Zeiten) direkt kodiert werden. In den folgenden Formeln wird aus Gründen der Übersichtlichkeit auf den Index B verzichtet (z.B. T_i statt T_{iB}).

Die *erste* T-Zeit T_1 , die dem lokalen Versatz entspricht, berechnet sich dabei direkt aus dem Gen θ_1 :

$$T_1 = (\tau_1^{\min} + \theta_1 \cdot (\tau_1^{\max} - \tau_1^{\min}))$$

Sie liegt also im Intervall

$$T_1 \in [\tau_1^{\min}, \tau_1^{\max}]$$

wobei sich die Intervallgrenzen τ_1^{\min} und τ_1^{\max} aus den vorhandenen T-Zeit-Grenzen-Bedingungen ergeben.

Für die untere Intervallgrenze τ_1^{\min} gilt:

$$\tau_1^{\min} = \max(T_1^{\min}, T_{1|2}^{\min}, T_{1|3}^{\min}, \dots, T_{1|n}^{\min})$$

mit

$$T_{1|j}^{\min} = T_j^{\min} + \sum_{l=j}^n Z_l \quad \text{für alle } j = 2..n$$

wobei $T_{1|j}^{\min}$ die Bedingung für die untere Intervallgrenze τ_1^{\min} ist, die sich aus der minimalen T-Zeit-Grenzen-Bedingung T_j^{\min} ergibt.

Für die obere Intervallgrenze τ_1^{\max} gilt:

$$\tau_1^{\max} = \min(T_1^{\max}, T_{1|2}^{\max}, T_{1|3}^{\max}, \dots, T_{1|n}^{\max})$$

mit

$$T_{1|j}^{\max} = T_j^{\max} - \sum_{l=1}^{j-1} Z_l \quad \text{für alle } j = 2..n$$

wobei $T_{1|j}^{\max}$ die Bedingung für die obere Intervallgrenze τ_1^{\max} ist, die sich aus der maximalen T-Zeit-Grenzen-Bedingung T_j^{\max} ergibt.

Alle weiteren T-Zeiten T_i werden sequentiell berechnet. Sie befinden sich jeweils im Intervall

$$T_i \in \left[\max(T_i^{\min}, T_{i-1} + Z_{i-1}), \min\left(T_i^{\max}, T_{i|i+1}^{\max}, T_{i|i+2}^{\max}, \dots, T_{i|n}^{\max}, T_1 + t_U - \sum_{l=i}^n Z_l\right) \right]$$

und werden dann analog wie T_1 über das zugehörige Gen θ_i berechnet.

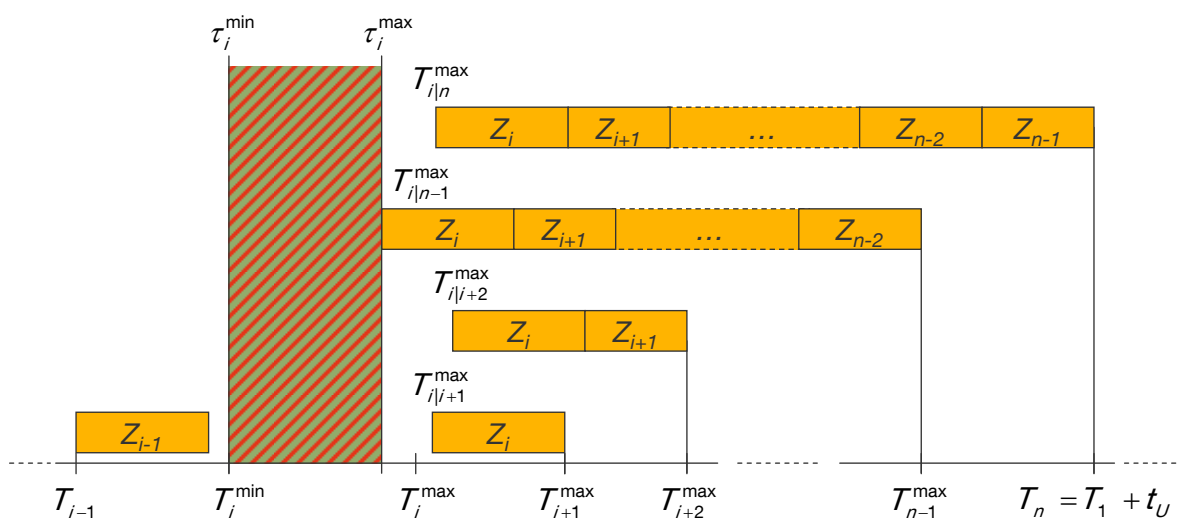


Abb. 4.5 Berechnung der Intervallgrenzen für T_i

Abb. 4.5 verdeutlicht die Einschränkung der Intervallgrenzen für T_i . Für die untere Intervallgrenze τ_i^{\min} ist das Maximum aus T_i^{\min} und der zuvor berechneten T-Zeit T_{i-1} zuzüglich des Mindestabstands Z_{i-1} zwischen T_{i-1} und T_i maßgebend. Für die obere Intervallgrenze τ_i^{\max} ist das Minimum aus T_i^{\max} , aller noch folgenden T-Zeit-Grenzen-Bedingungen abzüglich der dazwischen liegenden Mindestabstände, und $T_1 + t_U$ abzüglich aller noch fehlenden Mindestabstände.

Vergleich der beiden Kodierungen

Beide Kodierungen haben Vor- und Nachteile. Vorteil der relativen Kodierung ist die gleiche Bedeutung der θ_i innerhalb eines Chromosoms. Die einzelnen Phasendauern sind zwar durch die Summe der θ_i des Chromosoms voneinander abhängig, aber jedes θ_i hat das gleiche Gewicht. Diese Tatsache ist für die Mutation von Vorteil, sofern diese zur lokalen Suche eingesetzt wird.

Nachteil der relativen Kodierung ist die größere Zahl erforderlicher Reparaturen bzw. die Neuerzeugung von Chromosomen wegen der Nichteinhaltung der Randbedingungen der lokalen verkehrsabhängigen Steuerung. Dies ist gleichzeitig der Vorteil der sequentiellen Kodierung. Hier sind keine Reparaturversuche nötig. Sie hat jedoch den Nachteil, dass die einzelnen θ_i eines Chromosoms durch die sequentielle Auswertung eine unterschiedliche Bedeutung erhalten, da der Spielraum mit jedem ausgewerteten θ_i kleiner wird.

Beide Kodierungen sind grundsätzlich geeignet und werden im Rahmen der Implementierung (vgl. Kapitel 5.1) v.a. bezüglich des Rechenzeitbedarfs verglichen. Dabei stellt sich heraus, dass der Rechenzeitbedarf durch den erforderlichen Reparaturmechanismus bei der relativen Kodierung deutlich größer ist als bei der sequentiellen Kodierung.

4.4 Startpopulation

Die Startpopulation besteht in der Regel aus zufällig generierten Individuen. Durch die Zufälligkeit sollten die Individuen über den gesamten Suchraum einigermaßen gleichmäßig verteilt sein.

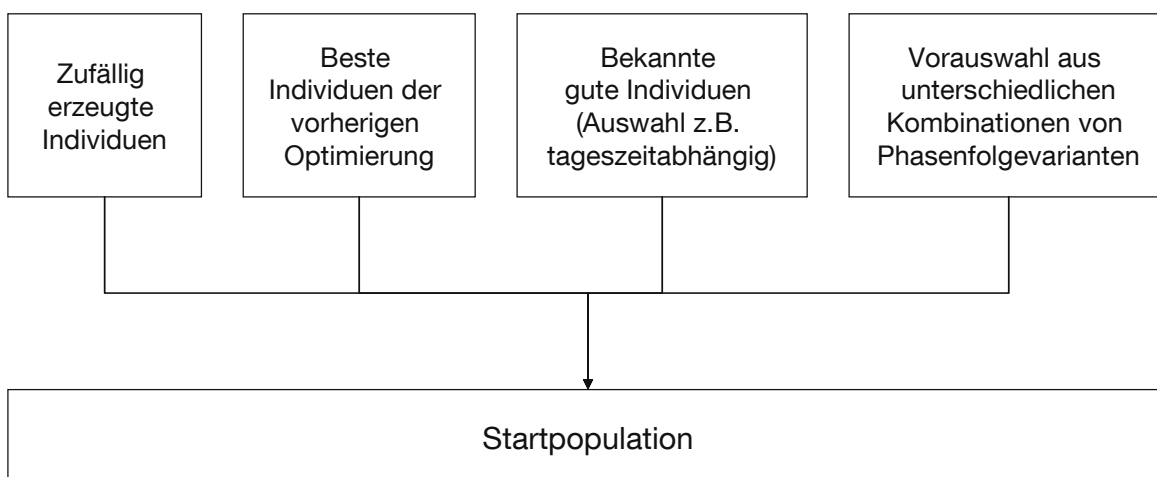


Abb. 4.6 Zusammensetzung der Startpopulation

Um jedoch bei beschränkter Rechenzeit schneller gute Lösungen zu erreichen, kann für die Online-Anwendung ein Teil der Startpopulation mit bekannten guten Lösungen oder mit guten Lösungen vorangegangener Optimierungsläufe bestückt werden. Um sicherzustellen, dass alle wichtigen Kombinationen von Phasenfolgevarianten angemessen in der Startpopulation berücksichtigt werden, kann ebenfalls eine Vorauswahl stattfinden.

4.5 Operatoren

Die Gestaltung der GA-Operatoren und deren Parametrierung haben einen großen Einfluss auf die Qualität des Optimierungsablaufs. Bei der Auswahl der Operatoren und deren Einstellung ist darauf zu achten, dass sich ein ausgewogenes Verhältnis zwischen Exploitation (lokale Suche) und Exploration (globale Suche) des Suchraums ergibt. Da ein Individuum alle Knotenpunkte im Verkehrsnetz repräsentiert, müssen die Operatoren auf die Kodierung abgestimmt werden. Wie oben beschrieben wird ein Knotenpunkt im Individuum als Satz von Genen (Chromosom) dargestellt. In den folgenden Abschnitten werden die Funktions- und Wirkungsweisen der in dieser Arbeit verwendeten und zum Teil erweiterten Operatoren allgemein und bezogen auf die Problemstellung erläutert.

4.5.1 Selektion

Die Selektion dient der Ermittlung der Eltern, d.h. derjenigen Individuen der aktuellen Generation, die zur nächsten Generation beitragen dürfen. Sie reduziert den potentiellen Überschuss an möglichen Lösungen, betont gute Lösungskandidaten, erzeugt selbst jedoch keine neuen Individuen.

Die Selektion steuert die Evolution und hat deshalb großen Einfluss auf den Verlauf der Konvergenz. Je nach Auswahl und Parametrierung des Selektionsoperators kann die Selektion im Suchprozess sowohl zufällig als auch gerichtet wirken. Häufig verwendete Varianten der Selektion sind:

- Abschneideselektion
- Rouletteselektion
- Stochastic Universal Sampling
- Turnierselektion

Bei der Abschneideselektion werden die besten x Prozent ausgewählt. Dieses Verfahren hat den Nachteil eines sehr künstlichen Vorgehens, was der Grundidee eines GA als Nachbildung natürlicher Vorgänge widerspricht. Bei der Rouletteselektion und beim Stochastic Universal Sampling findet eine fitnessproportionale Selektion statt, d.h. die Individuen werden entsprechend ihres absoluten Fitnesswerts ausgewählt. Die Rouletteselektion funktioniert nach dem Prinzip *Glücksrad*. Jedes Individuum bekommt einen Sektor auf dem Glücksrad wobei die Größe des Sektors von seiner Fitness abhängt. Bessere Individuen

bekommen größere Sektoren. Das Glücksrad wird so oft gedreht, bis die Anzahl benötigter Individuen selektiert wurde. Das Stochastic Universal Sampling funktioniert ähnlich, wobei so viele Zeiger wie Individuen ausgewählt werden sollen, gleichmäßig um das Glücksrad verteilt werden und dieses nur einmal gedreht wird. Eine genauere Beschreibung findet sich z.B. in POHLHEIM [1999].

Bei der Turnirselektion werden jeweils (mindestens) zwei zufällig ausgewählte Individuen miteinander verglichen und das bessere wird als Elternteil ausgewählt (Prinzip: Ziehen mit Zurücklegen).

Da es sich bei der verwendeten Zielfunktion um die gewichtete Summe unterschiedlicher Größen handeln kann, ist in dieser Arbeit die *Turnirselektion* die Selektionsmethode der Wahl, weil sie nur die Rangfolge der Fitnesswerte berücksichtigt und nicht die absoluten Fitnesswerte selbst.

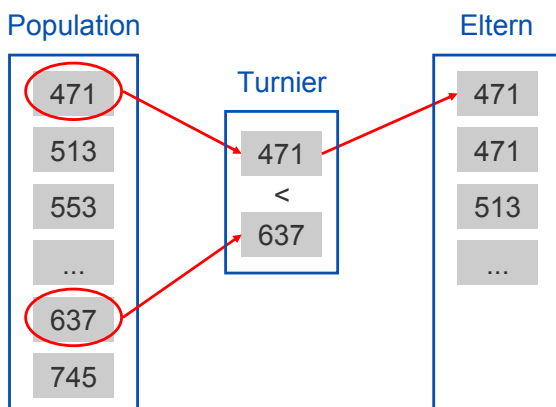


Abb. 4.7 Funktionsweise der Turnirselektion

Um sicherzustellen, dass die beste Lösung einer Generation nicht verloren geht, kommt in dieser Arbeit zusätzlich die sog. *Eliteselektion* zum Einsatz. Das bedeutet, dass die k ($k > 0$) besten Individuen unverändert in die nächste Generation übernommen werden. Allerdings erhöht dieses Konzept auch die Gefahr der vorzeitigen Konvergenz, d.h. das Erreichen eines lokalen Optimums. Deshalb müssen die Evolutionären Operatoren so verwendet bzw. deren Parameter so eingestellt werden, dass eine ausreichende Diversifikation der Population sichergestellt wird.

Der Selektionsdruck kann über die *Anzahl der Nachkommen* gesteuert werden, die pro Generation erzeugt werden. Werden mehr Nachkommen als die Populationsgröße erzeugt, wird beispielsweise durch eine Abschneideselektion wieder die ursprüngliche Populationsgröße hergestellt, bevor die nächste Generation erzeugt wird. Das Vorgehen entspricht einer sog. Umweltselektion, die in dieser Arbeit ebenfalls zum Einsatz kommt. Je größer die Anzahl der Nachkommen, desto höher ist der Selektionsdruck.

4.5.2 Rekombination

Der Operator Rekombination kombiniert die einzelnen Elternindividuen zu neuen Individuen. Er benötigt (mindestens) zwei Eltern, deren Gene und/oder Chromosomen neu kombiniert werden. Die Rekombination sorgt durch das Mischen des Erbguts zum Einen für eine zufällige und ungerichtete Suche im Lösungsraum, zum Anderen besteht die Chance, dass gute Ansätze kombiniert werden. Folgende Rekombinationsvarianten werden häufig verwendet [vgl. z.B. POHLHEIM, 1999]:

- Diskrete Rekombination (uniform crossover)
- Intermediäre Rekombination

Bei der diskreten Rekombination wird für jedes Gen des Nachkommen entschieden, von welchem Elter das Gen verwendet wird. Bei der intermediären Rekombination wird aus zwei Elterngenen ein kombinierter Wert (z.B. Mittelwert) gebildet. Insbesondere bei den Evolutionsstrategien werden die Rekombinationsvarianten häufig *global* verwendet. D.h. für jedes einzelne Gen werden zwei neue Eltern gesucht, es kommt also zu einer rein zufälligen Neuzusammensetzung aus dem insgesamt vorhandenen Genmaterial.

Im Rahmen dieser Arbeit werden zwei Rekombinationsvarianten ergänzt, die chromosomweise Rekombination und die Rekombination nach lokaler Fitness.

Chromosomweise Rekombination

Standardmäßig werden bei der Rekombination von zwei Eltern-Individuen *einzelne Gene* zur Erzeugung der Nachwuchs-Individuen verwendet. Um Freigabezeitaufteilungen an einem Knotenpunkt weitgehend zu erhalten, wird die Rekombination deshalb dahingehend erweitert, dass mit einer Wahrscheinlichkeit p die Rekombination auf die vollständigen Chromosomen (Knotenpunkte) angewendet wird. Damit wird im Fall der relativen Kodierung gleichzeitig vermieden, dass die Rekombination zu Chromosomen führt, die bezüglich der Randbedingungen der lokalen verkehrsabhängigen Steuerung ungültig wären und einem Reparaturversuch unterzogen werden müssten.

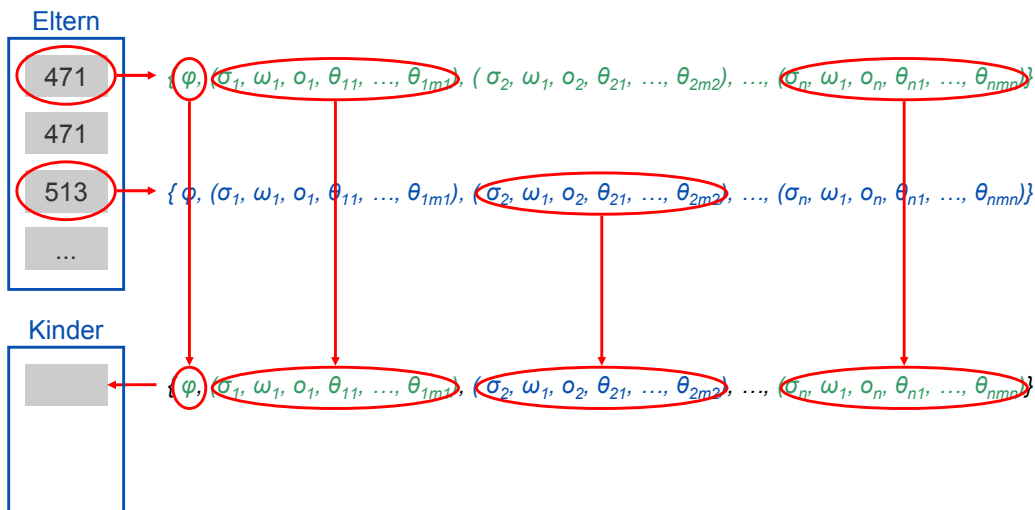


Abb. 4.8 Prinzip der chromosomweisen Rekombination

Rekombination nach lokaler Fitness

Als weitere Rekombinationsvariante kann ein Teil der Nachkommen nach der jeweiligen lokalen Fitness (Wirkungskenngrößen eines Knotenpunkts) der Steuerung der einzelnen Knotenpunkte (Chromosomen) rekombiniert werden. D.h. für jedes Elternpaar gibt es genau einen Nachkommen, der – mit Ausnahme der Umlaufzeit – das jeweils bessere Chromosom erhält. Die Umlaufzeit wird, sofern sie optimiert werden soll, mit je 50% Wahrscheinlichkeit von Elter 1 oder Elter 2 übernommen. Obwohl die lokale Fitness auch von den Chromosomen der Nachbarknoten abhängt, setzt eine gute lokale Fitness zumindest eine geeignete Aufteilung der Phasendauern voraus, die bei dieser Rekombinationsvariante mit einer höheren Wahrscheinlichkeit in die nächste Generation vererbt wird.

4.5.3 Mutation

Durch die Mutation werden die Genwerte des Individuums mit einer bestimmten Wahrscheinlichkeit verändert. Dabei können zwei Ziele verfolgt werden:

- Lokale Optimierung
- Vielfältigkeit in der Population

Im ersten Fall wird die Mutation durch das Erzeugen kleiner Änderungen zur lokalen Suche verwendet. Im zweiten Fall wird durch große Änderungen versucht, lokale Optima wieder zu verlassen. Die Mutation kann demnach, sofern eine reellwertige Kodierung vorliegt, sowohl zur gerichteten als auch zur ungerichteten Suche eingesetzt werden. Im Fall einer binären Kodierung besteht die Mutation immer aus einer Änderung eines Gens von 0 zu 1 oder umgekehrt, was zur Folge hat, dass sich die Mutation nicht an der phänotypischen Nachbarschaft orientiert. Dies ist einer der Hauptgründe, dass eine reellwertige Kodierung gewählt

wurde. Grundsätzlich ist die Mutation unabhängig von der Anzahl der Eltern einsetzbar, da sie aus einem Individuum ein neues erzeugt.

Um eine Konvergenz im gesuchten Optimum unabhängig von der Anfangspopulation zu ermöglichen, sollte eine Erreichbarkeit aller Punkte im Suchraum durch den Mutationsoperator gegeben sein. [WEICKER, 2007]

Folgende Mutationsvarianten kommen sehr häufig zum Einsatz und werden auch in dieser Arbeit genutzt:

- Uniforme Mutation innerhalb eines vorgegebenen Intervalls
- Normalverteilte Mutation

Bei der uniformen Mutation wird innerhalb eines vorgegebenen Intervalls $[X - \varepsilon, X + \varepsilon]$ zufällig ein neuer Wert für das zu mutierende Gen erzeugt. Bei der normalverteilten Mutation liegt für einen zum Originalwert zu addierenden oder subtrahierenden Zufallswert eine Normalverteilung zu Grunde. Die Mutationsschrittweite wird im ersten Fall über die Größe des Intervalls ($2 \cdot \varepsilon$), im zweiten Fall über die Standardabweichung σ der Normalverteilung eingestellt.

4.5.4 Parameteradaption

Im Lauf einer Optimierung ist meist von Vorteil, unterschiedliche Schrittweiten in der Mutation zu verwenden. Hierzu kann man einen festen Plan vorgeben, wofür aber ein guter Überblick über den Verlauf der Optimierung erforderlich ist. Eine andere Möglichkeit besteht darin, dass sich die Schrittweiten der Mutationen je nach Optimierungszustand adaptiv anpassen. Dies kann entweder über den Einsatz von konkurrierenden Unterpopulationen erreicht werden [vgl. POHLHEIM, 1999], oder mittels zusätzlicher Variablen im Individuum, die die adaptierten Schrittweiten speichern. Letzterer Ansatz wird im Rahmen dieser Arbeit genutzt.

Die zusätzlichen Variablen sollen eine Rückkopplung vom Verlauf der Optimierung zur Wirkungsweise der Operatoren erlauben und unterliegen selbst einer Mutation. Die Idee ist, dass die durch vorteilhafte Mutationen entstandenen Nachkommen die Information über die Schrittweitenvariable eher an nachfolgende Generationen weitergeben, weil sie eine bessere Fitness haben. Die Schrittweitenvariable wird somit indirekt bewertet.

Die normalverteilte Mutation ist dafür besonders gut geeignet, weil die Standardabweichung der Verteilung als einfacher Regler für eine Anpassung der Mutationsschrittweite zur Verfügung steht. In der einfachen Variante wird jedes Individuum um eine zusätzliche Variable für die Mutationsschrittweite ergänzt. Es ist aber auch möglich, für jedes einzelne Gen eine eigene Schrittweitenvariable zu verwenden. SCHWEFEL [1995] kritisiert allerdings, dass für eine große Anzahl von Variablen ($n \gg 1$) wegen des Gesetzes der großen Zahlen die mittlere

re Gesamtschrittweite von Eltern- und Nachkommenvektor nicht substantiell voneinander abweichen kann.

4.6 Verjüngung

Zur Vermeidung vorzeitiger Konvergenz im Optimierungsablauf kann bei zu geringen Fitnesswertdifferenzen innerhalb der Population ein Teil der Individuen durch zufällig neu generierte Individuen ersetzt werden (Verjüngung). Geringe Fitnessdifferenzen deuten darauf hin, dass sich die Individuen sehr ähnlich sind, wodurch die Gefahr besteht, dass die Population in einem lokalen Optimum hängen bleibt. Durch die frischen Individuen werden wieder andere Bereiche des Lösungsraums durchsucht. Diese Option wird für GALOP genutzt.

4.7 Abbruchkriterien

Um das Ende der Optimierung zu bestimmen, ist mindestens ein Abbruchkriterium erforderlich. Folgende Abbruchkriterien kommen häufig zum Einsatz:

- Maximale Anzahl von Generationen
- Maximale Rechenzeit
- Finden einer Lösung mit vorgegebener Mindestgüte
- Keine Verbesserung der besten Lösung innerhalb der letzten x Generationen
- Unterschreitung einer vorgegebenen mittleren Mindestverbesserung in den letzten x Generationen
- Konvergenz der Population

Die *maximale Anzahl von Generationen* und die *maximale Rechenzeit* sind direkte Abbruchkriterien. Mindestens eines dieser beiden Kriterien sollte immer gesetzt sein, um nicht in einer Endlosschleife zu landen, falls andere Kriterien nicht greifen. Die maximale Rechenzeit ist primär für Echtzeitanwendungen relevant und wird von der Vorgabe bestimmt, innerhalb welcher Zeit spätestens ein Ergebnis vorliegen muss.

Sofern man nur eine bekannte Mindestgüte erreichen will, kann die Optimierung abgebrochen werden, wenn diese erreicht ist. Außerdem können Kriterien festgelegt werden, die von der Konvergenz des Optimierungsablaufs abhängen. So kann ein Abbruch stattfinden, wenn sich die beste Lösung über mehrere Generationen hinweg nicht verbessert, oder wenn eine mittlere Mindestverbesserung über die letzten Generationen unterschritten wird. Ebenso kann die Konvergenz der gesamten Population z.B. über die Standardabweichung der Fitnesswerte oder den Vergleich der besten mit der schlechtesten Lösung einer Generation als Abbruchkriterium herangezogen werden. Das Heranziehen der gesamten Population hat jedoch den Nachteil, dass der Abbruch von der Skalierung der Zielfunktion abhängt. Außerdem können wenige schlechte Zielfunktionswerte das Erreichen des Abbruchwerts ver-

hindern. Alternativ kann die Konvergenz der Population über die Ähnlichkeit der Genwerte der Individuen überprüft und als Abbruchkriterium festgelegt werden.

Für GALOP sind als Abbruchkriterien die maximale Anzahl von Generationen und die maximale Rechenzeit relevant.

4.8 Zwischenfazit

In den vorangegangenen Abschnitten wurden die wesentlichen Merkmale und Bestandteile des entwickelten Evolutionären Algorithmus GALOP dargestellt. Die wesentliche Leistung beim Entwerfen eines Evolutionären Algorithmus besteht in der Kodierung und der Anpassung der Standardoperatoren. Es wurden zwei Kodierungsvarianten entworfen, die die problemspezifischen Randbedingungen berücksichtigen. Neben planerischen Vorgaben wie erlaubte Umlaufzeiten und erlaubte Phasenfolgen sind dies zwingende Randbedingungen (Zwischenzeiten, Mindestfreigabedauern) und Randbedingungen, die sich aus der lokalen verkehrsabhängigen Steuerung ergeben. Um die Freigabezeitaufteilungen an einem Knotenpunkt weitgehend zu erhalten, wurde zusätzlich zu den Standard-Rekombinationsoperatoren die chromosomweise Rekombination ergänzt.

5 Bewertung des Optimierungsverfahrens GALOP

In diesem Kapitel wird der Evolutionäre Algorithmus bewertet. Wie bereits in Abschnitt 4.2 zur Entwurfsmethodik erwähnt, handelt es sich beim Entwurf eines EA um einen iterativen Prozess. Eine separate Betrachtung von Entwicklung, Umsetzung und Bewertung eines Algorithmus ist deshalb eher theoretischer Natur. Bereits während der Entwicklung ist eine ständige Bewertung erforderlich.

In Abschnitt 5.1 wird die Umsetzung des Algorithmus erläutert. Die verwendete Bewertungsmethodik wird in Abschnitt 5.2 dargestellt, Abschnitt 5.3 erläutert den Versuchsaufbau und die Vorgehensweise bei der Bewertung. Vor der eigentlichen Bewertung in den Abschnitten 5.5 bis 5.8 wird in Abschnitt 5.4 die Güteverteilung im Suchraum untersucht. Schließlich wird in Abschnitt 5.9 ein Fazit der Bewertung gezogen.

5.1 Umsetzung des Algorithmus

Die Umsetzung des Evolutionären Algorithmus fand im Rahmen des Forschungsprojekts TRAVOLUTION² statt. Das bisher in der adaptiven Netzsteuerung BALANCE verwendete Optimierungsverfahren (Hill-Climbing Algorithmus) wurde durch GALOP ersetzt.

GALOP wurde auf Basis der C++ Bibliothek EO-Lib (Evolving Objects) [KEIJZER et al., 2001] aufgebaut. Die Vorteile dieser Bibliothek liegen in der freien Wahl der Repräsentation der Individuen, des objektorientierten Programmieransatzes und der Möglichkeit zur einfachen Parallelisierung des Algorithmus. Die Bibliothek unterstützt zahlreiche Varianten unterschiedlicher Evolutionärer Algorithmen und kann relativ leicht erweitert werden. Die einzelnen GA-Parameter können über eine Textdatei eingegeben werden.

Als zusätzliche Rekombinationsvariante wurde die chromosomweise Rekombination ergänzt (vgl. Abschnitt 4.5.2). Nicht umgesetzt wurde die Rekombination nach lokaler Fitness. Über die Textdatei kann festgelegt werden, mit welcher Wahrscheinlichkeit die Individuen einer Generation jeweils durch die unterschiedlichen Rekombinationsvarianten entstehen.

Für die Mutation stehen ebenfalls mehrere Varianten zur Verfügung. Sie spielt eine wesentliche Rolle für die adaptiven Elemente des Algorithmus. Die Mutationsschrittweiten einer normalverteilten Mutation können über einen oder mehrere zusätzliche Parameter im Indivi-

² TRAVOLUTION war ein vom Freistaat Bayern gefördertes Forschungsprojekt, das von den Projektpartnern AUDI AG, GEVAS software GmbH und Lehrstuhl für Verkehrstechnik der TU München in Zusammenarbeit mit dem Amt für Verkehrsmanagement und Geoinformation der Stadt Ingolstadt durchgeführt wurde

duum festgelegt werden, die ebenfalls der Evolution unterliegen. Ziel ist es, dass der Algorithmus dadurch robuster wird, weil sich gute Mutationsschrittweiten im Lauf der Optimierung durchsetzen (vgl. Abschnitt 4.5.4).

Ebenfalls ergänzt wurde die Option der Verjüngung, bei der bei zu geringer Varianz der Fitnesswerte in der Population ein Teil der Population durch frische zufällig generierte Individuen ersetzt wird (vgl. Abschnitt 4.6).

Bei GALOP besteht zudem die Möglichkeit, bereits als gut bekannte Individuen, die bei einer bestimmten Verkehrslage sinnvolle Signalpläne repräsentieren, einzulesen und weiterzuentwickeln, was gerade im Hinblick auf den eng bemessenen Optimierungszyklus von Vorteil ist.

Als Kodierungsvariante wurde nur die sequentielle Kodierung vollständig umgesetzt (vgl. Abschnitt 4.3.6). Während der Umsetzung der relativen Kodierung erwies sich der Rechenzeitbedarf durch den erforderlichen Reparaturmechanismus als zu groß, weshalb auf eine vollständige Umsetzung verzichtet wurde.

5.2 Bewertungsmethodik

Die Entwicklung und die Bewertung eines Algorithmus können nicht getrennt werden. Im Zuge der Entwicklung müssen die unterschiedlichen Formen des Algorithmus getestet werden, wobei die gleichen Kriterien zur Anwendung kommen wie bei der eigentlichen abschließenden Bewertung.

Notwendige Tests bei der Entwicklung eines Evolutionären Algorithmus teilen GERDES et al. [2004] in drei verschiedene Klassen ein:

- Parameteradaption
- Leistungsfähigkeit
- Vergleich mit anderen Optimierungsmethoden

Bei der *Parameteradaption* werden unterschiedliche Parametereinstellungen bei sonst identischen Algorithmen getestet. Der Test auf *Leistungsfähigkeit* kann sich auf eine spezielle Aufgabe beziehen, wie z.B. ein spezielles Netz bzw. eine bestimmte Netzgröße in Kombination mit einer speziellen Verkehrsnachfragestruktur. Er kann sich aber auch auf unterschiedliche Aufgaben beziehen. Dadurch kann getestet werden, ob der Algorithmus für unterschiedliche Netze und Verkehrsnachfragestrukturen genauso leistungsfähig ist, oder ob der Algorithmus für unterschiedliche Netzgrößen und Nachfragestrukturen unterschiedlich sein sollte. Um festzustellen, wie der Algorithmus im *Vergleich mit anderen Optimierungsmethoden* abschneidet, ist mindestens eine Referenzmethode erforderlich, der die gleiche Zielfunktion und das gleiche Verkehrsmodell verwendet sowie den gleichen Randbedingungen

unterliegt. Als Referenz steht hierfür der bisher in BALANCE enthaltene Hill-Climbing Algorithmus zur Verfügung.

5.2.1 Bewertungskriterien

Für die Bewertung von GALOP werden folgende *Bewertungskriterien* herangezogen:

- Qualität
- Konsistenz
- Robustheit
- Effizienz

Qualität

Für die Beurteilung der Qualität der Bestlösung ist eine Referenzgröße nötig, mit der die erzielte Güte des Optimierungsergebnisses verglichen werden kann. Da das tatsächliche Optimum nicht bekannt ist, dient das Ergebnis des in BALANCE implementierten Hill-Climbing Algorithmus als Referenz. Als unterer Referenzwert kann das Ergebnis einer Zufallssuche dienen.

Durch einen sehr langen, mehrstündigen Optimierungslauf mit gut eingestellten Parametern kann zudem eine sehr gute (quasi-optimale) Lösung gefunden werden, die als obere Referenz für die kurzen, im Minutenbereich liegenden Optimierungsläufe dienen kann.

Konsistenz

Die Unterschiedlichkeit der Optimierungsergebnisse mehrerer unabhängiger Optimierungsläufe mit unterschiedlichen *Startzufallszahlen* aber ansonsten gleichen Bedingungen spiegelt die Konsistenz der Lösungsqualitäten des Algorithmus wider.

Robustheit

Unter Robustheit bzw. Sensitivität versteht man die Zuverlässigkeit, mit der das Optimum bzw. eine gute Lösung gefunden wird, wenn sich Umgebungsbedingungen ändern. Zu den Umgebungsbedingungen zählen z.B. die Startpopulation und der Einsatz der Evolutionären Operatoren und deren Parameter (Parametersensitivität). Ein Algorithmus ist umso robuster, je weniger sensitiv er sich gegenüber den Umgebungsbedingungen verhält.

Effizienz

Die Effizienz der Lösungsermittlung eines Algorithmus ergibt sich aus der Zeit, die erforderlich ist, um eine ausreichend gute Lösung bzw. das Optimum zu erreichen. Der Zeitbedarf

hängt im Wesentlichen von der Anzahl der dazu benötigten Lösungsbewertungen und vom Rechenaufwand pro Lösung (Individuum) ab.

5.2.2 Bewertungsmetriken

Für die Bewertung der Leistung eines Evolutionären Algorithmus stehen u.a. folgende *Metriken* zur Verfügung [vgl. WEICKER, 2007]:

- Beste Güte
- Mittlere beste Güte
- Durchschnittliche Güte
- Rechenzeit

Die *beste Güte* ist der beste Gütewert in allen betrachteten Generationen. Sie ist der Hauptindikator für die *Qualität* des Algorithmus.

Die *mittlere beste Güte* berechnet sich aus den besten Gütewerten jeder Generation. Sie ist ein Indikator für die Konvergenzgeschwindigkeit. Eine hohe *Konvergenzgeschwindigkeit* deutet meist auch auf einen effizienten Algorithmus hin. Je schneller der Algorithmus konvergiert, desto früher wird normalerweise auch eine ausreichend gute Lösung erreicht. Allerdings besteht bei zu hohen Konvergenzgeschwindigkeiten die Gefahr, dass nur eines der nächstbesten lokalen Optima erreicht wird.

Die *durchschnittliche Güte* aller bewerteten Individuen berücksichtigt neben der Konvergenzgeschwindigkeit die Anzahl der evaluierten Individuen.

Die *Rechenzeit* ist insbesondere für die Online-Anwendung von Bedeutung. Sie hängt wesentlich von der Anzahl zu bewertender Individuen und der benötigten Zeit pro Individuum ab, was wiederum stark durch den verwendeten Rechner bestimmt wird. Da das Ziel ist, alle fünf Minuten ein neues Optimierungsergebnis zu erreichen, darf der Algorithmus diese Zeit nicht überschreiten.

Eine weitere häufig vorgeschlagene Metrik ist die *Fitness-Distanz-Korrelierung*. Sie beschreibt die Beziehung zwischen der Fitness eines jeden Individuums und der jeweiligen Distanz der Individuen zum besten Individuum. Die Fitness-Distanz-Korrelation soll als Maß für die Bewertung des Zusammenhangs zwischen der Fitness eines Individuums und seinem Abstand zum besten Individuum dienen. Die Distanz zwischen den Individuen im Lösungsraum soll dabei die Ähnlichkeit widerspiegeln. Genau hier liegt allerdings auch das Problem. Die Aussagekraft der euklidischen Distanz der Genwerte zweier Individuen geht bei komplizierten, hochdimensionalen Lösungsräumen gegen null [vgl. z.B. POHLHEIM, 1999]. Aus diesem Grund wird die Fitness-Distanz-Korrelierung in dieser Arbeit nicht als Bewertungsmetrik herangezogen.

5.3 Versuchsaufbau und Vorgehensweise

GALOP wird an Hand von drei realen Netzen in Ingolstadt bewertet. Die drei Netze sind in Abb. 5.1 dargestellt.

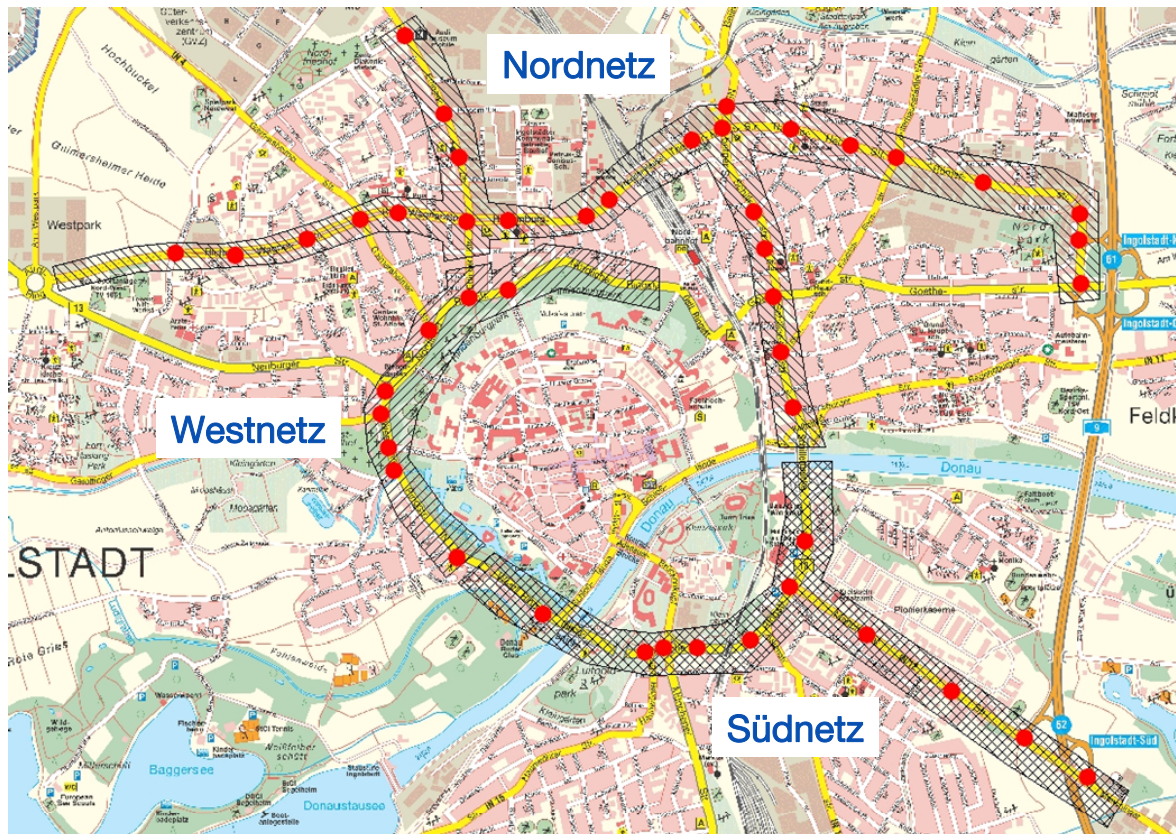


Abb. 5.1 Drei Netze in Ingolstadt (Karte: Amt für Verkehrsmanagement und Geoinformation, Stadt Ingolstadt)

Das Nordnetz hat 27 LSA auf drei Streckenzügen. Das Südnetz besteht aus zwei Streckenzügen mit insgesamt 11 LSA. Das Westnetz ist ein Streckenzug mit 8 LSA.

Obwohl der Algorithmus die Umlaufzeit mitoptimieren kann, wird hier die durch die vorhandenen Programme vorgegebene Umlaufzeit übernommen, die je nach Programm und Teilnetz bei 70 s oder 90 s liegt.

Getestet wird jeweils für ein Optimierungsintervall. Die für das Verkehrsmodell zu Grunde gelegte Verkehrsnachfrage ergibt sich aus den jeweiligen Zählwerten der letzten 15 min. Es wird für jedes Nachfrageszenario eine Mikrosimulation mit NONSTOP durchgeführt. Nach mindestens 15 min Vorlauf, um das gesamte Netz zu füllen, werden 15 min als Verkehrsnachfrageszenario aufgezeichnet. Die aufgezeichneten Zählwerte der Nachfrageszenarien stehen dann zum Testen für jedes der drei Netze offline zur Verfügung. Während der Aufzeichnung läuft eine Basissteuerung. Dabei handelt es sich um die Festzeitsteuerung, die

bisher in Ingolstadt als Grundlage für die lokale verkehrsabhängige Steuerung verwendet wird.

Tab. 5.1 gibt einen Überblick über die getesteten Kombinationen von Netzen und Verkehrsnachfrageszenarien und deren Verwendung für die Bewertung.

Nachfrageszenario	Nordnetz	Südnetz	Westnetz
Morgenspitze (P2)	Konsistenz Robustheit Effizienz Qualität	Robustheit Effizienz Qualität	Qualität
Abendspitze (P4)	Robustheit Effizienz Qualität	Qualität	Qualität
Schwachlast (P1)	Qualität	Qualität	Qualität
Morgenspitze doppelt (P2-doppelt)	Qualität	Qualität	Qualität

Tab. 5.1 Testszzenarien für Kalibrierung, Sensitivitätsuntersuchung und Bewertung

Die *Robustheit* des Algorithmus wird mittels einer *Sensitivitätsanalyse* untersucht. Diese dient zum einen dazu, den Einfluss der Änderung einzelner Parameter des Algorithmus auf das Ergebnis abzuschätzen (Parametersensitivität), zum anderen kann damit festgestellt werden, welche Auswirkungen andere Randbedingungen, wie eine andere Verkehrsnachfrage oder ein anderes Netz, auf das Verhalten des Algorithmus haben. Die Robustheit wird für das Nordnetz mit den Nachfrageszenarien Morgen- und Abendspitze, sowie für das Südnetz mit dem Nachfrageszenario Morgenspitze untersucht. Auf diese Weise wird der Algorithmus gleichzeitig kalibriert.

Die *Qualität der Bestlösung* wird für alle zwölf Fälle im Vergleich zum Ergebnis des Hill-Climbing Algorithmus, zu einer quasi-optimalen³ Lösung, die mit Hilfe eines mehrere Stunden dauernden Optimierungslaufs mit gut eingestellten Parametern ermittelt wird, und im Vergleich zum Gütewert der bei der Aufzeichnung gelaufenen Basissteuerung, bewertet.

Als Zielfunktion wird eine gewichtete Wartezeitsumme verwendet. Für jede Signalgruppe wird vom Verkehrs- bzw. Wirkungsmodell die Wartezeit aller betroffenen Fahrzeuge in Sekunden berechnet, die für das Netz zusammenaddiert wird. Bis auf wenige Ausnahmen

³ Die optimale Lösung könnte nur durch vollständige Enumeration ermittelt werden, was jedoch auf Grund des astronomisch großen Lösungsraums (vgl. Abschnitt 1.2) nicht möglich ist.

werden die Wartezeiten der einzelnen Signalgruppen einfach gewichtet, einige wichtige Signalgruppen werden dreifach gewichtet. Für das Optimierungsverfahren spielt das allerdings keine große Rolle, da es lediglich den einen skalaren Wert pro Individuum benötigt.

5.4 Güteverteilung im Suchraum

Zur Abschätzung der Güteverteilung im Suchraum wird eine extensive Zufallssuche durchgeführt, d.h. es werden für das Nordnetz und das Nachfrageszenario Morgenspitze 100 000 Individuen zufällig erzeugt und deren Fitness berechnet. In Abb. 5.2 ist die Verteilung der Gütewerte dargestellt.

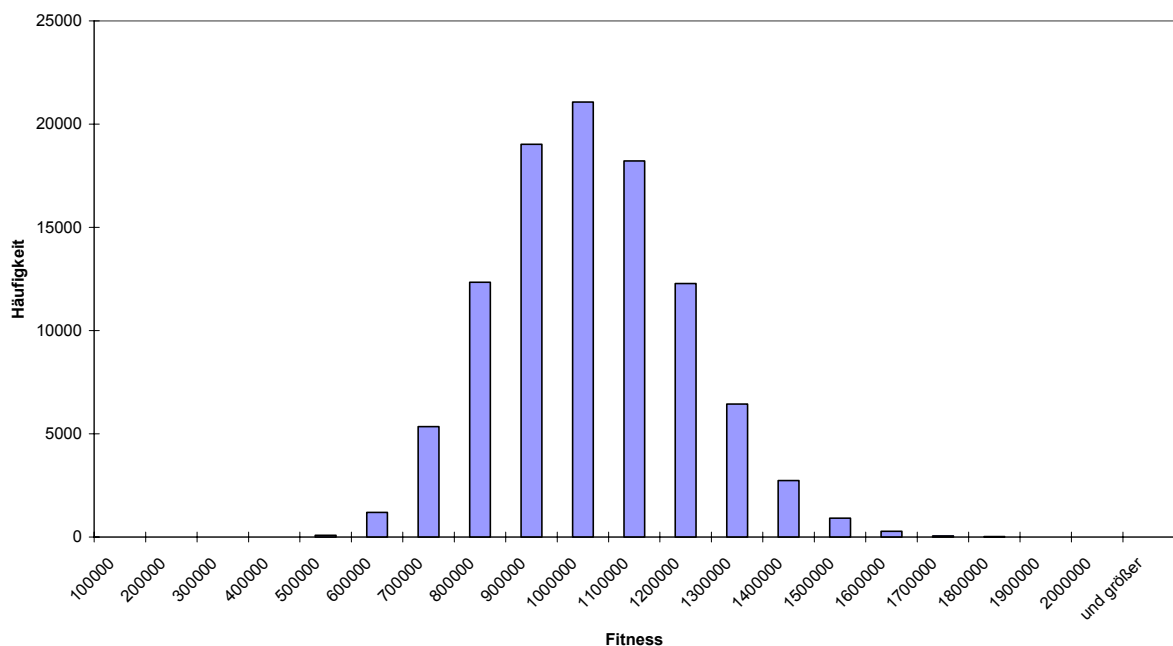


Abb. 5.2 Histogramm der Fitnesswerte von 100 000 zufällig erzeugten Individuen

Im Durchschnitt wird durch die Zufallssuche eine Fitness von 965 046 erreicht, die Standardabweichung liegt bei 185 739. Das schlechteste Individuum hat eine Fitness von 11 778 600. Die beste zufällig erreichte Fitness lag bei 416 980 und damit in der Größenordnung der Güte, die die Basissteuerung erreicht (456 595). Die beste durch einen mehrstündigen Optimierungslauf mit GALOP erreichte Lösung liegt bei 238 983, unter normalen Bedingungen erreicht der kalibrierte GALOP einen Gütewert von 254 918. Der Hill-Climbing Algorithmus kommt auf 355 552.

Neben dem gewonnenen Eindruck über die Güteverteilung im Suchraum wird damit gleichzeitig deutlich, dass eine Zufallssuche keine Alternative zum Evolutionären Algorithmus oder zum Hill-Climbing Algorithmus ist.

5.5 Einfluss der Startzufallszahl (Konsistenz)

Um den Einfluss der Startzufallszahl für die Erzeugung der Startpopulation und damit die *Konsistenz* des Algorithmus abzuschätzen, wird eine für gut befundene Variante des Algorithmus 40 mal mit verschiedenen Startzufallszahlen (ZZ) ausgeführt. Abb. 5.3 gibt einen Eindruck über die Streuung der Ergebnisse.

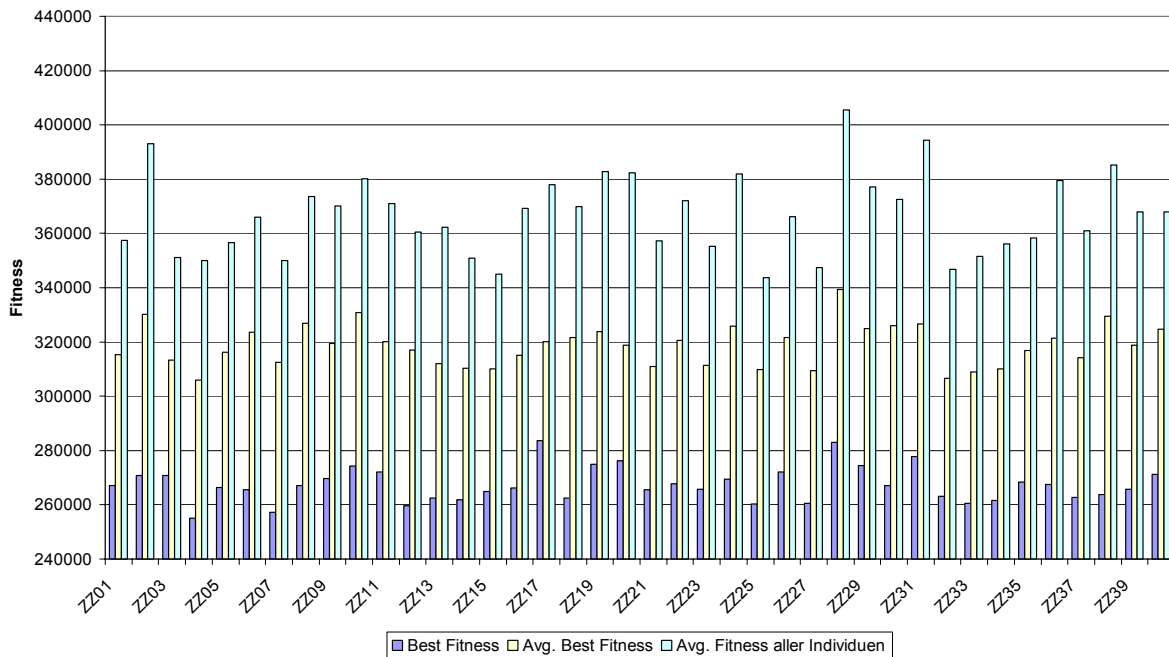


Abb. 5.3 Einfluss der Startzufallszahl

Dargestellt sind für jeden der 40 bis auf die Startzufallszahl identischen Optimierungsläufe die beste erreichte Fitness, die durchschnittliche beste Fitness und die durchschnittliche Fitness aller Individuen.

In Tab. 5.2 sind die Ergebnisse der 40 Läufe zusammengestellt.

	Durchschnitt	Std.-Abw.	Minimum	Maximum
Beste Fitness	267 432	6 382	255 166	283 753
erreicht in Generation	78	2,20	71	80
Mittlere Beste Fitness	318 531	7 657	306 012	339 427
Mittlere Fitness aller Individuen	366 716	14 760	343 791	405 480
Generationen	80	0	80	80
Bewertete Individuen	6 440	0	6 440	6 440
Rechenzeit [s]	367	1,25	365	370
Rechenzeit [min]	6,12	0,0208	6,08	6,17
RZ Initialisierung [s]	169	0,827	168	171
Rechenzeit/Gen. [s]	2,47	0,0107	2,45	2,49
Rechenzeit/Ind. [s]	0,0307	0,00013	0,0305	0,0310

Tab. 5.2 Ergebnisse der Konsistenzuntersuchung

Die Standardabweichung der Besten Fitness liegt mit 6 382 bei 2,4% vom Mittelwert. Das beste Ergebnis (255 166) ist um 4,6% besser als der Mittelwert, das schlechteste Ergebnis (283 753) ist um 6,1% schlechter als der Mittelwert. Angesichts dieser relativ geringen Schwankungsbreite können die Ergebnisse als ausreichend *konsistent* angesehen werden.

Die Rechenzeit für einen Optimierungslauf liegt mit durchschnittlich 6,12 min etwas über der Zielgröße von 5 min. Für die Realanwendung werden deshalb gute Individuen in die Startpopulation eingelesen, die es ermöglichen, mit weniger Bewertungen eine gute Lösung zu erreichen und damit unter den 5 min zu bleiben.

Neben der Bewertung der Konsistenz des Algorithmus lässt sich aus den 40 Läufen die Anzahl benötigter identischer Optimierungsläufe mit unterschiedlichen Startzufallszahlen abschätzen, um mit hoher Wahrscheinlichkeit sagen zu können, dass eine Algorithmusvariante bzw. ein Parametersatz besser oder schlechter ist als ein anderer.

Die Anzahl n ergibt sich dabei aus folgender Betrachtung:

$$n \geq \frac{t(\alpha, n-1)^2 \cdot s^2}{e_a^2}$$

mit:

$t(\alpha, n-1)$ Wert aus der Student-Verteilung

s^2 Streuung der Stichprobe

e_a^2 Gewünschte absolute Genauigkeit (Konfidenzintervall)

Da n sowohl links als auch rechts der Ungleichung vorkommt, muss iterativ vorgegangen werden. Setzt man als gewünschte Genauigkeit 3% vom Mittelwert, also 8 023, an, dann erreicht man bereits mit 3 Läufen ein Konfidenzniveau α zwischen 80% und 90%. Ist man mit einer Genauigkeit von 4% vom Mittelwert (10 697) zufrieden, reichen 2 Läufe aus, um ein Konfidenzniveau zwischen 80% und 90% zu erreichen.

5.6 Auswahl einer Grundvariante

Die EO-Bibliothek bietet vier Grundvarianten Evolutionärer Algorithmen an. Diese werden von den Entwicklern der Bibliothek wie folgt bezeichnet:

- EO-Real
- EO-ES-Simple
- EO-ES-Stdev
- EO-ES-Full

EO-Real ist ein Standard-GA ohne adaptive Komponente. Dafür sind drei unterschiedliche, beliebig kombinierbare Mutationsvarianten integriert. Bei den anderen drei Varianten ist die Mutation adaptiv. Bei der Variante EO-ES-Simple wird pro Individuum ein adaptiver Mutationsschrittweitenparameter ergänzt, der für alle Gene die Standardabweichung für die normalverteilte Mutationsschrittweitenwahrscheinlichkeit festlegt. Der Parameter unterliegt ebenfalls der Evolution, wodurch sich vorteilhafte Mutationsschrittweiten im Lauf der Optimierung durchsetzen sollen. Die Variante EO-ES-Stdev geht einen Schritt weiter und verwendet für jedes einzelne Gen einen eigenen adaptiven Mutationsschrittweitenparameter. Bei der komplexesten Variante, EO-ES-Full, hat jedes Individuum zusätzlich als weitere Parameter Korrelationskoeffizienten. Die theoretischen Hintergründe sind z.B. in HANSEN und OSTERMEIER [2001] beschrieben.

Bei allen Varianten wurden die chromosomweise Rekombination und ein Verjüngungsindikator ergänzt (vgl. Abschnitte 4.5.2 und 4.6).

Zur Auswahl einer geeigneten Grundvariante werden sechs Optimierungsläufe mit unterschiedlichen Parametereinstellungen durchgeführt. In den Testläufen (TL) mit den Nummern 1-4 wird jeweils eine Population von 100 Individuen und 100 Generationen verwendet, die Anzahl von Nachkommen ist auf 110% eingestellt. Variiert werden die Einstellungen für die Rekombinationsvarianten. Dadurch werden jeweils insgesamt 11 100 Individuen bewertet (die Startpopulation zählt als 0. Generation). Im Testlauf 5 (TL5) wird die Populationsgröße auf 36 reduziert und die Anzahl der Nachkommen auf 300% eingestellt, wodurch eine vergleichbare Gesamtzahl (10 836) an Individuen entsteht. Im TL6 wird die Anzahl der Nachkommen bei 300% belassen und die Populationsgröße wieder auf 100 gesetzt, wodurch etwa dreimal so viele Individuen (30 370) generiert werden.

Die Ergebnisse der *besten Lösungen* sind in Abb. 5.4 dargestellt.

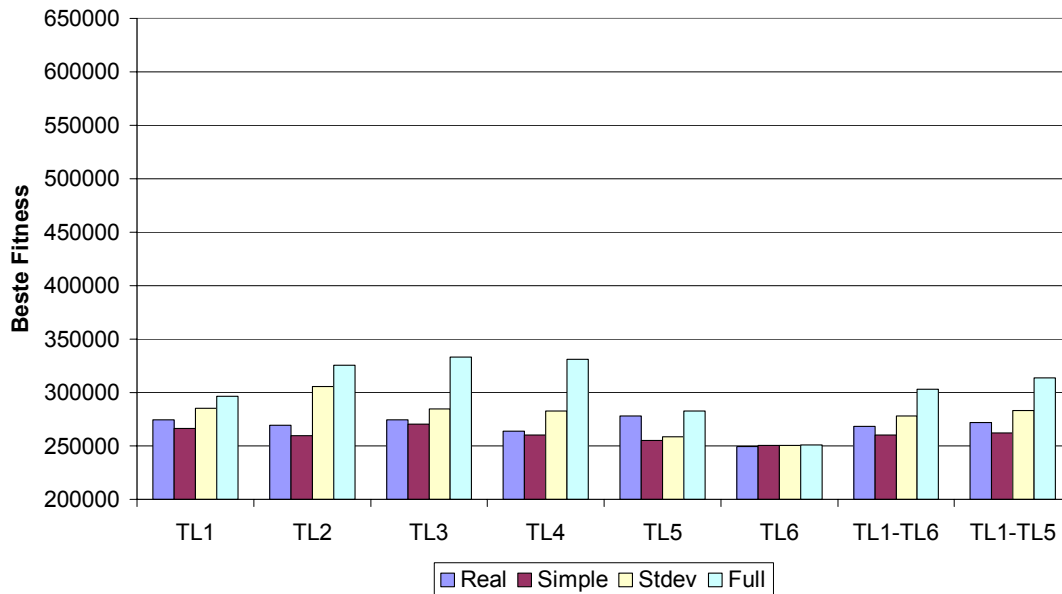


Abb. 5.4 Vergleich der besten erreichten Fitness

Abgesehen vom Testlauf 6, bei dem dreimal so viele Individuen erzeugt und bewertet werden, erzielt die Variante EO-ES-Simple in allen Läufen das beste Ergebnis.

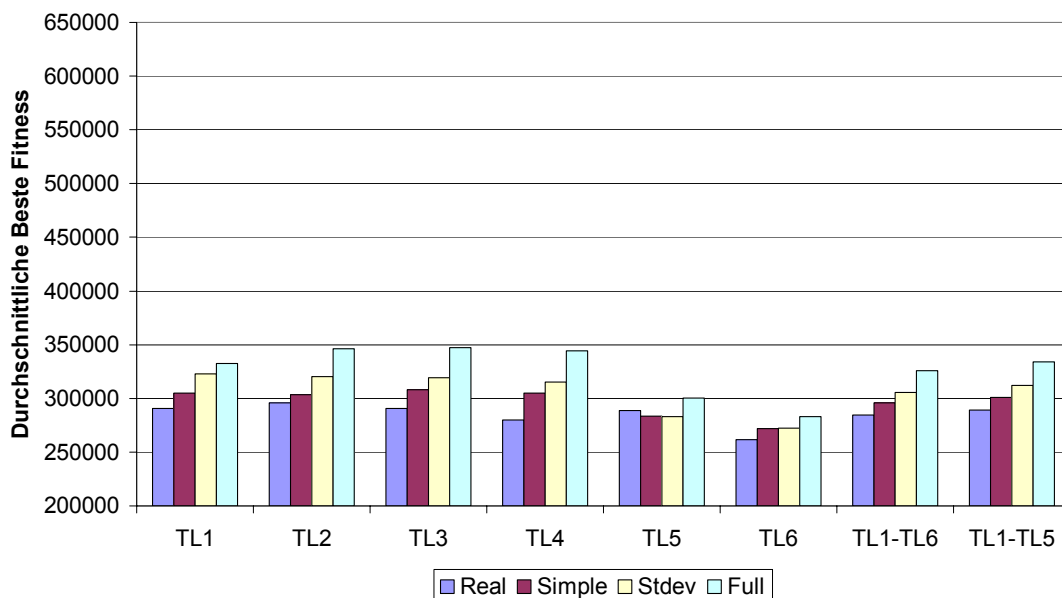


Abb. 5.5 Vergleich der durchschnittlichen Besten Fitness

Der *Durchschnitt der besten Fitnesswerte* ist ein Indikator für die Konvergenzgeschwindigkeit. Abb. 5.5 zeigt, dass die Konvergenzgeschwindigkeit mit der Komplexität des Algorithmus abnimmt.

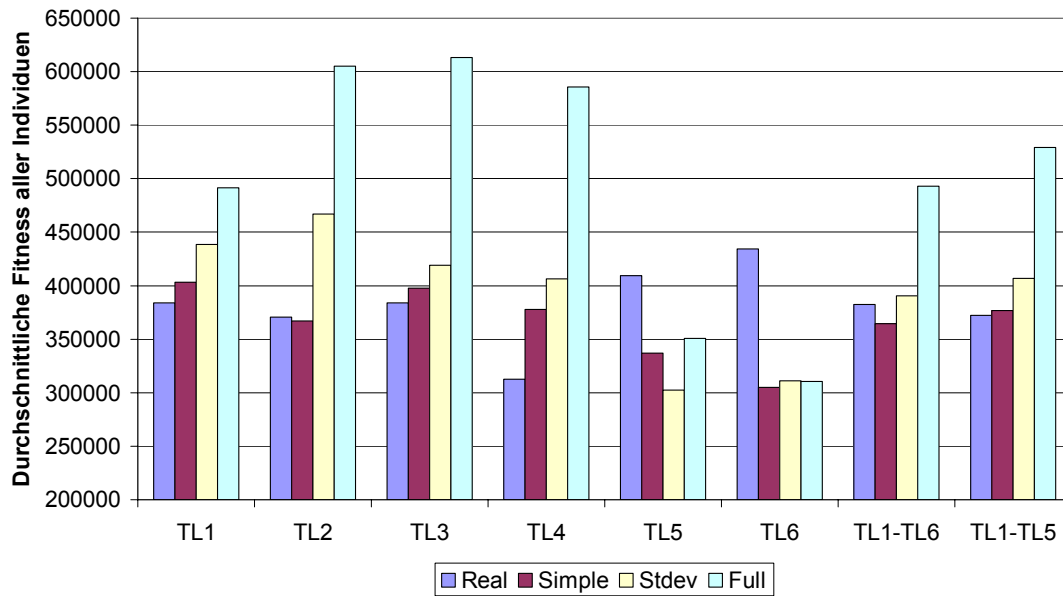


Abb. 5.6 Vergleich der durchschnittlichen Fitness aller bewerteten Individuen

Der Vergleich der *durchschnittlichen Fitness aller bewerteten Individuen* gibt Aufschluss über die Konvergenzgeschwindigkeit und die Anzahl der evaluierten Individuen. Im Durchschnitt der betrachteten Testläufe fällt der EO-ES-Full deutlich von den anderen drei Varianten ab (Abb. 5.6).

Die benötigte Rechenzeit eines Optimierungslaufs ist in Abb. 5.7 dargestellt.

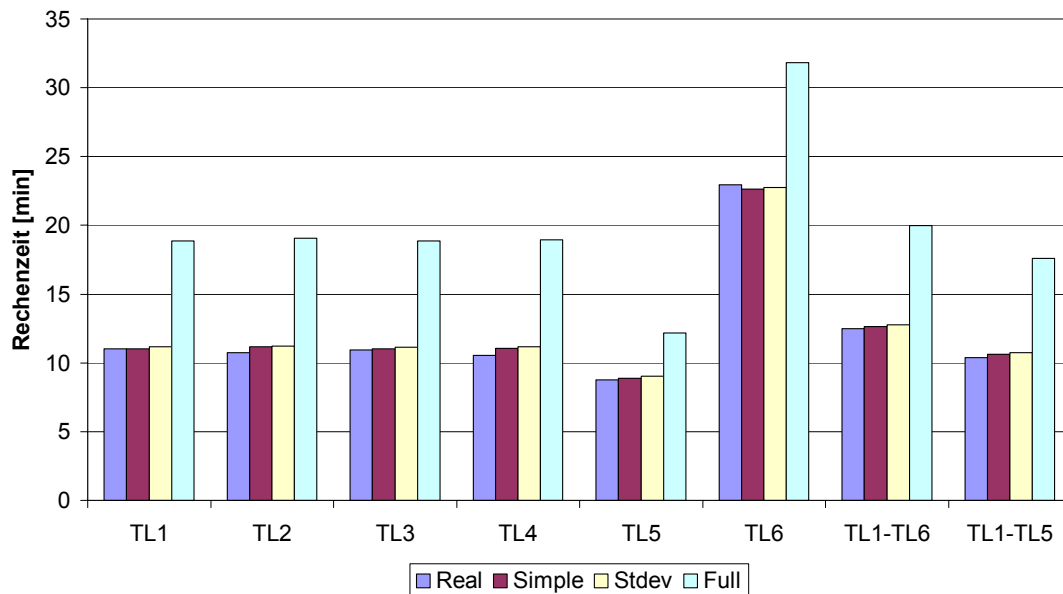


Abb. 5.7 Gesamte Rechenzeit eines Optimierungslaufs [min]

Der deutlich komplexere EO-ES-Full benötigt weitaus mehr Rechenzeit als die anderen drei Varianten. Die Tests wurden auf einem Rechner mit Intel® Xeon® E5320-Quadcore-Prozessoren (4 CPU-Kerne) mit einer Taktfrequenz von 1,86 GHz und einem 4 GB großen Arbeitsspeicher durchgeführt.

Abb. 5.8 zeigt die Rechenzeit, die pro Individuum benötigt wird.

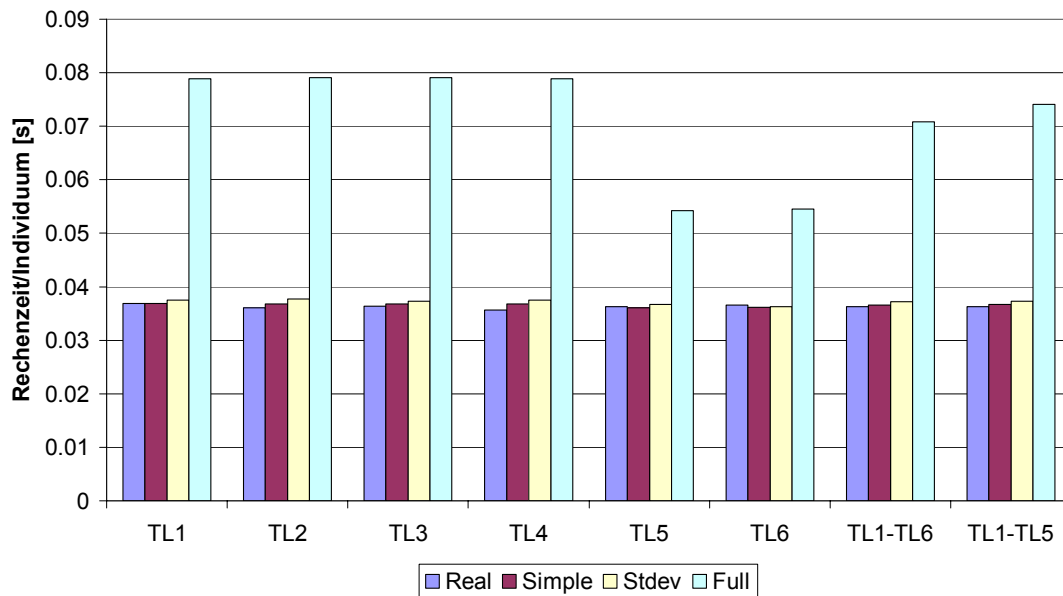


Abb. 5.8 Rechenzeit pro Individuum [s]

Wie erwartet nimmt die Rechenzeit pro Individuum mit der Komplexität des Algorithmus zu. EO-ES-Full benötigt deutlich mehr Rechenzeit als die anderen drei Varianten.

Zusammenfassend lässt sich feststellen, dass die beiden komplexeren Varianten EO-ES-Stdev und EO-ES-Full offensichtlich deutlich schlechter für dieses Optimierungsproblem geeignet sind als EO-Real und EO-ES-Simple. Abgesehen vom rechenzeitintensiven Testlauf 6, bei dem die beste erreichte Fitness bei allen vier EO-Varianten fast gleich ist, hat *EO-ES-Simple* in allen Testläufen das beste Ergebnis erzielt. Diese Variante wird deshalb für alle weiteren Untersuchungen verwendet.

5.7 Sensitivitätsanalyse

Im Rahmen der Sensitivitätsanalyse wird die Robustheit des Algorithmus zum einen bezüglich der Parametereinstellungen untersucht (Parametersensitivität), zum anderen wird ermittelt, inwieweit Robustheit und Effizienz von der Größe des Netzes und vom Nachfrageszenario abhängen.

Folgende Parameter werden detailliert untersucht:

- Rekombinationsvarianten
- Rekombinations- und Mutationswahrscheinlichkeit
- Anzahl der Nachkommen
- Startmutationsschrittweite

Um die Vergleichbarkeit zu gewährleisten wird keine Verjüngung zugelassen. Gemäß der Überlegungen in Abschnitt 5.5 werden pro Parametereinstellung zwei Optimierungsläufe mit zwei unterschiedlichen Zufallszahlen durchgeführt. Dabei werden jeweils ca. 6 400 Individuen bewertet.

Wechselwirkungen zwischen den Parametern können durch die aus Aufwandsgründen sequentielle Vorgehensweise nur eingeschränkt untersucht werden. Die Einstellung der jeweils nicht untersuchten Parameter wird aus Erfahrung und Expertenwissen vorgenommen.

5.7.1 Rekombinationsvarianten

In der implementierten Form von GALOP stehen drei Rekombinationsvarianten zur Verfügung, deren jeweiliger Anteil eingestellt werden kann:

- Intermediäre Rekombination (segment crossover)
- Diskrete Rekombination (uniform crossover)
- Chromosomweise Rekombination

Der Vergleich der drei Varianten ohne Mutation ergibt, dass die diskrete Rekombination und die chromosomweise Rekombination für sich genommen deutlich bessere Ergebnisse liefern als die intermediäre Rekombination. Die diskrete Rekombination schneidet dabei für sich genommen etwas besser ab. Offensichtlich kann sie die fehlende Mutation am besten kompensieren.

Der Vergleich der drei Varianten mit Mutation wird, wie auch alle weiteren Tests im Rahmen der Sensitivitätsanalyse, für folgende drei Szenarien durchgeführt:

- Nordnetz-Morgenspitze
- Nordnetz-Abendspitze
- Südnetz-Morgenspitze

Die Ergebnisse des Tests verschiedener Kombinationen der Anteile der drei Rekombinationsvarianten (mit Mutation) für die drei Szenarien sind in den Abb. 5.9 dargestellt.

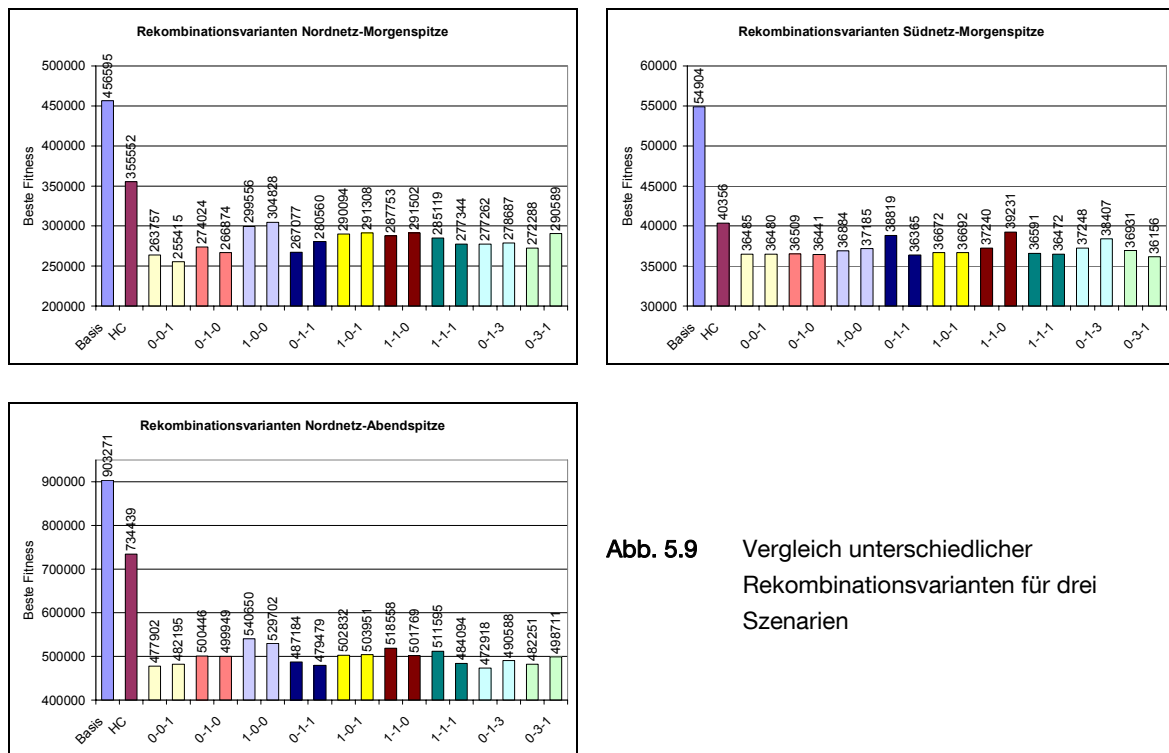


Abb. 5.9 Vergleich unterschiedlicher Rekombinationsvarianten für drei Szenarien

Die Einstellung 1-0-0 bedeutet, dass ausschließlich die intermediäre Rekombination zum Einsatz kommt. Entsprechend wird bei der Einstellung 0-3-1 zu 75% diskret und zu 25% chromosomweise rekombiniert. Für die Szenarien Nordnetz-Morgenspitze und Nordnetz-Abendspitze liefert die Einstellung 0-0-1, also die reine chromosomweise Rekombination, das beste Ergebnis der untersuchten Einstellungen. Am schlechtesten schneidet jeweils die Einstellung 1-0-0, also die reine intermediäre Rekombination, ab, gefolgt von den Varianten, bei denen zu 50% intermediär rekombiniert wird (1-0-1 und 1-1-0).

Der Mittelwert der besten Fitness aller Läufe liegt beim Szenario Nordnetz-Morgenspitze bei 280 780, die Standardabweichung (12 988) entspricht 4,6%, das beste und das schlechteste Ergebnis liegen jeweils etwa 9% vom Mittelwert entfernt. Beim Szenario Nordnetz-Abendspitze liegt das mittlere Ergebnis bei 498 043, die Standardabweichung (18 397) entspricht 3,7%, das beste Ergebnis ist um 5% besser, das schlechteste Ergebnis um 8,6% schlechter.

Weniger deutlich sind die Unterschiede der Ergebnisse der Rekombinationsvarianten beim Szenario Südnetz-Morgenspitze. Hier scheint der Algorithmus dahingehend sehr robust zu sein. Das zeigt sich auch an der Standardabweichung von nur 2,4%, was der Standardabweichung der Zufallssuche entspricht. Das Minimum ist nur um 2,4% besser als der Mittelwert, das Maximum um 5,9% schlechter.

Die Tatsache, dass der Algorithmus bezüglich der Einstellung des Verhältnisses der Rekombinationsvarianten beim Südnetz sehr robust ist, beim deutlich größeren Nordnetz jedoch

etwas sensitiver reagiert, deutet darauf hin, dass die Netzgröße dabei eine Rolle spielt. Offensichtlich ist es bei größeren Netzen von Vorteil, wenn alle oder ein Teil der Rekombinationen chromosomweise ablaufen, d.h. wenn die Signalpläne der einzelnen LSA nur über die Mutation verändert werden können.

Bei allen untersuchten Optimierungsläufen war das Ergebnis immer besser als das des Hill-Climbing Algorithmus.

5.7.2 Rekombinations- und Mutationswahrscheinlichkeit

Über die Rekombinationswahrscheinlichkeit (p_{Cross}) und die Mutationswahrscheinlichkeit (p_{Mut}) kann eingestellt werden, wie die neue Generation zusammengesetzt werden soll. Bei einer Einstellung von $p_{\text{Cross}} = 0,5$ und $p_{\text{Mut}} = 0,5$ wird beispielsweise jeweils etwa ein Drittel der neuen Individuen nur durch Rekombination, nur durch Mutation bzw. durch beides entstehen.

Um das Wechselspiel zwischen Rekombination und Mutation zu untersuchen, werden 16 verschiedene Kombinationen jeweils mit zwei verschiedenen Zufallszahlen durchgeführt. Beide Parameter werden über den Bereich von 0,3 bis 0,9 in Schritten von 0,2 variiert.

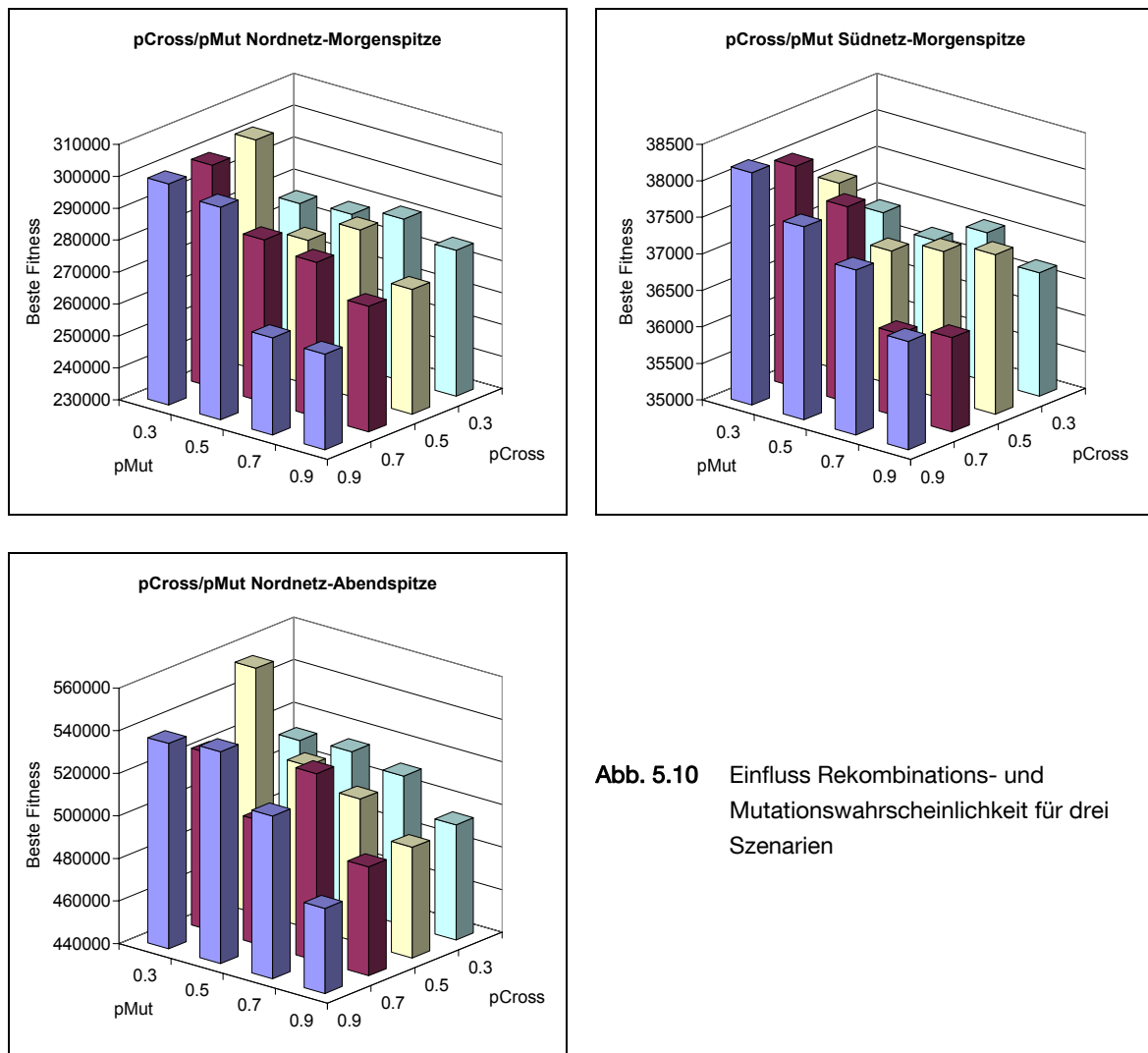


Abb. 5.10 Einfluss Rekombinations- und Mutationswahrscheinlichkeit für drei Szenarien

In den Abb. 5.10 sind für die drei Szenarien Nordnetz-Morgenspitze, Nordnetz-Abendspitze und Südnetz-Morgenspitze jeweils die Ergebnisse der 16 Kombinationen (Mittelwert der zwei Läufe) dargestellt.

Für das Szenario Nordnetz-Morgenspitze führt in jedem Fall eine hohe Mutationswahrscheinlichkeit zu besseren Ergebnissen als eine niedrige. Beste Kombination der getesteten 16 Varianten ist die Einstellung 0,9 für beide Parameter. Der Mittelwert der Ergebnisse (Beste Fitness) aller 32 Läufe liegt bei 280 220, die Standardabweichung bei 13 698, was 4,9% entspricht. Das schlechteste Ergebnis ist um 11,3% schlechter, das beste Ergebnis um 8,2% besser als der Mittelwert. Der Algorithmus reagiert für dieses Szenario offensichtlich relativ sensitiv auf die Einstellung der beiden Parameter.

Für das Szenario Nordnetz-Abendspitze sorgt ebenfalls die Mutationswahrscheinlichkeit von 0,9 für die besten Ergebnisse. Der Mittelwert der Ergebnisse aller 32 Läufe liegt für dieses Szenario bei 513 406, die Standardabweichung bei 23 393, was 4,6% entspricht. Das

schlechteste Ergebnis ist um 9,2% schlechter, das beste Ergebnis um 6,9% besser als der Mittelwert.

Auch beim Szenario Südnetz-Morgenspitze führen hohe Mutationswahrscheinlichkeiten in Kombination mit hohen Rekombinationswahrscheinlichkeiten zu guten Ergebnissen. Allerdings ist hier im Gegensatz zu den beiden Nordnetz-Szenarien die Einstellung von jeweils 0,7 am erfolgreichsten. Bei einer Standardabweichung von nur 1,7% vom Mittelwert der 32 besten Lösungen (37 102) und einem Unterschied von nur 3,3% des schlechtesten und 2,7% des besten Ergebnisses bezogen auf den Mittelwert, ist der Einfluss der beiden Parameter nicht sehr sensitiv, d.h. der Algorithmus ist diesbezüglich sehr robust.

Wie bereits bei den Rekombinationsvarianten scheint die Sensitivität des Algorithmus für das komplexere Netz größer zu sein, als für das kleinere Netz, wo er sich sehr robust verhält.

5.7.3 Anzahl der Nachkommen

Ein wichtiger Parameter für den Algorithmus ist die Anzahl der Nachkommen (nbOffspring), die pro Generation erzeugt werden. Sobald der Parameter auf mehr als 100% eingestellt wird, werden mehr Nachkommen als die Populationsgröße erzeugt, die dann anschließend per Abschneideselektion auf die Populationsgröße reduziert werden (Umweltselektion). Je höher der Parameter eingestellt wird, desto größer ist der Selektionsdruck.

Da aus Vergleichbarkeitsgründen die Gesamtzahl aller erzeugten und bewerteten Individuen gleich bleiben muss, wird die Populationsgröße entsprechend angepasst. Für jede Einstellung werden wieder zwei identische Läufe mit unterschiedlichen Startzufallszahlen durchgeführt. Die Optimierungsergebnisse in Abhängigkeit von der Anzahl der Nachkommen sind für die drei Szenarien in den Abb. 5.11 dargestellt.

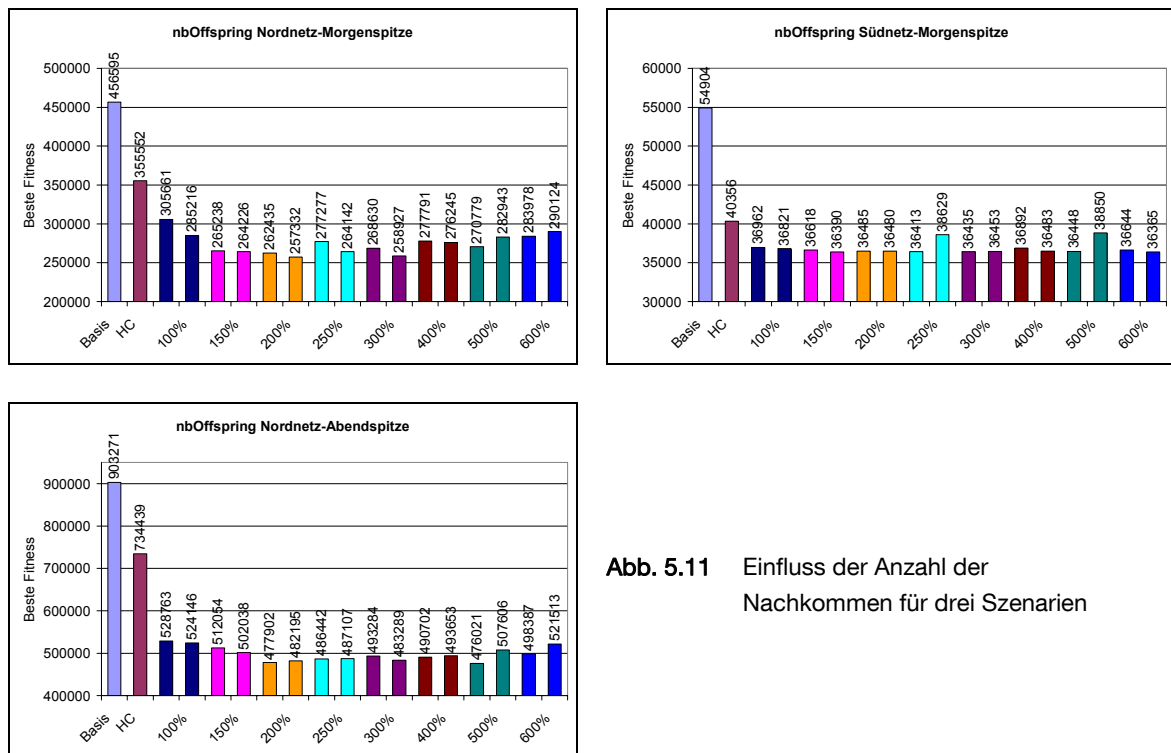


Abb. 5.11 Einfluss der Anzahl der Nachkommen für drei Szenarien

Wie die Ergebnisse der beiden Nordnetz-Szenarien zeigen, ist es offensichtlich von Vorteil, über die Anzahl der Nachkommen einen gewissen Selektionsdruck aufzubauen. Abgesehen von der Einstellung 100% ist der Ergebnisunterschied zwischen den anderen Einstellungen allerdings relativ gering. Für das Szenario Nordnetz-Morgenspitze beträgt die Standardabweichung der Optimierungsergebnisse 4,7% vom Mittelwert der 16 Läufe, der bei 274 434 liegt. Für das Szenario Nordnetz-Abendspitze beträgt die Standardabweichung nur 3,4% vom Mittelwert (497 819).

Beim Szenario Südnetz-Morgenspitze verhält sich der Algorithmus noch robuster. Hier kann kein Zusammenhang zwischen dem Optimierungsergebnis und der Einstellung des Parameters zur Festlegung der Anzahl der Nachkommen erkannt werden. Die Standardabweichung der Einzelergebnisse von nur 2,1% bezogen auf den Durchschnitt der Ergebnisse unterstreicht das. Sie liegt unter der Standardabweichung von 2,4%, die bei der Abschätzung des Einflusses der Startzufallszahl in Abschnitt 5.5 ermittelt wurde.

Die Auswirkung unterschiedlicher Einstellungen für die Anzahl der Nachkommen auf den Verlauf der Optimierung veranschaulicht Abb. 5.12. Exemplarisch für jeweils einen Lauf pro Parametereinstellung ist die Entwicklung der besten Fitness über die Generationen dargestellt.

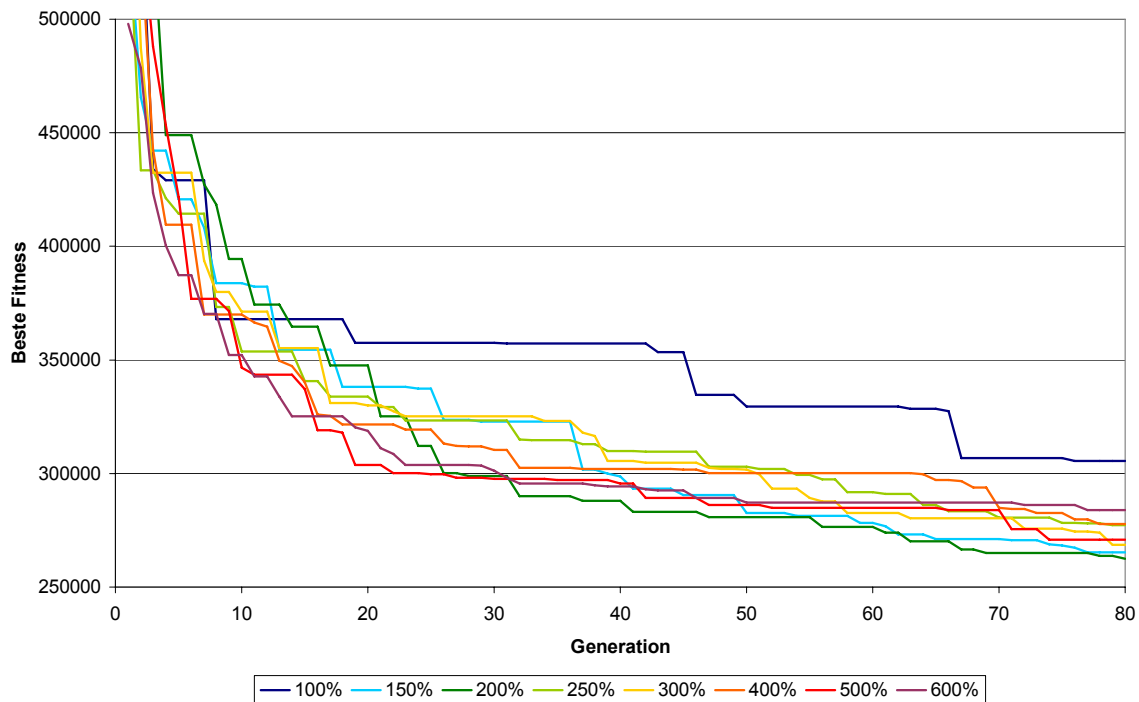


Abb. 5.12 Verlauf der besten Fitness für eine unterschiedliche Anzahl an Nachkommen

Erkennbar ist, dass eine hohe Anzahl von Nachkommen, also ein hoher Selektionsdruck, in den ersten 10 bis 20 Generationen der Optimierung schneller für relativ gute Individuen sorgt, dass aber im weiteren Verlauf keine wesentlichen Verbesserungen mehr stattfinden. Demgegenüber erreichen die Läufe mit moderatem Selektionsdruck auch in der zweiten Hälfte der Optimierung noch deutliche Verbesserungen. Der Optimierungslauf mit der Einstellung 100% (d.h. keine Umweltselektion) hat insbesondere zwischen der 20. und 40. Generation Schwierigkeiten, sich zu verbessern.

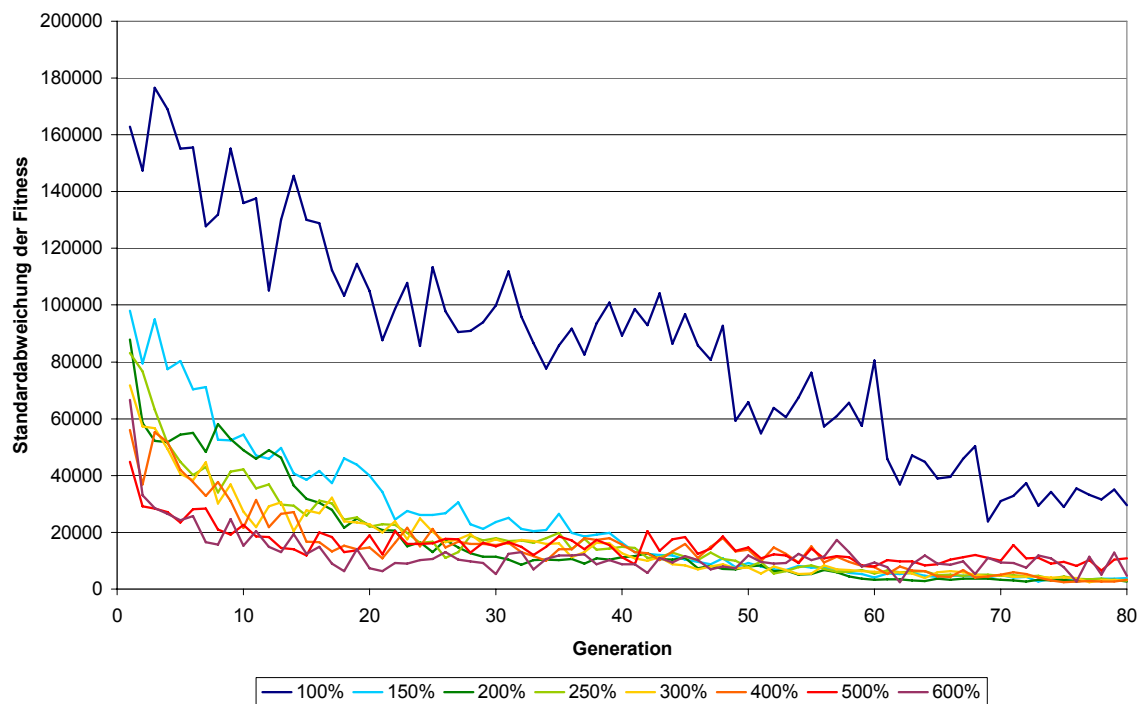


Abb. 5.13 Verlauf der Standardabweichung für eine unterschiedliche Anzahl an Nachkommen

Betrachtet man die Standardabweichung der Fitnesswerte innerhalb einer Generation (Abb. 5.13), werden die unterschiedlichen Verhaltensweisen des Algorithmus in Abhängigkeit von der Anzahl der Nachkommen deutlich. Je höher der Selektionsdruck ist, desto früher im Optimierungsverlauf nimmt die Standardabweichung der Fitnesswerte ab. D.h. der Algorithmus konvergiert wie erwartet bei höherem Selektionsdruck schneller.

5.7.4 Startmutationsschrittweite

Die Startmutationsschrittweite (σ_{init}) legt den Wert des adaptiven Parameters für die Individuen der Startpopulation fest. Der Parameter unterliegt selbst der Evolution, d.h. er wird ebenfalls mutiert und rekombiniert. Es werden für die drei Szenarien Nordnetz-Morgenspitze, Nordnetz-Abendspitze und Südnetz-Morgenspitze neun verschiedene Einstellungen für σ_{init} von 0,01 bis 0,50 getestet, jeweils wieder mit zwei unterschiedlichen Startzufallszahlen. Abb. 5.14 zeigen die Optimierungsergebnisse in Abhängigkeit von der Startmutationsschrittweite.

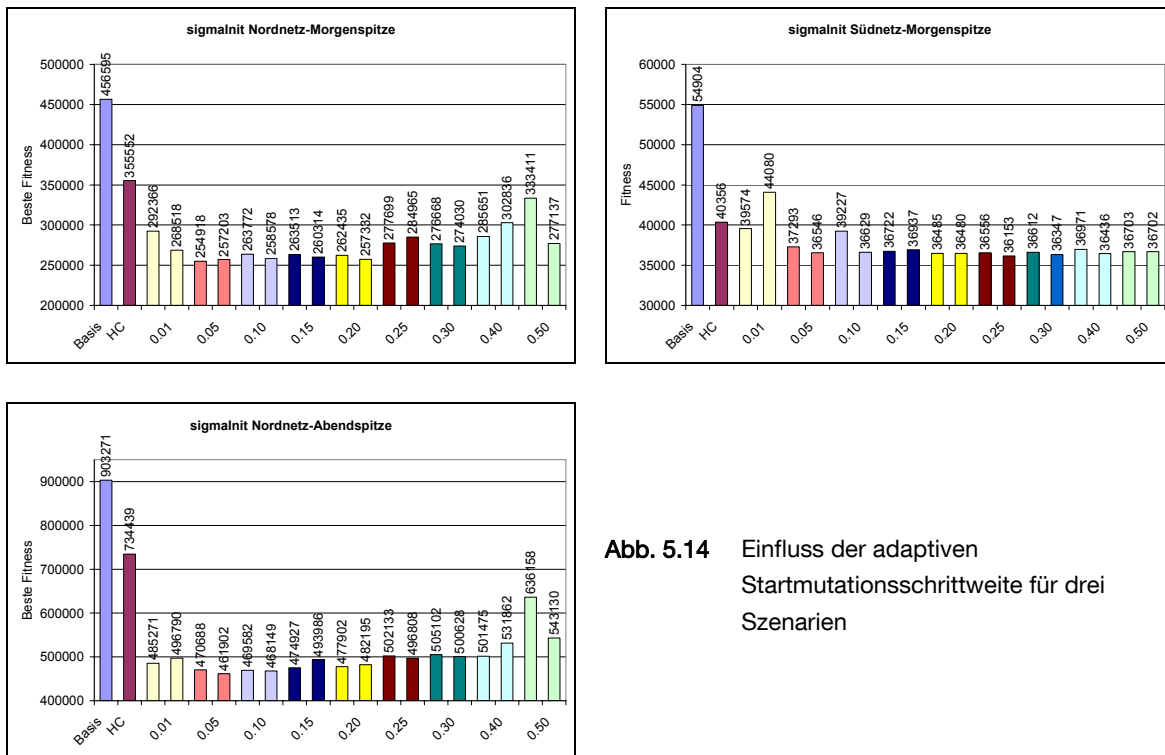


Abb. 5.14 Einfluss der adaptiven Startmutationsschrittweite für drei Szenarien

Für die beiden Nordnetz-Szenarien erhält man die besten Ergebnisse für eine Startmutationsschrittweite von 0,05 bis 0,20. Bei Anfangsschrittweiten größer als 0,25 werden die Ergebnisse schlechter. Beim Szenario Südnetz-Morgenspitze liefern alle Startmutationsschrittweiten von 0,05 bis 0,50 gute Ergebnisse, der Algorithmus verhält sich sehr robust.

Durch die Adaptivität soll sich die Mutationsschrittweite an den Verlauf der Optimierung anpassen. D.h. man geht davon aus, dass zu Beginn einer Optimierung größere Schrittweiten von Vorteil sind, um den Lösungsraum explorativ zu durchsuchen. Je weiter die Optimierung fortgeschritten ist, desto eher sollten kleinere Schrittweiten von Vorteil sein, um im Nahbereich guter Lösungen noch bessere zu finden. Dass dies offensichtlich tatsächlich der Fall ist, zeigen die Abb. 5.15.

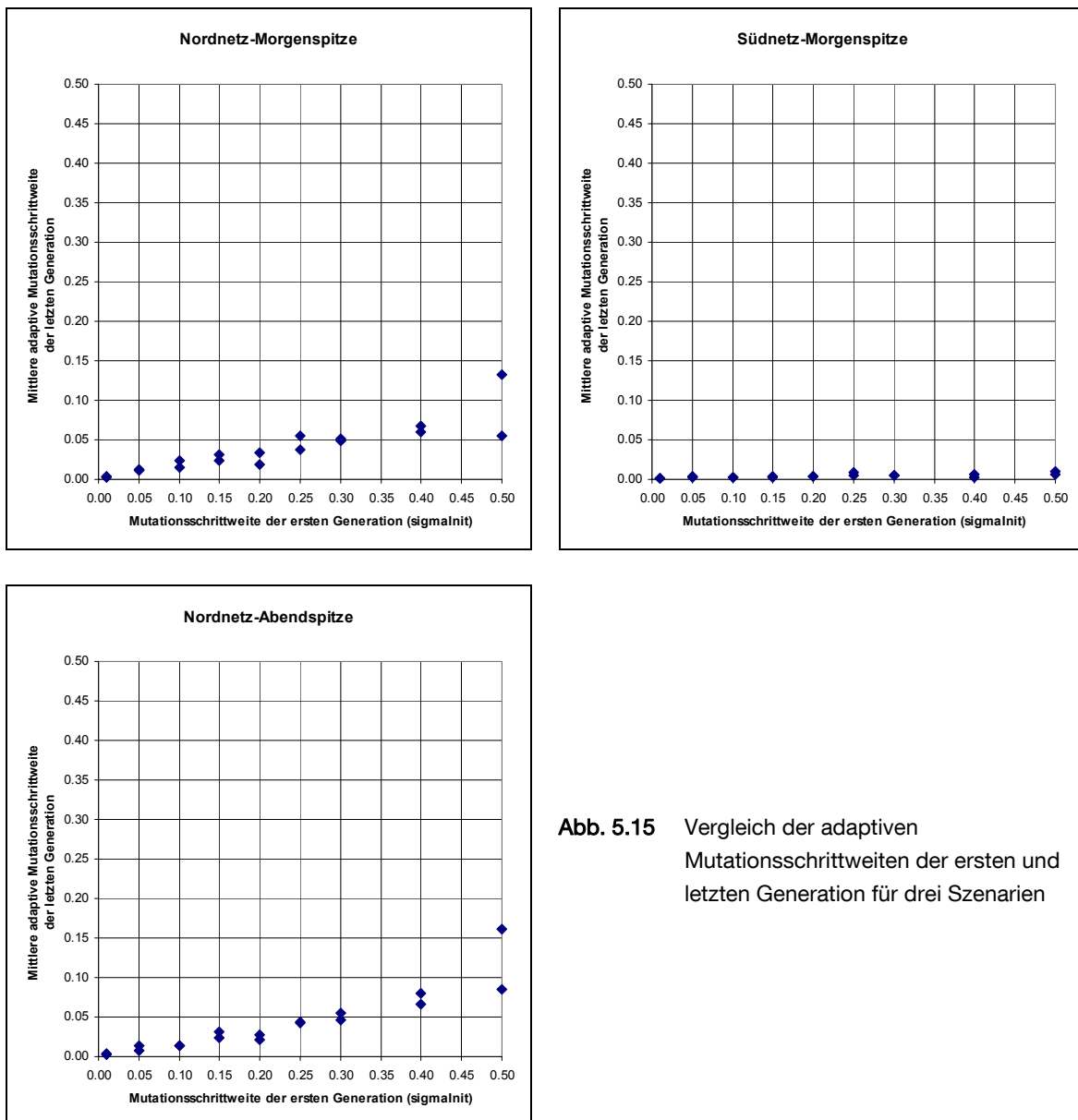


Abb. 5.15 Vergleich der adaptiven Mutationsschrittweiten der ersten und letzten Generation für drei Szenarien

Der Durchschnitt der adaptiven Mutationsschrittweiten der Individuen der letzten Generation ist in allen Fällen geringer als die Startmutationsschrittweite (sigmaInit). Am extremsten ist dieser Effekt beim Szenario Südnetz-Morgenspitze zu sehen. Fast unabhängig von der Startmutationsschrittweite befinden sich hier die Werte der adaptiven Mutationsschrittweite der letzten Generation unter 0,001. Das bedeutet, dass zum Ende der Optimierung fast keine Veränderung der Signalpläne mehr durch Mutation stattfindet.

5.7.5 Fazit der Sensitivitätsanalyse

Das Verhalten des Algorithmus scheint stärker vom Netz abzuhängen als von der Verkehrsnachfrage. Für das Szenario Südnetz-Morgenspitze verhält er sich sehr robust im Hinblick auf die Einstellungsparameter. Für die beiden Szenarien des größeren Nordnetzes ist der Algorithmus sensitiver bezüglich der Einstellungen, allerdings liefern auch ungünstige Einstellungen ein besseres Ergebnis als der Hill-Climbing Algorithmus.

Die Ergebnisse deuten darauf hin, dass insbesondere für große Netze (und damit entsprechend großen Lösungsräumen) eine Kalibrierung des Algorithmus durchgeführt werden sollte. Die Parametereinstellung, die das beste Ergebnis für das Szenario Nordnetz-Morgenspitze liefert, wird als kalibrierte Version für die weitere Bewertung verwendet.

5.8 Bewertung der Qualität und Effizienz

Die für das Szenario Nordnetz-Morgenspitze kalibrierte Version des Algorithmus GALOP wird auf alle zwölf Szenarien angewendet und validiert. Die Ergebnisse sind in den Abb. 5.16 jeweils zusammen mit der Güte der Basissteuerung (Basis), der durch den Hill-Climbing Algorithmus erreichten Güte (HC) und der quasi-optimalen Güte, die durch einen mehrstündigen Optimierungslauf mit einer Population von 200 Individuen und maximal 2 000 Generationen ermittelt wurde (200x2000), dargestellt.

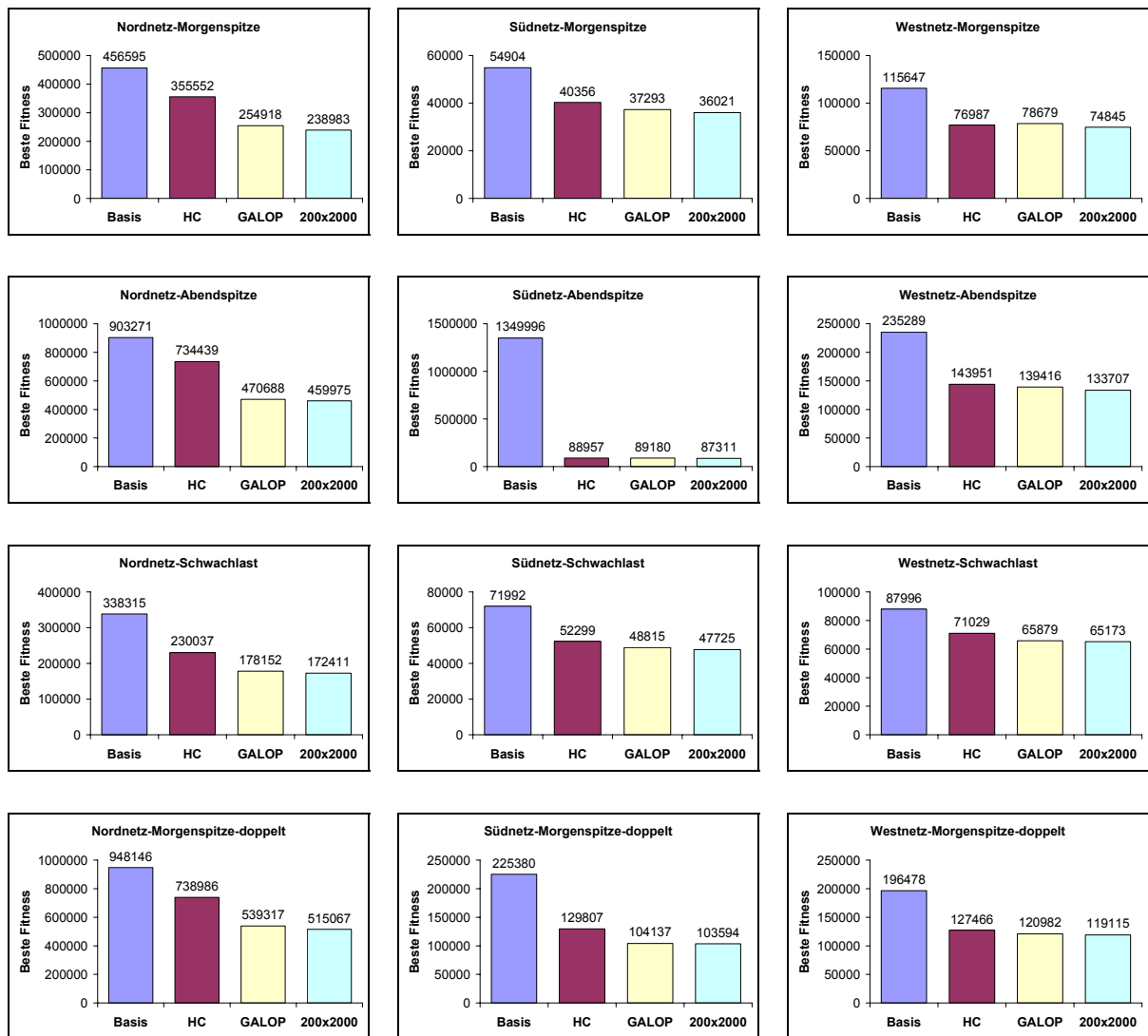


Abb. 5.16 Ergebnisse der Bewertung (Qualität der Bestlösung)

Die prozentualen Veränderungen gegenüber dem Referenzalgorithmus (HC) sind in Tab. 5.3 zusammengestellt.

Nachfrageszenario	Nordnetz	Südnetz	Westnetz
Morgenspitze (P2)	-28%	-8%	+2%
Abendspitze (P4)	-36%	0%	-3%
Schwachlast (P1)	-23%	-7%	-7%
Morgenspitze doppelt (P2-doppelt)	-27%	-20%	-5%

Tab. 5.3 Prozentuale Unterschiede der Optimierungsergebnisse zwischen GALOP und HC

Für das Nordnetz, das mit 27 LSA größte der drei untersuchten Netze, ist die Güte der durch GALOP gefundenen Lösungen im Vergleich zu den Lösungen des Hill-Climbing Algorithmus um 23% bis 36% besser. Für das Südnetz konnte GALOP um 0% bis 20% bessere Lösungen finden. Für das Westnetz, ein Streckenzug mit acht LSA, fand GALOP für das Nachfrageszenario Morgenspitze nur eine um 2% schlechtere Lösung, für die anderen Szenarien ist die Güte nur um 3% bis 7% besser. Offensichtlich liefert der Hill-Climbing Algorithmus auf einem relativ kleinen linienhaften Netz wie dem Westnetz genauso gute Ergebnisse wie GALOP. Je größer und komplexer das Netz wird, desto mehr kommen die Vorteile eines Evolutionären Algorithmus wie insbesondere die gleichzeitige Optimierung aller Steuerungsparameter zum Tragen. Die Struktur und Stärke der Verkehrsnachfrage scheint dabei keine Rolle für die Algorithmen zu spielen.

Der Evolutionäre Algorithmus ist in der Lage bereits mit 6 440 Individuen eine Qualität zu erreichen, die relativ nah an der quasi-optimalen Lösung ist. In den zwölf Szenarien beträgt der Abstand zur quasi-optimalen Lösung zwischen 1% und 7%. D.h. GALOP ist effizient genug, um selbst ohne das Einlesen guter Lösungen vorangegangener Optimierungsläufe innerhalb kurzer Zeit sehr gute Lösungen zu ermitteln.

5.9 Fazit der Bewertung

In den vorangegangenen Abschnitten wurde das neu entwickelte Optimierungsverfahren GALOP bewertet. Als Referenzverfahren diente der bisher in BALANCE enthaltene Hill-Climbing Algorithmus und eine gut geplante Basissteuerung. Die Güte wurde in allen Fällen durch das gleiche Verkehrsmodell berechnet, sodass eine Vergleichbarkeit gewährleistet war. Es konnte gezeigt werden, dass die durch GALOP erreichte *Qualität* der jeweiligen Bestlösung vor allem für das Nordnetz deutlich besser als die des Hill-Climbing Algorithmus ist.

Unterschiedliches Verhalten von GALOP und HC

Der wesentliche Vorteil des Evolutionären Algorithmus besteht in der gleichzeitigen Optimierung aller Steuerungsparameter. Im Gegensatz dazu optimiert der Hill-Climbing Algorithmus die Steuerungsparameter sequentiell, die entstandene Lösung hängt von der gewählten Reihenfolge ab. Begonnen wird bei einem Knotenpunkt, für den alle T-Zeiten im Rahmen des erlaubten Intervalls nacheinander optimiert werden. Anschließend folgt der nächste Knotenpunkt, bis alle Knotenpunkte des Netzes optimiert wurden. Bei dem in BALANCE implementierten Hill-Climbing Algorithmus wird diese Vorgehensweise aufbauend auf dem jeweils optimierten Parametersatz dreimal wiederholt.

Durch diese Vorgehensweise werden beispielsweise für das Nordnetz 2 200 bis 2 500 Lösungen erzeugt und bewertet. Die Zahl ist nicht fix, weil sie vom Optimierungsverlauf ab-

hängt. Sobald für einen Steuerungsparameter in die eine Richtung keine bessere Lösung mehr gefunden wird, wird die Suche für eine andere Richtung fortgesetzt. Das unterschiedliche Verhalten von Hill-Climbing Algorithmus und Evolutionärem Algorithmus ist in Abb. 5.17 dargestellt.

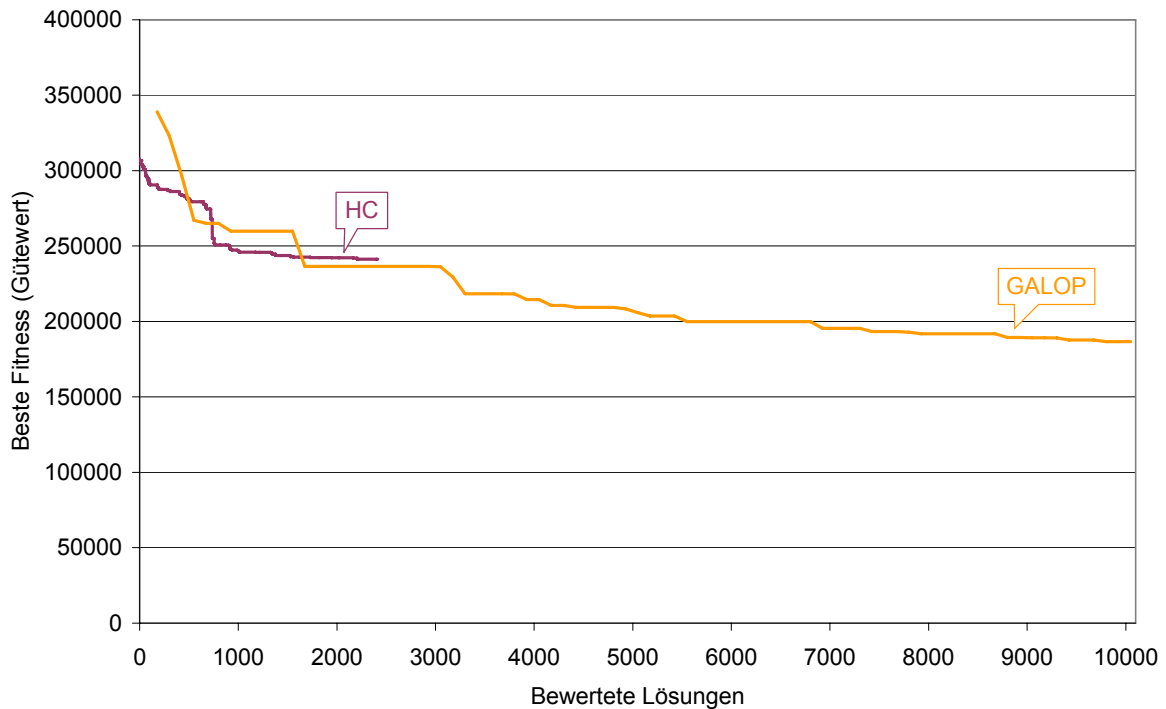


Abb. 5.17 Vergleich der Fitnessentwicklung von GALOP und HC

Der Hill-Climbing Algorithmus startet mit den Signalplänen der Basissteuerung als Startlösung, die bereits eine relativ gute Güte erreicht. Im Gegensatz dazu startet der Evolutionäre Algorithmus mit einer zufälligen Startpopulation. Das beste Individuum der ersten Generation (nach 175 bewerteten Individuen) erreicht diese Güte noch nicht. Während der Hill-Climbing Algorithmus allerdings nach 2 404 bewerteten Lösungen bei einer Lösungsgüte von 241 742 abbricht, weil er keine bessere Lösung mehr findet, ist der Evolutionäre Algorithmus nach 2 300 Individuen in der 18. Generation bereits bei einer Fitness von 236 556 angelangt. Nach 80 Generationen und 10 050 bewerteten Lösungen hat GALOP eine Fitness von 186 559 erreicht. Dieses Beispiel verdeutlicht nochmals, dass die *Effizienz* von GALOP für die Echtzeitanwendung absolut ausreichend ist.

Die *Konsistenz* der Lösungsqualitäten wurde in Abschnitt 5.5 untersucht und nachgewiesen. Dazu wurde eine für gut befundene Variante des Algorithmus 40-mal mit verschiedenen Startzufallszahlen ausgeführt. Die Standardabweichung der Besten Fitness lag bei 2,4% vom Mittelwert der 40 Läufe.

Die *Robustheit* bezüglich unterschiedlicher Parametereinstellungen ist vor allem bei den kleineren Netzen sehr hoch. Bei großen Netzen reagiert der Algorithmus sensitiver. Deshalb sollte insbesondere für komplexere Netze nicht auf eine Kalibrierung verzichtet werden, um möglichst gute Lösungen zu finden. Durch das Einlesen guter Lösungen aus früheren Optimierungsläufen kann der Algorithmus noch robuster gemacht werden, weil von Anfang an relativ gutes Genmaterial vorhanden ist.

Auf Grund der Tatsache, dass das Optimierungsverfahren ein Verkehrsmodell als Bewertungsmodell benötigt, ist die Frage, wie sich der Algorithmus unter Verwendung eines anderen Verkehrsmodells verhält, im Rahmen dieser Arbeit nicht zu klären. Für ähnlich strukturierte Modelle dürfte die Übertragbarkeit allerdings gegeben sein.

6 Anwendung des Verfahrens

In diesem Kapitel wird die erste Real-Anwendung des entwickelten Optimierungsverfahrens beschrieben. GALOP wurde im Rahmen des Forschungsprojekts TRAVOLUTION durch die GEVAS software GmbH in BALANCE integriert und in Feldversuchen in der Stadt Ingolstadt getestet. Dabei wurden drei Steuerungsvarianten miteinander verglichen:

- Basis
- BALANCE mit Hill-Climbing Algorithmus (B-HC)
- BALANCE mit GALOP (B-GALOP)

Unter *Basis* wird der Zustand vor dem Einsatz der adaptiven Netzsteuerung verstanden. In allen drei Fällen besteht eine lokale verkehrabhängige Steuerung, in die eine ÖPNV-Priorisierung integriert ist. Außerdem sind einige reine Fußgängerschutzanlagen im Testgebiet, die nur auf Anforderung per Taster reagieren.

Gemessen wurde an drei mal zwei Tagen, Tab. 6.1 gibt eine Übersicht über die Messtage:

Abk.	Datum	Szenario
F11	Mi. 28.06.2006	Basis
F12	Do. 29.06.2006	Basis
F21	Mi. 28.05.2008	B-HC
F22	Do. 29.05.2008	B-HC
F31	Mi. 18.06.2008	B-GALOP
F32	Do. 19.06.2008	B-GALOP

Tab. 6.1 Übersicht Messtage

In den folgenden Abschnitten wird zunächst das Untersuchungsgebiet dargestellt (6.1), anschließend werden die Verkehrsstärkeganglinien der einzelnen Messtage verglichen (Abschnitt 6.2). In Abschnitt 6.3 wird die verwendete Messmethodik erläutert und in Abschnitt 6.4 finden sich die Ergebnisse der Feldversuche.

6.1 Untersuchungsgebiet

Das in Abb. 6.1 dargestellte Ingolstädter Testgebiet umfasst insgesamt 46 LSA. In unterschiedlichen Schraffierungen sind die drei Teilnetze markiert, deren LSA-Steuerung getrennt optimiert wird.

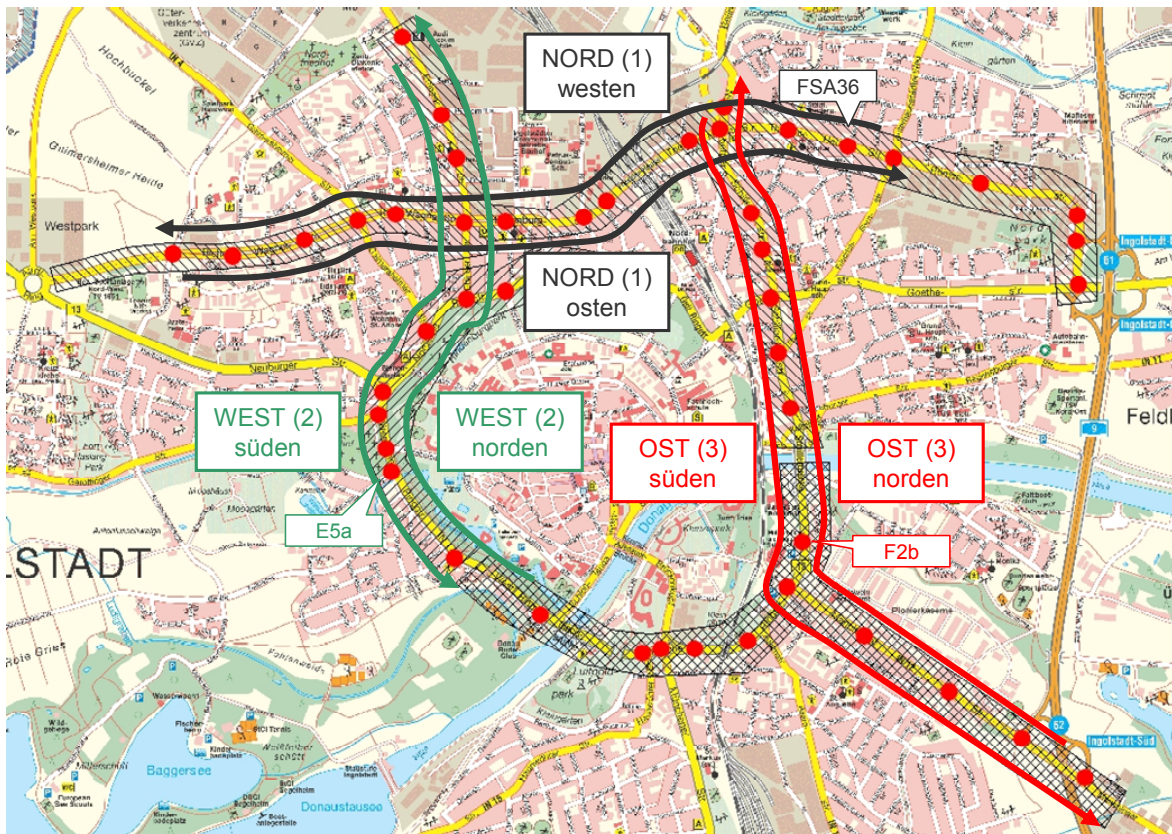


Abb. 6.1 Testgebiet Ingolstadt mit gemessenen Routen

Für die Feldversuche wurden sechs Routen festgelegt (vgl. Abb. 6.1):

- Route NORD (1) Richtung Osten
- Route NORD (1) Richtung Westen
- Route WEST (2) Richtung Norden
- Route WEST (2) Richtung Süden
- Route OST (3) Richtung Norden
- Route OST (3) Richtung Süden

Die Routen WEST und OST erstrecken sich über unterschiedliche Teilnetze hinweg. Ziel war es, eine möglichst nutzerorientierte Aussage über das gesamte Netz treffen zu können. Es ist zu betonen, dass man durch die Messungen auf den genannten Routen keine Gesamtaussagen für alle Verkehrsteilnehmer im gesamten Netz treffen kann. Alle erzielten Ergebnisse beziehen sich ausschließlich auf die genannten Routen.

6.2 Verkehrsstärkeganglinien

Die Verkehrsnachfrage in Ingolstadt ist wesentlich durch den Verkehrserzeuger Audi-Werk, das im Nord-Westen der Stadt liegt, geprägt. Dadurch unterscheiden sich die Verkehrsstär-

keganglinien je nach Ort sehr stark. Die folgenden Abbildungen zeigen die Verkehrsstärkeganglinien dreier Anlagen, die in Abb. 6.1 gekennzeichnet sind. Die gemessenen Verkehrsstärken sind in Fahrzeugen pro 15 Minuten angegeben. Es handelt sich jeweils um aggregierte Detektorzählwerte eines Querschnitts (meist zwei Fahrstreifen). Alle Fahrzeuge die zwischen 6:00 und 6:15 Uhr detektiert wurden, sind bei 6:00 Uhr angetragen.

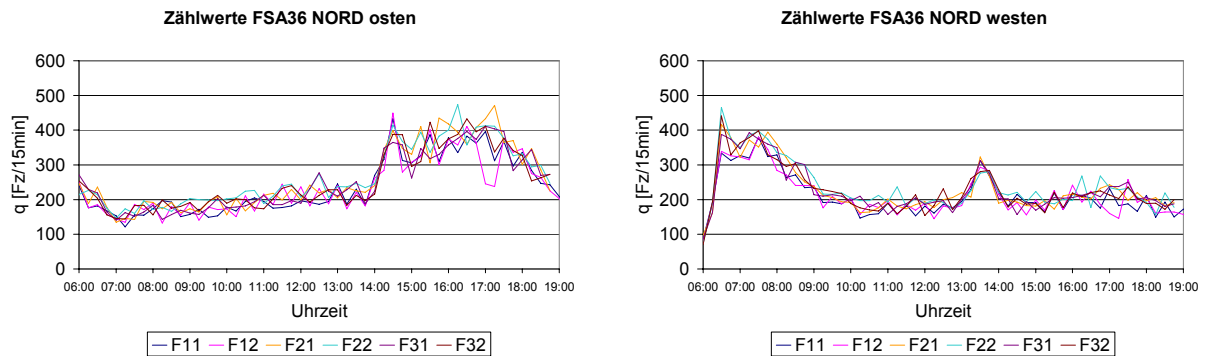


Abb. 6.2 Ganglinienvergleich Routen NORD (FSA36)

In den Abb. 6.2 sind exemplarisch für die Routen NORD die Ganglinien an der Fußgänger-
schutzanlage FSA36 dargestellt. In Richtung Westen, d.h. in Richtung Audi-Werk, gibt es
eine ausgeprägte Morgenspitze ab etwa 6:30 Uhr bis 9 Uhr, während die Abendspitze sehr
flach verläuft (Abb. 6.2, rechts). Klar erkennbar ist der Audi-Schichtwechsel am frühen
Nachmittag, der zu einer deutlichen und kurzfristigen Zunahme der Verkehrsnachfrage in
Richtung Audi-Werk zwischen 13 und 14 Uhr führt. Die Ganglinien der einzelnen Messtage
an der FSA36 in Richtung Westen sind ziemlich deckungsgleich. In der Morgenspitze war
etwas weniger Verkehr an den Feldtesttagen F11 und F12 (Basis), während insbesondere an
den Tagen F21, F22 und F32 eine ausgeprägte Spitze um 6:30 Uhr gemessen wurde.

In Richtung Osten zeigt sich an der FSA36 ein anderer Verlauf (Abb. 6.2, links). In den Mor-
genstunden werden hier keine besonders hohen Verkehrsstärken gemessen. Zwischen 14
und 15 Uhr ist der Audi-Schichtwechsel als kurze Verkehrsnachfragespitze vom Audi-Werk
weg erkennbar, ab 15 Uhr beginnt die ausgedehnte Abendspitze. Die Ganglinien der ein-
zelnen Messtage in Richtung Osten sind ebenfalls ziemlich deckungsgleich. In der Abend-
spitze wurden an den Feldtesttagen F21 und F22 (B-HC) leicht höhere Verkehrsstärken ge-
messen. Um 17 Uhr bei F12 (Basis) ist ein etwas stärkerer Rückgang der gemessenen Ver-
kehrsstärke zu verzeichnen, die Ursache dafür ist nicht bekannt.

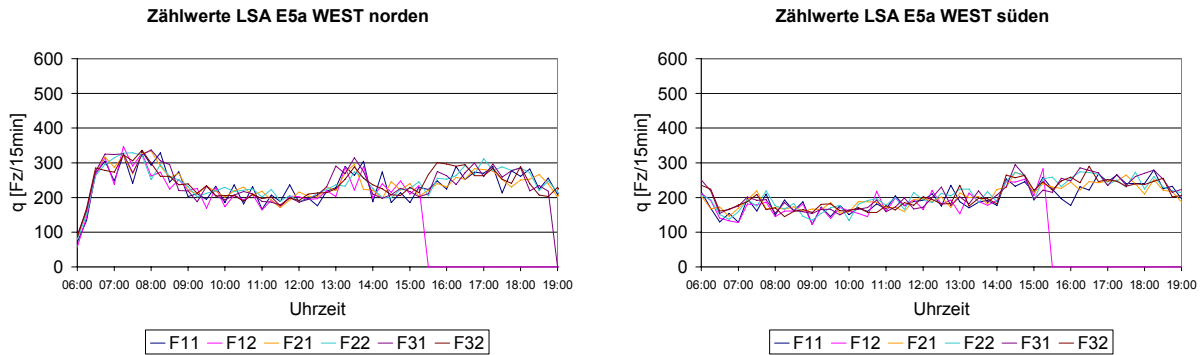


Abb. 6.3 Ganglinienvergleich Routen WEST (LSA E5a)

Die durch den Schichtwechsel bedingte höhere Nachfrage zwischen 13 und 14 Uhr ist auch auf der Route WEST in Richtung Norden erkennbar, ebenso wie die Morgenspitze zwischen ca. 6:30 Uhr und 9 Uhr und die Abendspitze ab etwa 15 Uhr (Abb. 6.3, links). Aus den Ganglinien der LSA E5a, einer typischen Anlage der Routen WEST, sind keine nennenswerten Unterschiede zwischen den einzelnen Messtagen erkennbar, abgesehen von Messwertausfällen bei F12 und F32.

Auf der Route WEST in Richtung Süden ist der Schichtwechsel an den nördlichen Anlagen D10 und D11 (Audi-Forum) ausgeprägt, der Rest der Ganglinien ist relativ flach mit leicht erhöhten Verkehrsstärken am Morgen und am Abend (Abb. 6.3, rechts). Die einzelnen Ganglinien verlaufen ebenfalls sehr ähnlich. Auch hier gab es einen Messwertausfall bei F12.

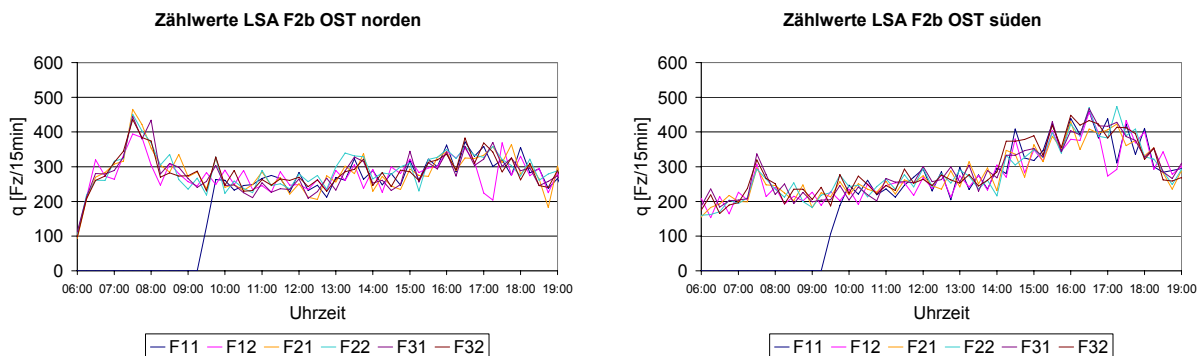


Abb. 6.4 Ganglinienvergleich Routen OST (LSA F2b)

Ein prinzipiell ähnliches Muster ist auf den Routen OST erkennbar. Der Audi-Schichtwechsel ist auch hier an der LSA F2b in Richtung Norden (Abb. 6.4, links) zwischen 13 und 14 Uhr (Zufluss) und in Richtung Süden (Abb. 6.4, rechts) zwischen 14 und 15 Uhr (Abfluss) in den Ganglinien erkennbar. Die Morgenspitze lässt sich auch an Hand dieser Diagramme in etwa auf 6:30 Uhr bis 9 Uhr festlegen, die Abendspitze beginnt gegen 15 Uhr und dauert

bis ca. 19 Uhr. Abgesehen von einem Messwertausfall bis 9:30 Uhr bei F11 und einem Einbruch bei F12 gegen 17 Uhr verlaufen die Ganglinien sehr ähnlich. In der Gegenrichtung sind bei F11 ebenfalls keine Messwerte bis 9:30 Uhr vorhanden, und gegen 17 Uhr geht die Verkehrsstärke bei F12 kurzfristig zurück. Ansonsten sind die Ganglinien vergleichbar.

Zusammenfassend kann festgestellt werden, dass trotz der fast zwei Jahre, die zwischen dem Feldtest für das Basisszenario und den Feldtests für die Szenarien BALANCE-HC und BALANCE-GALOP liegen, die tageszeitlichen Ganglinienverläufe der Verkehrsstärken gut übereinstimmen. Es kann deshalb davon ausgegangen werden, dass die Ergebnisse der Feldversuche nicht durch unterschiedliche Verkehrsnachfragen beeinflusst wurden.

6.3 Messmethodik

Zur Bewertung wurden zwei unterschiedliche Messmethoden verwendet:

- Befahrungen (Floating Car Data, FCD)
- Fahrzeugwiedererkennung

An allen sechs Messtagen wurden mit jeweils drei Fahrzeugen die sechs Routen befahren. Dabei wurden die Fahrten über GPS-Koordinaten aufgezeichnet und parallel weitere fahzeuggenerierte Daten, wie beispielsweise die vom Fahrzeug gemessene Geschwindigkeit, aus dem CAN-Bus ausgelesen.

Zusätzlich zu den Befahrungen wurden auf den Routen NORD zwei Infrarotkameras zur Fahrzeugwiedererkennung mittels automatischer Kennzeichenerfassung eingesetzt. Damit kann die Reisezeit zwischen zwei Punkten direkt gemessen werden. In Richtung Osten wurde jeweils am ersten Messtag (Mittwoch), in Richtung Westen jeweils am zweiten Messtag (Donnerstag) gemessen.

Auf Basis der Verkehrsstärkeganglinien (vgl. Abschnitt 6.2) werden neben der tagesdurchschnittlichen Bewertung (6:00-19:00) folgende Zeitbereiche separat betrachtet:

- Morgens: 06:30-09:00
- Tagsüber: 09:00-15:00
- Abends: 15:00-19:00

6.4 Ergebnisse

Die Ergebnisse der Feldversuche sind in drei Abschnitten dargestellt. In Abschnitt 6.4.1 werden zunächst die Ergebnisse der Reisezeitmessung durch Fahrzeugwiedererkennung auf den Routen NORD erläutert. Anschließend folgen die Ergebnisse aus den Befahrungen (Abschnitt 6.4.2). Die Ergebnisse aus den beiden unterschiedlichen Messmethoden werden

in Abschnitt 6.4.3 gegenübergestellt und daraus kombinierte Ergebnisse für die Verlustzeit gebildet.

6.4.1 Reisezeitmessung durch Fahrzeugwiedererkennung

Auf den Routen NORD wurden die Reisezeiten durch Fahrzeugwiedererkennung mittels automatischer Kennzeichenerfassung durch Infrarotkameras gemessen.

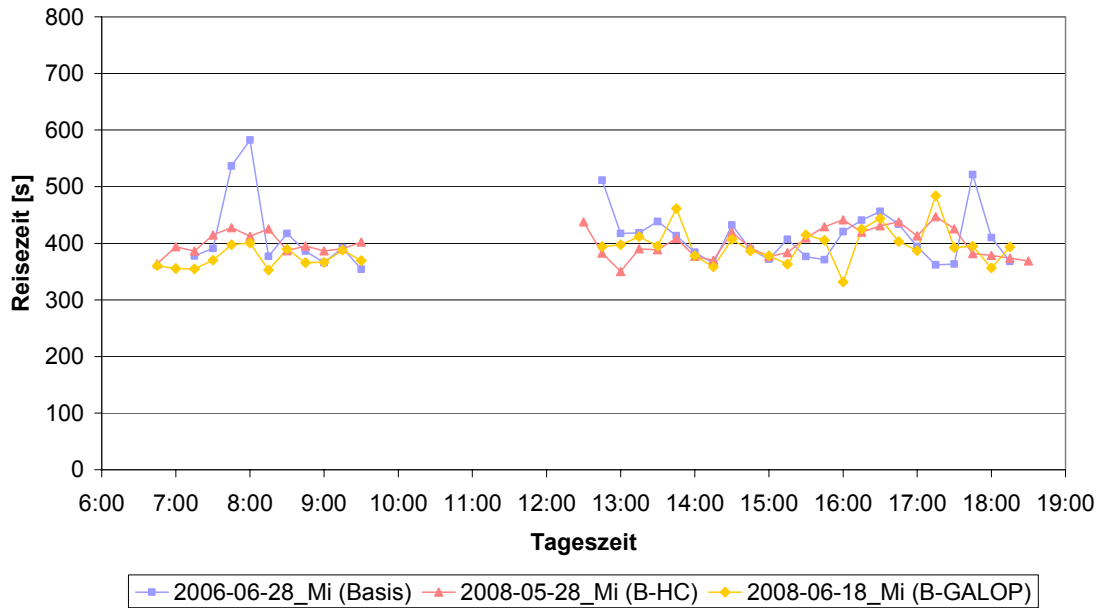


Abb. 6.5 Reisezeiten auf der Route NORD Richtung Osten

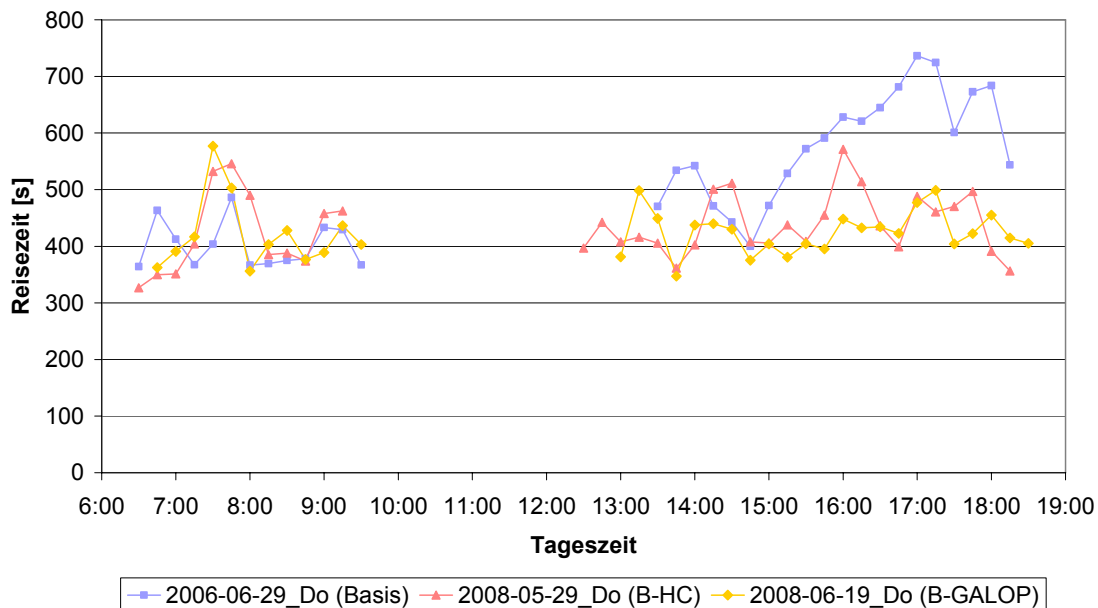


Abb. 6.6 Reisezeiten auf der Route NORD Richtung Westen

Der zeitliche Verlauf der Reisezeit in Richtung Osten ist in Abb. 6.5, in Richtung Westen in Abb. 6.6 dargestellt. Jeder Punkt in der Grafik repräsentiert den Mittelwert eines 15-Minuten-Intervalls. Zwischen 9:30 und 12:30 Uhr wurden keine Messungen durchgeführt.

In Richtung Osten steigt die Reisezeit am 28.06.2006 (Basis) ab 7:45 Uhr deutlich an, während sie zur gleichen Tageszeit am 28.05.2008 (B-HC) und am 18.06.2008 (B-GALOP) nur sehr leicht ansteigt. In Richtung Westen steigt demgegenüber die Reisezeit am 29.05.2008 (B-HC) und am 19.06.2008 (B-GALOP) etwas stärker an als am 29.06.2006 (Basis).

Nachmittags und abends sind die Reisezeiten in Richtung Osten auf einem ähnlichen Niveau. In Richtung Westen ist beim Basisfall ein sehr deutlicher Anstieg der Reisezeit ab 15:00 Uhr zu verzeichnen. Zudem schlägt sich der Audi-Schichtwechsel in höheren Reisezeiten zwischen 13:30 und 14:15 Uhr nieder. In diesen Zeiten bleibt die Reisezeit in den Fällen B-HC und B-GALOP niedrig, was dafür spricht, dass die adaptive Steuerung gut auf die sich ändernden Verkehrsstärken reagiert.

Die Reisezeit beträgt im Durchschnitt über die 15-Minuten-Intervalle auf der 3 813 m langen Route NORD in Richtung Osten 414 s im Basisfall, 401 s bei BALANCE mit HC und 390 s bei BALANCE mit GALOP. Das entspricht einer durchschnittlichen Reisegeschwindigkeit von 33 km/h, 34 km/h bzw. 35 km/h.

In Richtung Westen (Länge: 3 844 m) ist die durchschnittliche Reisezeit (Reisegeschwindigkeit) im Basisfall bei 503 s (27 km/h), bei BALANCE mit HC bei 434 s (32 km/h) und bei BALANCE mit GALOP bei 423 s (32 km/h).

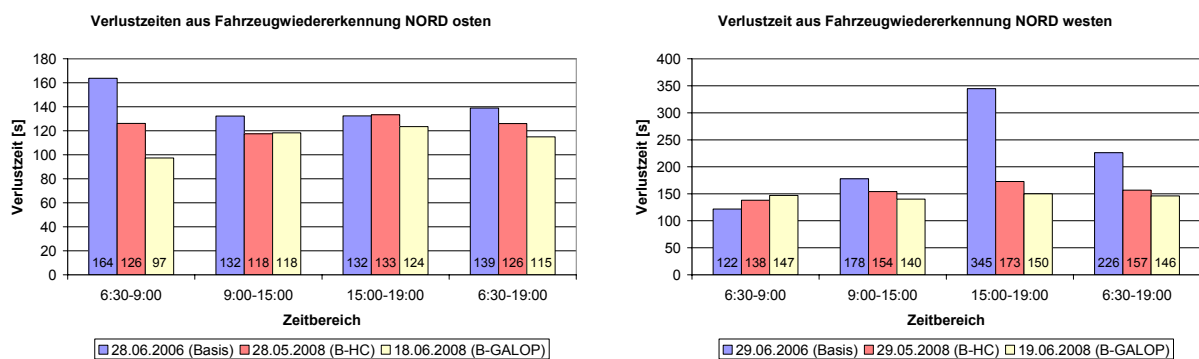


Abb. 6.7 Vergleich der Verlustzeiten aus der Fahrzeugwiedererkennung

In den Abb. 6.7 sind die gemittelten *Verlustzeiten*, also die Differenz aus gemessener und idealer Reisezeit, dargestellt. Die ideale Reisezeit ist die Zeit, die für die Route benötigt würde, wenn konstant die zulässige Höchstgeschwindigkeit (50 km/h) gefahren werden könnte. Die Verlustzeit auf einer Route setzt sich somit primär aus den LSA-bedingten Wartezeiten und sonstigen Zeitverlusten (z.B. durch ein-/ausparkende Fahrzeuge oder Ein-

/Abbieger) zusammen. Die gemittelte Verlustzeit eines Zeitbereichs ist der Durchschnitt der Mittelwerte aller darin enthaltenen 15-Minuten-Intervalle.

Für den Zeitbereich 6:30-9:00 konnte eine sehr starke Verringerung der Verlustzeit in Richtung Osten erreicht werden, allerdings auf Kosten der morgens stärker belasteten Route in Richtung Westen. Für den Zeitbereich 9:00-15:00 sind Verringerungen für beide Richtungen zu verzeichnen. Für den Abend (15:00-19:00) konnten die Verlustzeiten ebenfalls deutlich reduziert werden. In allen Fällen konnte nicht nur eine Verbesserung durch den Einsatz von BALANCE mit dem Hill-Climbing Algorithmus erreicht werden, sondern die Verlustzeiten wurden durch GALOP nochmals reduziert.

Über den gesamten Tag gesehen wurde die Verlustzeit durch BALANCE mit GALOP auf der Route NORD in Richtung Osten um 17% und in Richtung Westen sogar um 35% gegenüber dem Basisszenario reduziert. Der Unterschied zwischen BALANCE-HC und dem Basisszenario beträgt in Richtung Osten 9% und in Richtung Westen 31%. GALOP erreichte gegenüber dem Hill-Climbing Algorithmus eine Verbesserung um weitere 9% in Richtung Osten und um 7% in Richtung Westen.

6.4.2 Befahrungen

Die in Abb. 6.1 dargestellten sechs Routen wurden mit drei Fahrzeugen befahren. Fahrten, die gemäß den Fahrtenprotokollen nicht repräsentativ waren, weil beispielsweise ein Unfall auf der Route war, wurden nicht ausgewertet. Da die Anzahl der Fahrten großen Einfluss auf die Aussagekraft der Ergebnisse hat, wird in diesem Abschnitt ein tabellarischer Überblick gegeben. Anschließend werden die Ergebnisse (durchschnittliche Verlustzeiten und Zahl der Halte) sowohl getrennt nach Tageszeitbereichen, als auch für alle Fahrten, dargestellt.

Anzahl der Fahrten

Die Tab. 6.2 geben einen Überblick über die Anzahl der Fahrten, die pro Route ausgewertet wurden. Je nach Zeitbereich und Route stehen zwischen 5 und 30 Messfahrten zur Verfügung.

Route NORD (1)	Richtung Osten			Richtung Westen		
	28.6.2006 29.6.2006 Basis	28.5.2008 29.5.2008 B-HC	18.6.2008 19.6.2008 B-GALOP	28.6.2006 29.6.2006 Basis	28.5.2008 29.5.2008 B-HC	18.6.2008 19.6.2008 B-GALOP
06:30-09:00	10	8	12	10	7	12
09:00-15:00	7	14	18	7	12	16
15:00-19:00	10	16	16	12	17	17
06:00-19:00	29	40	47	30	37	45

Route WEST (2)	Richtung Norden			Richtung Süden		
	28.6.2006 29.6.2006 Basis	28.5.2008 29.5.2008 B-HC	18.6.2008 19.6.2008 B-GALOP	28.6.2006 29.6.2006 Basis	28.5.2008 29.5.2008 B-HC	18.6.2008 19.6.2008 B-GALOP
06:30-09:00	5	10	10	6	11	11
09:00-15:00	13	13	17	12	12	17
15:00-19:00	11	14	16	10	14	14
06:00-19:00	30	37	43	29	37	43

Route OST (3)	Richtung Norden			Richtung Süden		
	28.6.2006 29.6.2006 Basis	28.5.2008 29.5.2008 B-HC	18.6.2008 19.6.2008 B-GALOP	28.6.2006 29.6.2006 Basis	28.5.2008 29.5.2008 B-HC	18.6.2008 19.6.2008 B-GALOP
06:30-09:00	12	11	12	10	11	12
09:00-15:00	12	17	20	12	19	30
15:00-19:00	9	15	19	9	14	20
06:00-19:00	33	43	52	32	45	53

Tab. 6.2 Anzahl der ausgewerteten Fahrten

Ergebnisse morgens (6:30-9:00)

Die durchschnittlichen Verlustzeiten und die durchschnittliche Zahl der Halte auf den Routen sind in den Abb. 6.8 dargestellt. Zusätzlich sind die Standardabweichungen eingezeichnet.

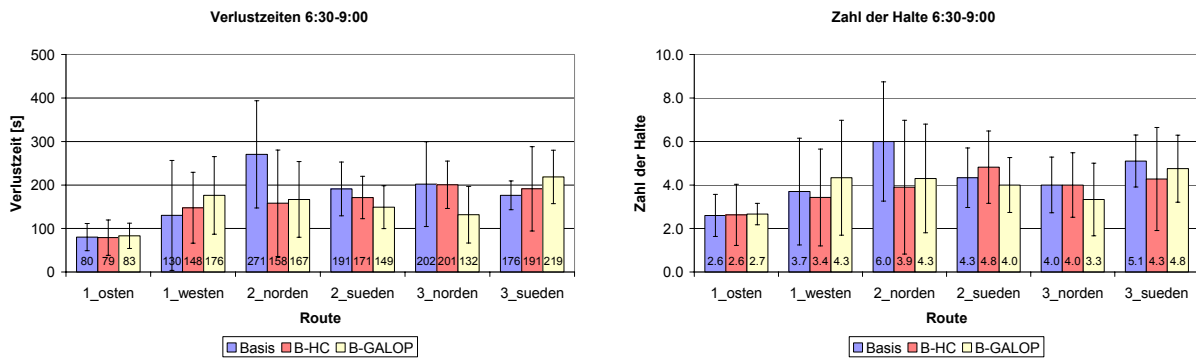


Abb. 6.8 Vergleich der Verlustzeiten und der Zahl der Halte morgens (6:30-9:00 Uhr)

Nach dem t-Test sind die Mittelwertdifferenzen der Verlustzeiten auf der Route OST in Richtung Norden (3_norden) von B-GALOP (132 s) zu B-HC (201 s) und zum Basisszenario (202 s), sowie auf der Route OST in Richtung Süden (3_sueden) zwischen B-GALOP (219 s) und Basis (176 s) auf dem 10%-Niveau signifikant.

Wie aus den eingezeichneten Standardabweichungen erkennbar ist, streuen die Ergebnisse relativ stark. Bei der Zahl der Halte zwischen 6:30 und 9 Uhr ist keine der Mittelwertdifferenzen auf dem 10%-Niveau signifikant.

Ergebnisse tagsüber (9:00-15:00)

Die Abb. 6.9 zeigen die Befahrungsergebnisse auf den sechs Routen für den Zeitbereich von 9:00 bis 15:00 Uhr.

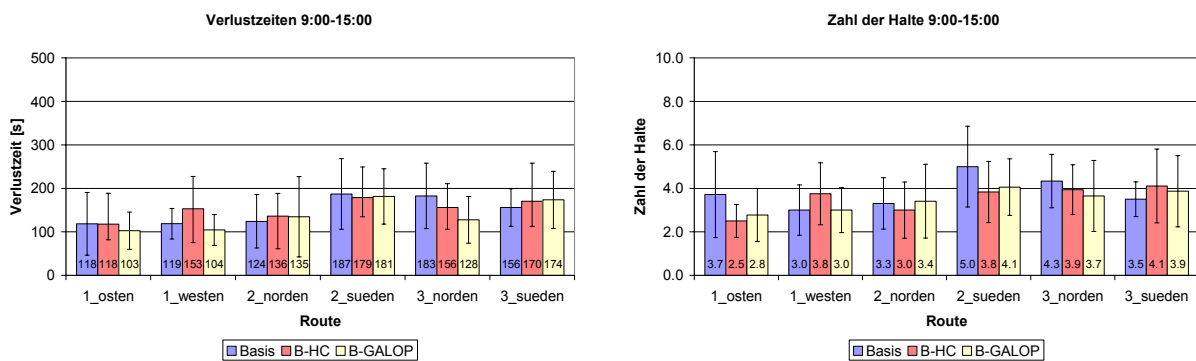


Abb. 6.9 Vergleich der Verlustzeiten und der Zahl der Halte tagsüber (9:00-15:00 Uhr)

Weder bei den durchschnittlichen Verlustzeiten noch bei der mittleren Zahl der Halte können wesentliche Unterschiede festgestellt werden. Signifikant (10%-Niveau) sind die Unterschiede der Verlustzeiten auf der Route OST in Richtung Norden zwischen B-GALOP (128 s) und Basis (183 s) und auf der Route NORD in Richtung Westen zwischen GALOP (104 s) und HC (153 s). Hinsichtlich der Halte ist keine der Mittelwertdifferenzen signifikant.

Ergebnisse abends (15:00-19:00)

Die Ergebnisse der Befahrungen auf den sechs Routen für den Zeitbereich von 15:00 bis 19:00 Uhr sind in den Abb. 6.10 zusammengefasst.

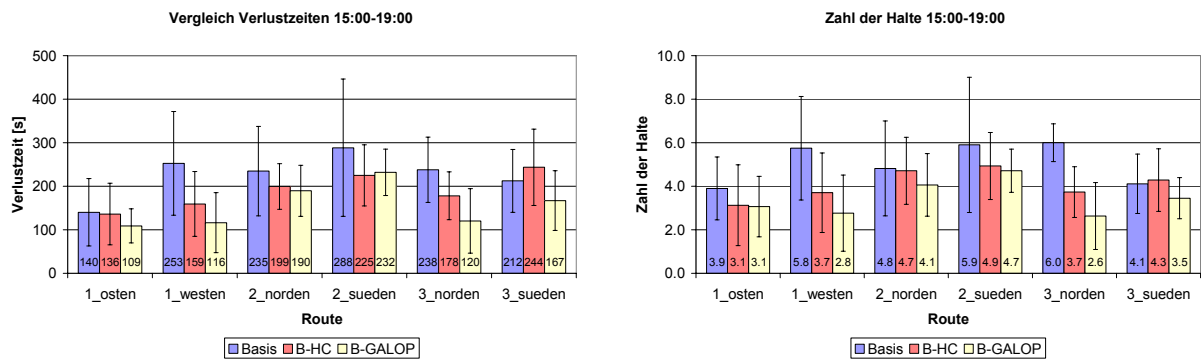


Abb. 6.10 Vergleich der Verlustzeiten und der Zahl der Halte abends (15:00-19:00 Uhr)

Für diesen Zeitbereich sind deutliche Verbesserungen sowohl durch BALANCE mit HC gegenüber dem Basisszenario als auch durch B-GALOP im Vergleich zu B-HC zu erkennen. Signifikant auf 10%-Niveau sind die Differenzen der Verlustzeiten zwischen B-HC und Basis sowie zwischen B-GALOP und Basis auf der Route NORD in Richtung Westen, von B-GALOP zur Basis und zu B-HC auf der Route OST in Richtung Norden, und zwischen GALOP und HC auf der Route OST in Richtung Süden.

Bei der Zahl der Halte ist der Unterschied zwischen B-GALOP und der Basis auf der Route NORD in Richtung Westen und der Route OST in Richtung Norden auf 10%-Niveau signifikant. Auf beiden Routen ist auch die Differenz zwischen BALANCE-HC und der Basis signifikant. Außerdem ist auch der Unterschied zwischen GALOP und HC auf den Routen OST in beiden Richtungen signifikant.

Ergebnisse aller Fahrten

Fasst man alle Fahrten zusammen, erhält man die in den Abb. 6.11 dargestellten Mittelwerte der Verlustzeiten und der Zahl der Halte für die sechs Routen.

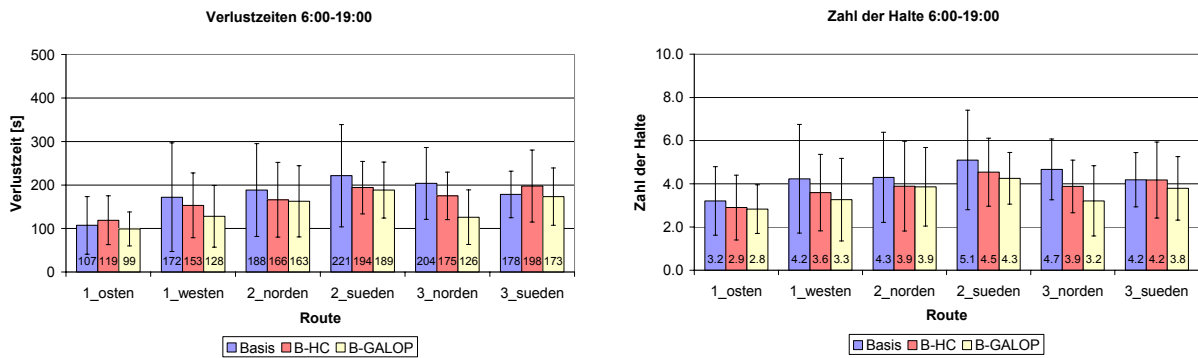


Abb. 6.11 Vergleich der Verlustzeiten und der Zahl der Halte aller Fahrten (6:00-19:00 Uhr)

Auf allen sechs Routen hat BALANCE mit GALOP im Tagesdurchschnitt die geringste Verlustzeit zu verzeichnen. Die Differenz der Mittelwerte der Verlustzeiten ist dabei nach dem t-Test auf einem 10%-Niveau signifikant auf der Route NORD in Richtung Osten zwischen B-HC (119 s) und B-GALOP (99 s), auf der Route NORD in Richtung Westen zwischen B-GALOP (128 s) und Basis (172 s), sowie auf der Route WEST in Richtung Norden zwischen B-GALOP (163 s) und Basis (188 s) wie auch zwischen GALOP und HC (166 s).

Bei den Halten sind die Differenzen zwischen B-GALOP und Basis auf der Route NORD in Richtung Westen, auf der Route WEST in Richtung Süden und auf der Route OST in Richtung Norden signifikant (10%-Niveau). Auf der Route OST in Richtung Norden sind zudem die Unterschiede zwischen B-HC und Basis, sowie zwischen GALOP und HC signifikant.

Summen über alle Routen

Summiert man die Verlustzeiten und die Zahl der Halte aller Routen auf, erhält man für jeden der Zeitbereiche einen Gesamteindruck (Abb. 6.12).

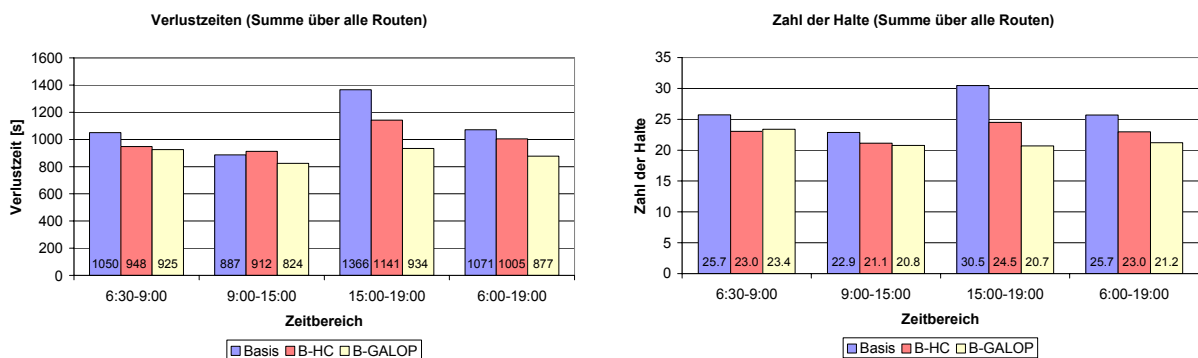


Abb. 6.12 Vergleich der Summen der Verlustzeiten und der Zahl der Halte

Die Summe der durchschnittlichen Verlustzeiten aller Fahrten auf allen Routen (Gesamtlänge 22,7 km) liegt für das Basisszenario bei 1 071 s, für BALANCE mit HC bei 1 005 s und für

BALANCE mit GALOP bei 877 s. Das entspricht einer Reduktion der Verlustzeit durch B-GALOP um durchschnittlich 18% gegenüber dem Basisszenario und um 13% gegenüber B-HC. Die durchschnittliche Zahl der Halte aller Fahrten konnte durch B-GALOP gegenüber dem Basisszenario in der Summe um 17% von 25,7 auf 21,2 reduziert werden.

Die prozentualen Unterschiede der Verlustzeiten und der Zahl der Halte aller Routen sind in Tab. 6.3 zusammengefasst. Je nach Zeitbereich beträgt die Verbesserung von HC zu GALOP durchschnittlich zwischen 2% und 18%.

	B-HC / Basis		B-GALOP / Basis		B-GALOP / B-HC	
	VZ	ZH	VZ	ZH	VZ	ZH
06:30-09:00	-10%	-10%	-12%	-9%	-2%	+1%
09:00-15:00	+3%	-8%	-7%	-9%	-10%	-2%
15:00-19:00	-16%	-20%	-32%	-32%	-18%	-16%
06:00-19:00	-6%	-11%	-18%	-17%	-13%	-8%

Tab. 6.3 Prozentuale Unterschiede der Verlustzeiten (VZ) und Zahl der Halte (ZH) (Befahrungsergebnisse)

Morgens zwischen 6:30 und 9 Uhr konnten gemäß den Befahrungsergebnissen die Verlustzeiten durch B-GALOP insgesamt um 12% im Vergleich zum Basisszenario reduziert werden. Der Unterschied zwischen GALOP und HC beträgt für diesen Zeitbereich im Mittel für alle Routen 2%.

Tagsüber wurde eine Reduzierung der Verlustzeit durch GALOP gegenüber HC um 10% gemessen, die Zahl der Halte wurde um 2% reduziert.

Für den Zeitraum 15 bis 19 Uhr wurden die größten Verbesserungen erreicht. Hier wurde die Verlustzeit im Mittel um 32% durch BALANCE mit GALOP im Vergleich zum Basisfall reduziert. Gegenüber HC konnte GALOP eine Reduktion um 18% erreichen. Die Differenz der Zahl der Halte beträgt zwischen dem Basisszenario und BALANCE mit GALOP insgesamt 32%, gegenüber HC war GALOP demnach um 16% besser.

Im Tagesdurchschnitt wurden gemäß der Befahrungsergebnisse die Verlustzeiten durch GALOP um 13% gegenüber HC reduziert, die Zahl der Halte konnte um 8% verringert werden.

Koordinierungsmaß

Direkt aus der Zahl der Halte bzw. des Prozentsatzes der Durchfahrten ohne Halt kann das Koordinierungsmaß berechnet werden. Daraus lässt sich Qualitätsstufe des Verkehrsablaufs (QSV) gemäß HBS [FGSV, 2001] entsprechend der folgenden Tabelle bestimmen.

QSV	Prozentsatz der Durchfahrten ohne Halt [%]
A	≥ 95
B	≥ 85
C	≥ 75
D	≥ 65
E	≥ 50
F	< 50

Tab. 6.4 Qualitätsstufen des Verkehrsablaufs nach HBS

Das jeweilige Koordinierungsmaß ist in den folgenden Tabellen gemeinsam mit der zugehörigen Qualitätsstufe des Verkehrsablaufs (QSV) dargestellt.

Route NORD (1)	Richtung Osten						Richtung Westen					
	28.6.2006 29.6.2006		28.5.2008 29.5.2008		18.6.2008 19.6.2008		28.6.2006 29.6.2006		28.5.2008 29.5.2008		18.6.2008 19.6.2008	
	Basis		B-HC		B-GALOP		Basis		B-HC		B-GALOP	
06:30-09:00	80	C	80	C	79	C	72	D	74	D	67	D
09:00-15:00	71	D	81	C	79	C	77	C	71	D	77	C
15:00-19:00	70	D	76	C	76	C	56	E	71	D	79	C
06:00-19:00	75	C	78	C	78	C	67	D	72	D	75	D

Tab. 6.5 Koordinierungsmaß auf den Routen NORD

Eine Verbesserung bei der Qualitätsstufe des Verkehrsablaufs ist auf der Route NORD in Richtung Osten für die Zeitbereiche 9:00-15:00 und 15:00-19:00 feststellbar. Tagsüber gab es in Richtung Westen gleichzeitig eine Verschlechterung durch BALANCE-HC, BALANCE-GALOP war demgegenüber wieder genauso gut wie das Basisszenario.

Die deutliche Verbesserung für den Abend in Richtung Westen ist auch an der Änderung der QSV von E über D bis C erkennbar.

Route WEST (2)	Richtung Norden						Richtung Süden					
	28.6.2006 29.6.2006		28.5.2008 29.5.2008		18.6.2008 19.6.2008		28.6.2006 29.6.2006		28.5.2008 29.5.2008		18.6.2008 19.6.2008	
	Basis		B-HC		B-GALOP		Basis		B-HC		B-GALOP	
06:30-09:00	45	F	65	E	61	E	57	E	52	E	60	E
09:00-15:00	70	D	73	D	69	D	50	E	62	E	59	E
15:00-19:00	56	E	57	E	63	E	41	F	51	E	53	E
06:00-19:00	61	E	65	E	65	E	49	F	55	E	57	E

Tab. 6.6 Koordinierungsmaß auf den Routen WEST

Auf den Routen WEST ergaben sich Verbesserungen der QSV in Richtung Norden für den Morgen und in Richtung Süden für den Abend sowie für den Tagesdurchschnitt.

Route OST (3)	Richtung Norden						Richtung Süden					
	28.6.2006 29.6.2006		28.5.2008 29.5.2008		18.6.2008 19.6.2008		28.6.2006 29.6.2006		28.5.2008 29.5.2008		18.6.2008 19.6.2008	
	Basis		B-HC		B-GALOP		Basis		B-HC		B-GALOP	
06:30-09:00	67	D	67	D	72	D	58	E	64	E	60	E
09:00-15:00	64	E	67	D	70	D	71	D	66	D	68	D
15:00-19:00	50	E	69	D	78	C	66	D	64	E	71	D
06:00-19:00	61	E	68	D	73	D	65	D	65	D	68	D

Tab. 6.7 Koordinierungsmaß auf den Routen OST

Die QSV auf der Route OST in Richtung Norden wurde durch B-HC und B-GALOP gegenüber der Basis tagsüber und abends sowie im Tagesdurchschnitt verbessert. Für den Abend gab es durch GALOP gegenüber HC eine Verbesserung um eine Stufe.

In der Gegenrichtung hat sich die QSV nur für den Abend von der Basis von D zu B-HC auf E und für B-GALOP wieder auf D geändert. Allerdings ist der Unterschied des zu Grunde liegenden Prozentsatzes der Durchfahrten ohne Halt zwischen Basis (66%) und B-HC (64%) fast identisch.

6.4.3 Kombinierte Ergebnisse für Verlustzeit

Die Reisezeiten und damit die *Verlustzeiten* wurden auf den Routen NORD sowohl durch Befahrungen, als auch durch Fahrzeugwiedererkennung gemessen. Die Einzelwerte der Reisezeiten des Befahrungsfahrzeugs können für diejenigen Fahrten, bei denen das Messfahrzeug von den Kameras erfasst wurde, direkt mit der zugehörigen Reisezeit aus der

Fahrzeugwiedererkennung verglichen werden. Die Abweichungen liegen im Bereich von +/- 2 s.

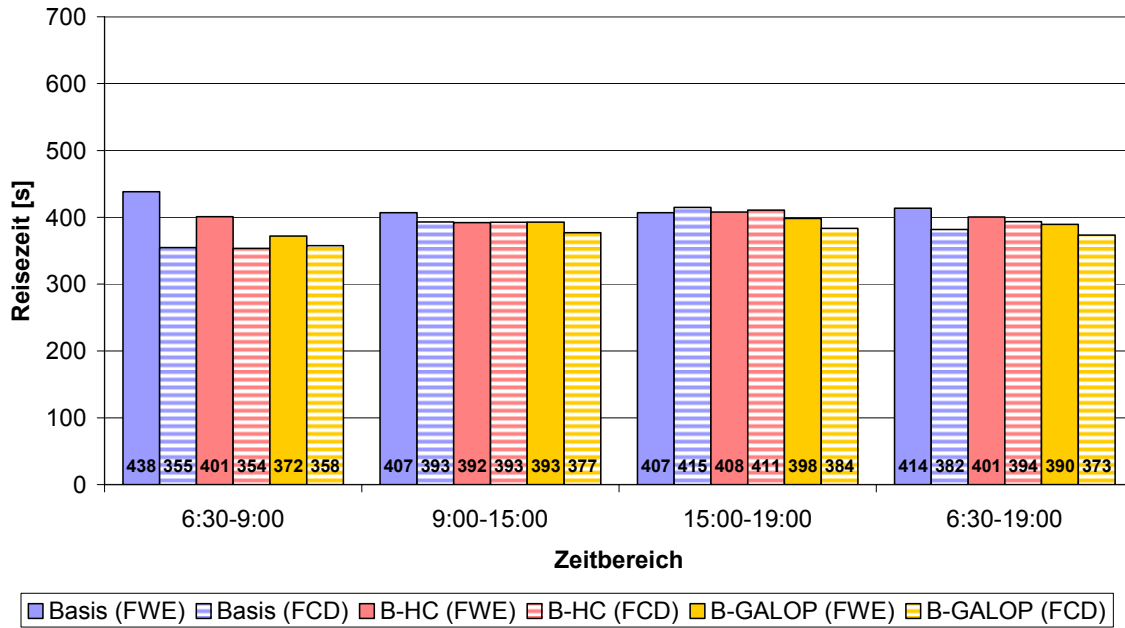


Abb. 6.13 Vergleich der durch Fahrzeugwiedererkennung (je 1 Tag) und Befahrung (je 2 Tage) ermittelten Reisezeiten auf der Route NORD in Richtung Osten

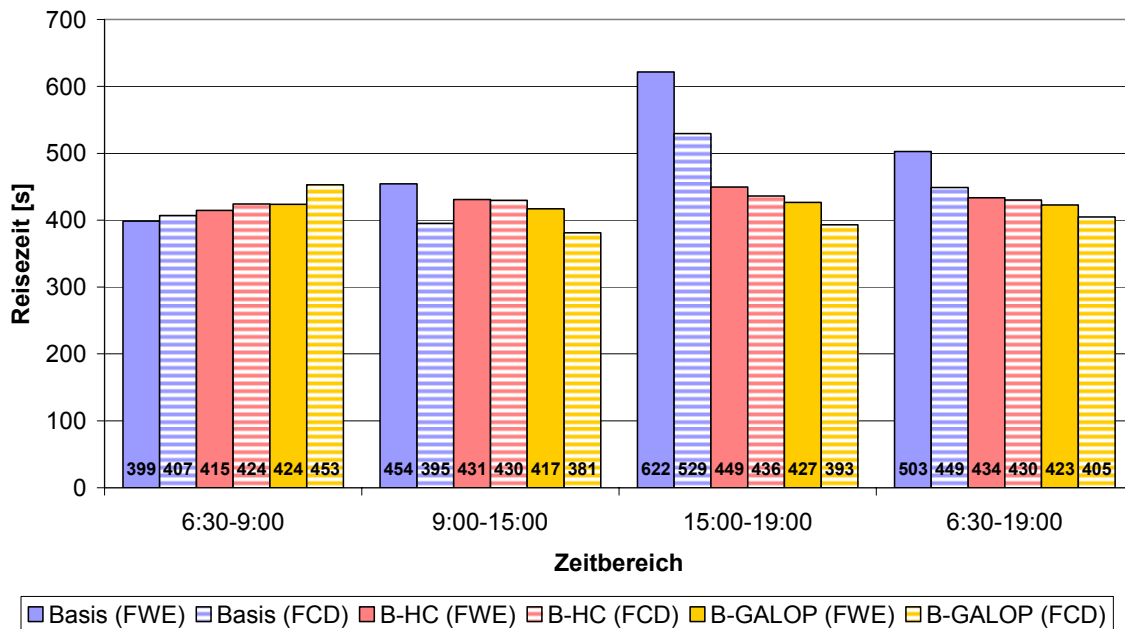


Abb. 6.14 Vergleich der durch Fahrzeugwiedererkennung (je 1 Tag) und Befahrung (je 2 Tage) ermittelten Reisezeiten auf der Route NORD in Richtung Westen

Abb. 6.13 und Abb. 6.14 veranschaulichen die zum Teil deutlichen Unterschiede der *gemittelten* Ergebnisse der Reisezeitmessung durch Befahrungen (Floating Car Data, FCD) und durch Fahrzeugwiedererkennung (FWE).

Auffällig ist insbesondere der Unterschied zwischen der Kameramessung in Richtung Osten für den Zeitbereich 6:30 bis 9 Uhr. Hier werden die Reisezeiten von den Befahrungsergebnissen deutlich unterschätzt. Da das Unterschätzen der Reisezeit allerdings beim Basisszenario mit 83 s stärker als bei BALANCE-HC (47 s) und hier wiederum stärker als bei BALANCE mit GALOP (14 s) ausfällt, wird die Reduktion der Verlustzeiten nur aus der Fahrzeugwiedererkennung deutlich.

In Richtung Westen wird die Reisezeit im Zeitbereich 9 bis 15 Uhr für B-HC im Verhältnis zu den anderen Fällen überschätzt. Ebenso werden die Verbesserungen durch B-HC und B-GALOP im Zeitbereich 15 bis 19 Uhr durch die Befahrungsdaten unterschätzt.

Die gemessenen Reisezeiten aus der Fahrzeugwiedererkennung können grundsätzlich als zuverlässiger angesehen werden. Ersetzt man die durch die Befahrung gemessenen Verlustzeiten auf der Route NORD durch die Verlustzeiten aus der Fahrzeugwiedererkennung, ergeben sich die in Tab. 6.8 dargestellten prozentualen Unterschiede:

	B-HC / Basis	B-GALOP / Basis	B-GALOP / B-HC
06:30-09:00	-12%	-19%	-8%
09:00-15:00	-5%	-9%	-4%
15:00-19:00	-21%	-32%	-15%
06:00-19:00	-12%	-21%	-10%

Tab. 6.8 Prozentuale Unterschiede der Verlustzeiten (Befahrungsergebnisse und Fahrzeugwiedererkennung auf Routen NORD)

Demnach wurden die Verlustzeiten durch BALANCE mit GALOP im Vergleich zum Basis-szenario im Tagesdurchschnitt um 21% reduziert, die reinen Befahrungsergebnisse ergaben eine Reduktion von 18% (vgl. Tab. 6.3). Die Verbesserung durch GALOP zu HC liegt demnach bei 10%, gegenüber 13% aus den reinen Befahrungsergebnissen.

Deutlich wird, dass die Befahrungsergebnisse zwar grundsätzlich dazu geeignet sind, tendenzielle Aussagen über die Qualität unterschiedlicher Steuerungen zu treffen, dass sie aber trotzdem nur eine Stichprobe sind, die relativ stark streut. Grund dafür sind neben dem Einfluss des Fahrverhaltens der unterschiedlichen Fahrer v.a. mehr oder weniger zufällige Ereignisse wie beispielsweise Fußgänger-Anforderungen und ÖPNV-Priorisierungen.

6.5 Fazit aus den Feldversuchsergebnissen

Die Ergebnisse aus der Bewertung von GALOP im Vergleich zu HC (Kapitel 5) konnten durch die Feldversuche bestätigt werden. Die Feldversuchsergebnisse sprechen dafür, dass BALANCE mit GALOP als Optimierungsverfahren auch in der Realität einen Mehrwert bezogen auf die Verlustzeiten und die Zahl der Halte gegenüber BALANCE mit dem bisherigen Hill-Climbing Algorithmus bietet.

7 Zusammenfassung und Ausblick

7.1 Zusammenfassung der Arbeit

In der vorliegenden Arbeit wurde mit GALOP ein echtzeitfähiger Genetischer Algorithmus zur netzweiten Optimierung der Lichtsignalsteuerung entwickelt und bewertet. GALOP optimiert die Signalpläne aller Lichtsignalanlagen des jeweiligen Netzes, wobei alle Steuerungsparameter gleichzeitig optimiert werden können. Daraus können Rahmensignalpläne generiert werden, die dann beispielsweise alle fünf Minuten einer lokalen verkehrsabhängigen Steuerung zur Verfügung gestellt werden können.

Grundannahme war eine einheitliche Umlaufzeit für das zu optimierende Netz. Es handelte sich also um einen klassischen *zentralen* Ansatz. Vorausgesetzt wurden bereits vorhandene Signalpläne mit vorab geplanten möglichen Phasenfolgen (phasenorientierter Ansatz) für die einzelnen Knotenpunkte.

Das Verfahren wurde so konzipiert, dass es tatsächlich *auf die Straße* gebracht werden konnte, weil die Randbedingungen der in Deutschland weit verbreiteten lokalen verkehrsabhängigen Steuerung berücksichtigt werden.

Bei einer Online-Optimierung handelt es sich um ein *zeitabhängiges* Optimierungsproblem, da sich die Verkehrsnachfrage ständig ändert. Für das Optimierungsmodell bedeutet das, dass gleiche Lösungen je nach Verkehrsnachfrage unterschiedliche Zielfunktionswerte haben, d.h. mit der Änderung der Verkehrsstärke ändert sich auch die Gütelandschaft und damit die Lage des globalen Optimums im Suchraum. Ein zeitabhängiges Optimierungsproblem lässt sich jedoch als Folge statischer Optimierungsprobleme betrachten, wobei für jedes Zeitintervall eine optimale Lösung gefunden werden soll. Für jedes dieser Zeitintervalle wird die Verkehrsbelastung als konstant angesetzt.

Das Optimierungsverfahren greift auf das mesoskopische BALANCE-Verkehrsmodell zurück, das als Wirkungskenngrößen für jede Signalgruppe Wartezeiten, Zahl der Halte und Rückstaulängen für den Kraftfahrzeugverkehr liefert. Diese können in einer Zielfunktion zusammengefasst und beliebig gewichtet werden. D.h. die Gewichte können auch so gewählt werden, dass der Zielfunktionswert Kraftstoffverbrauch oder Emissionen widerspiegelt.

Die direkte Integration der Ziele der Fußgänger und Radfahrer sowie des ÖPNV in die Optimierung wurde in dieser Arbeit nicht vorgenommen. Der ÖPNV kann unabhängig von den zentralen Vorgaben des Optimierungsverfahrens durch eine lokale verkehrsabhängige Steuerung sekundengenau priorisiert werden, wie es dem Stand der Technik entspricht. Die Wartezeiten der Fußgänger und Radfahrer können durch die Vorgabe einer (maximalen) Umlaufzeit begrenzt werden.

Die Umsetzung von GALOP fand im Rahmen des Forschungsprojekts TRAVOLUTION statt. Der bisher in der adaptiven Netzsteuerung BALANCE verwendete Hill-Climbing Algorithmus wurde durch GALOP ersetzt. Der Hill-Climbing Algorithmus wurde als eine Referenz für die *Bewertung* des umgesetzten Evolutionären Algorithmus verwendet. Zur Bewertung wurden insgesamt zwölf Fälle aus drei Netzen und vier Verkehrsnachfrageszenarien herangezogen.

Für das Nordnetz, das mit 27 LSA größte der drei untersuchten Netze, ist der Gütewert (Fitness) der durch GALOP gefundenen Lösungen im Vergleich zu den Lösungen des Hill-Climbing Algorithmus je nach Nachfrageszenario um 23% bis 36% besser. Für das Südnetz (11 LSA) konnte GALOP um 0% bis 20% bessere Lösungen erreichen. Für das Westnetz, ein Streckenzug mit acht LSA, fand GALOP für das Nachfrageszenario Morgenspitze nur eine um 2% schlechtere Lösung, für die anderen Szenarien ist die Güte um 3% bis 7% besser. Ganz offensichtlich liefert der Hill-Climbing Algorithmus auf einem relativ kleinen linienhaften Netz wie dem Westnetz genauso gute Ergebnisse wie GALOP. Je größer und komplexer das Netz wird, desto mehr kommen die Vorteile eines Evolutionären Algorithmus, wie insbesondere die gleichzeitige Optimierung aller Steuerungsparameter, zum Tragen. Die Struktur und Stärke der Verkehrsnachfrage scheint dabei keine Rolle für die Algorithmen zu spielen.

Die Parametersensitivität ist vor allem bei den kleineren Netzen sehr gering. Beim Nordnetz verhält sich der Algorithmus etwas sensitiver, liefert allerdings selbst bei ungünstigen Einstellungen ein besseres Ergebnis als der Hill-Climbing Algorithmus. Es ist deshalb zu empfehlen, dass der Algorithmus insbesondere bei komplexeren Netzen ausreichend kalibriert wird, um möglichst gute Lösungen zu finden. Durch das Einlesen guter Lösungen aus früheren Optimierungsläufen kann der Algorithmus noch robuster gemacht werden, weil von Anfang an relativ gutes Genmaterial vorhanden ist.

Im Rahmen des Projekts TRAVOLUTION wurde GALOP in Ingolstadt implementiert. Bei den durchgeführten Feldversuchen wurden drei Steuerungsvarianten miteinander verglichen: das Basisszenario als Vorher-Fall ohne adaptive Netzsteuerung, BALANCE mit Hill-Climbing Algorithmus und BALANCE mit GALOP. In allen drei Fällen besteht eine lokale verkehrsunabhängige Steuerung, in die eine ÖPNV-Priorisierung integriert ist. Außerdem sind einige reine Fußgängerschutzanlagen im Testgebiet, die nur auf Anforderung per Taster reagieren.

Die Qualität der Steuerungen wurde durch Befahrungen und zum Teil durch Reisezeitmessung mittels Fahrzeugwiedererkennung bewertet. Die Auswertungen ergaben, dass die Verlustzeiten durch BALANCE mit GALOP im Vergleich zum Basisszenario im Tagesdurchschnitt um insgesamt 21% reduziert wurden. Die Verbesserung durch GALOP zu HC liegt bei 10%.

7.2 Erweiterungsmöglichkeiten und Ausblick

Zum Abschluss der Arbeit werden einige Erweiterungsmöglichkeiten und Lösungsansätze diskutiert und daraus zum Teil weiterer Forschungsbedarf abgeleitet. Dazu werden im Verlauf der Arbeit nicht umgesetzte oder betrachtete Teilaspekte und festgestellte offene Punkte oder Problemstellungen nochmals aufgegriffen.

Rekombination nach lokaler Fitness

Nicht umgesetzt wurde bisher die Rekombination nach *lokaler Fitness*. Dabei entsteht aus jedem Elternpaar genau ein Nachkomme, der das jeweils bessere Chromosom erhält. Es wäre zu überprüfen, inwieweit durch den Einsatz dieser Rekombinationsvariante für einen Teil der Nachkommen noch schneller als bisher gute Individuen gefunden werden.

Versatzzeitenoptimierung

Auf die Optimierung der Versatzzeiten im Sinn einer Änderung der Koordinierungsstruktur wurde im Rahmen dieser Arbeit verzichtet, obwohl dies die Kodierung prinzipiell zulässt. Grund ist die Annahme, dass sich bei einer starken Änderung der Versatzzeiten von einem zum nächsten Optimierungsintervall unerwünschte Effekte im Verkehrsablauf ergeben, da sich die Koordinierungsstruktur plötzlich wesentlich ändert.

Hierbei spielt v.a. die Tatsache eine Rolle, dass es je nach gegebener Netzstruktur und Nachfragestruktur sehr gute Lösungen mit gleichen Fitnesswerten geben kann, die jedoch eine komplett andere Lösungsform haben. Typisches Beispiel ist eine symmetrische Strecke mit gleicher Nachfrage in beiden Richtungen. Offensichtlich gibt es dafür zwei gleich gute Lösungen, nämlich eine Koordinierung in die eine oder in die andere Richtung. Schränkt man in diesem Fall die Versatzzeitenoptimierung nicht ein, wird es durch kleinste Änderungen der Verkehrsnachfrage zu einer kompletten Änderung der Koordinierungsrichtung von einem zum nächsten Optimierungsintervall kommen. Dafür werden üblicherweise Umschaltprogramme verwendet.

Das verwendete Verkehrsmodell kann die *Änderung* der Signalpläne (inklusive evtl. erforderlicher Umschaltprogramme) und damit der Koordinierungsstruktur nicht abbilden, d.h. die sich daraus ergebenden Effekte können auch in der Optimierung nicht erkannt werden. Allerdings stellt die Optimierung der Freigabezeiten bzw. T-Zeiten bis zu einem gewissen Grad eine *eingeschränkte Versatzzeitenoptimierung* im Rahmen der vorgegebenen Intervalle $[T_{iB}^{\min}; T_{iB}^{\max}]$ dar.

Der Autor ist jedoch der Meinung, dass die Versatzzeiten in einem nächsten Schritt in die Optimierung miteinbezogen werden sollten, um auch die Koordinierungsstruktur online an die Verkehrsnachfrage anpassen zu können. In zukünftigen Forschungsarbeiten wäre dabei

zu untersuchen, wie stark die Versatzzeiten von einem zum nächsten Optimierungsintervall variiert werden dürfen, ohne dass es zu unerwünschten Effekten im Verkehrsablauf kommt. Dafür müssten dann entsprechende Bedingungen im Algorithmus formuliert werden.

Ein weiterer Lösungsansatz ist die Verwendung bzw. Entwicklung eines Verkehrsmodells, das in der Lage ist, die *Signalplanänderung* inklusive evtl. notwendiger Umschaltprogramme mit abzubilden. Bei der Verkehrsflussmodellierung müssen dabei jeweils zuerst die aktuellen Signalpläne, und dann ohne Unterbrechung der Verkehrsflussprofilmodellierung die Signalpläne des jeweiligen Lösungskandidaten gelten.

In jedem Fall sind Mikrosimulationsstudien erforderlich, um die Auswirkungen auf den Verkehrsablauf zu untersuchen.

Problematik der Phasenfolge

In einigen Fällen wird die Phasenfolge von der lokalen verkehrsabhängigen Steuerung bestimmt. Obwohl die Phasenfolge zentral mitoptimiert wird, kann es deshalb vorkommen, dass die lokale verkehrsabhängige Steuerung eine andere, netzweit betrachtet nicht optimale Phasenfolge auswählt. Wird die zentral betrachtet optimale Phasenfolge von der lokalen Steuerung überstimmt, dann hat das zur Folge, dass sich der Verkehrsablauf in der Realität anders darstellt, als er im Verkehrsmodell abgebildet wurde. D.h. es kann nicht sichergestellt werden, dass der andere Phasenablauf, der von der lokalen Steuerung als optimal betrachtet wird, nicht den netzweiten Verkehrsablauf verschlechtert, da er möglicherweise die Koordinierungsstruktur verändert.

Diese Problematik ist zwar keine Einschränkung des Evolutionären Algorithmus, da dieser keine direkte Rückmeldung von der lokalen Steuerung erhält. Allerdings besteht hier offensichtlich Handlungsbedarf beim Zusammenspiel zwischen zentraler und lokaler Steuerung. Als Lösungsmöglichkeit könnte gefordert werden, dass die Phasenfolge ausschließlich zentral optimiert werden darf, dass also die Kompetenz zur Auswahl der Phasenfolge nicht mehr bei der lokalen Steuerung liegt. Dazu muss jedoch sichergestellt sein, dass keine Bedarfsphasen vorhanden sind, d.h. dass Verkehrsströme ausschließlich auf Anforderung über die lokale Steuerung freigegeben werden, da diese sonst je nach Optimierungsergebnis keine Freigabe erhielten. Je nach Vorgaben der Politik wäre die ÖPNV-Priorisierung davon auszunehmen, um eine vollständige Priorisierung zu gewährleisten.

Bisher werden für alle Phasenfolgen die im Lauf der Optimierung jeweils besten Individuen gespeichert. Dadurch wird sichergestellt, dass für jede der möglichen Phasenfolgen aller Knotenpunkte, die nicht im besten Individuum vorkommen aber möglicherweise durch die lokale Steuerung trotzdem geschaltet werden, ein möglichst guter Rahmensignalplan vorliegt. Es wäre zu untersuchen, ob die Verwendung der *lokalen Fitness* - d.h. die Fitness des jeweiligen einzelnen Chromosoms statt der Fitness des ganzen Individuums - als Kriterium

zur Auswahl der netzweit nicht optimalen Phasenfolgen, zu einer Verbesserung führt, die im Rahmen des Algorithmus realisiert werden kann.

Verwendung anderer Zielfunktionen

Ein Evolutionärer Algorithmus ist grundsätzlich in der Lage, beliebige Zielfunktionen zu verarbeiten. Der Einfluss der Zielfunktion auf das Optimierungsergebnis wurde im Rahmen der Arbeit zwar nicht untersucht, es kann jedoch davon ausgegangen werden, dass die Zielfunktion für das Verhalten des Algorithmus selbst keine wesentliche Rolle spielt. Bisher stehen die vom Verkehrsmodell gelieferten Wirkungskenngrößen zur Verfügung, die über Gewichte beliebig kombiniert werden können.

Als Ergänzung wäre die Integration von Fußgängerwartezeiten aus dem Signalplan in die Zielfunktion möglich. Allerdings ist auf Grund fehlender Fußgängerdetektion (abgesehen von einer Anforderung per Taster) häufig nicht bekannt, ob überhaupt Fußgänger betroffen sind.

Untersucht werden könnte weiterhin, welche Auswirkungen eine Änderung der Gewichte in Abhängigkeit vom Verkehrszustand hat. Beispielsweise könnte es sinnvoll sein, bei unter-sättigtem Verkehrszustand die Zahl der Halte hoch zu gewichten, bei übersättigtem Verkehrszustand hingegen die Gewichte der Rückstaulängen hoch zu setzen.

Thema zukünftiger Forschung dürfte eine dynamische und gezielte Änderung der Zielfunktion in Abhängigkeit übergeordneter Vorgaben sein. Solche Vorgaben könnten beispielsweise aus einem Modul kommen, das Umweltmesswerte wie Feinstaubkonzentration, CO₂ oder NO_x auswertet und die einzelnen Gewichte der Zielfunktionsterme (automatisch) verändern kann. Eine dynamische Änderung der Gewichte könnte auch durch ein übergeordnetes Strategiemangement erfolgen, das eine großräumigere Beeinflussung des Verkehrs verfolgt und ggf. auch Umweltmesswerte mitberücksichtigt.

Verkehrsdatenerfassung und Verkehrsmodell

Die Verkehrsdatenerfassung und das darauf aufbauende Verkehrsmodell sind bekanntermaßen die Grundlage für eine Steuerung. Diese Thematik war nicht Gegenstand dieser Arbeit, als Verkehrsmodell wurde das etablierte BALANCE-Verkehrsmodell verwendet. Trotzdem sind auch in diesen Bereichen noch Verbesserungsmöglichkeiten gegeben. Oben genannt wurde bereits der Ansatz, die Auswirkungen von Signalplanänderungen auf den Verkehrsablauf im Verkehrsmodell mit abzubilden.

Um die Verkehrszustandserfassung zu verbessern, könnten weitere Datenquellen wie z.B. fahrzeuggenerierte Daten verwendet und verarbeitet werden. Hierbei wird die Fahrzeug-Infrastruktur Kommunikation zunehmend an Bedeutung gewinnen. Ebenso können aus bereits vorhandenen Daten durch intelligente Verfahren zusätzliche Informationen gewonnen

werden. Ein Beispiel dafür ist das Fuzzy-Expertensystem zur Schätzung der Verlustzeit vor Lichtsignalanlagen mittels halteliniennaher Detektoren [BRAUN, 2002], das entsprechend weiterentwickelt werden könnte.

Ein weiterer Ansatz, der beispielsweise von DÜRR [2001] verfolgt wurde, ist die Integration der Fahrlinien der ÖPNV-Fahrzeuge in das Verkehrsmodell, um diese in die Optimierung mit einbeziehen zu können.

Erweiterung des Evolutionären Algorithmus

Optimierungsstrategien sollten neben einem breiten Abtasten des Suchraums und dem Informationsaustausch über vielversprechende Regionen des Suchraums auch eine gezielte Suche enthalten. Die gezielte Suche wird bei Evolutionären Algorithmen jedoch häufig etwas vernachlässigt, weil sie keine besonderen Voraussetzungen an den Suchraum oder die Zielfunktion stellen. [GERDES, 2004]

Diese Tatsache wurde beim linienhaften Westnetz deutlich, wo der Hill-Climbing Algorithmus und GALOP ähnliche Ergebnisse lieferten. Es wäre zu untersuchen, ob die Kombination aus Evolutionärem Algorithmus und Hill-Climbing Algorithmus (memetischer Algorithmus) noch bessere Ergebnisse als der Evolutionäre Algorithmus liefern würde.

Ansätze zur Beeinflussung des Fahrverhaltens

Aus den zunehmenden Möglichkeiten der Fahrzeug-Infrastruktur Kommunikation ergeben sich Ansätze, das Fahrverhalten zu beeinflussen. Aktuelle Forschungen befassen sich damit, dem Fahrer in Abhängigkeit von prognostizierten Schaltzuständen Empfehlungen für sein Fahrverhalten, z.B. die Anpassung der Geschwindigkeit, zu geben [MENIG, et al., 2008]. Auf diese Weise wird nicht die LSA-Steuerung an den Verkehr angepasst, sondern der Verkehr an die LSA-Steuerung. Diese Ansätze werfen zahlreiche Fragen v.a. in Bezug auf die Auswirkungen auf den Verkehrsablauf auf. Wesentlicher Forschungsbedarf besteht in der Untersuchung der wechselseitigen Abhängigkeiten dieser beiden Ansätze und der Möglichkeiten, diese zu koppeln.

Literatur

- ABU-LEBDEH, G.; BENEKOHAL, R.F. [1999]: Computational issues in micro-genetic algorithms for traffic management. Paper presented at the 78th Annual Meeting of the Transportation Research Board, Washington D.C., Jan. 1999.
- ALMASRI, E. [2006]: A New Offset Optimization Method for Signalized Road Networks. Dissertation an der Fakultät für Bauingenieurwesen und Geodäsie der Universität Hannover. Veröffentlichung des Instituts für Verkehrswirtschaft, Straßenwesen und Städtebau.
- BIELEFELDT, C.; KELLER, H.; PHILIPPS, P.; PLOSS, G. [1986]: Projektstudie zur Entwicklung eines verkehrsabhängigen Steuerungsverfahrens MOTION, im Auftrag der SIEMENS AG, München.
- BOHLINGER, M. [2006]: Koordinierte Lichtsignalsteuerung. OptiV – Erschließung von Entscheidungs- und Optimierungsmethoden für die Anwendung im Verkehr. www.optiv.de (Stand: 09.11.2006)
- BOILLOT, F.; BLOSSEVILLE, J.; LESORT, J.; MOTYKA, V.; PAPAGEORGIOU, M.; SELAM, S. [1992]: Optimal signal control of urban traffic networks. Proc. 6th Int. Conf. Road Traffic Monitoring and Control, 1992, S 75.
- BRANKE, J.; GOLDATE, P.; PROTHMANN, H. [2007]: Actuated traffic signal optimisation using evolutionary algorithms. In Proceedings of the 6th European Congress and Exhibition on Intelligent Transport Systems and Services. ERTICO - ITS Europe.
- BRAUN, R. [2002]: Online-Bewertung Grüner Wellen – Entwicklung eines Fuzzy-Experten-systems zur Schätzung der Verlustzeit vor Lichtsignalanlagen mittels halteliniennaher Detektoren. Diplomarbeit am Fachgebiet für Verkehrstechnik und Verkehrsplanung, TU München.
- BRAUN, R.; MÄNNICHE, S.; WEICHENMEIER, F. [2005]: Anwendung und Bewertung eines Verfahrens zur netzweiten Offline-Optimierung der Lichtsignal-Koordinierung mittels genetischer Algorithmen. Tagungsband Verkehrswissenschaftliche Tage, Dresden, 19.-20. Sept. 2005.
- BRAUN, R.; WEICHENMEIER, F. [2005]: Automatische Offline-Optimierung der lichtsignaltechnischen Koordinierung des MIV im städtischen Netz unter Verwendung genetischer Algorithmen. Straßenverkehrstechnik, 05/2005, FGSV (Hrsg.), Kirschbaum Verlag, Bonn.
- BUSCH, F., KRUSE, G. [1993]: MOTION – ein neues Verfahren für die städtische Lichtsignalsteuerung in Netzen. Tagungsbericht HEUREKA, Karlsruhe, 1993.
- CEYLAN, H. [2006]: Developing combined genetic algorithm - hill-climbing optimization method for area traffic control. Journal of Transportation Engineering, 132(8): S. 663 - 671.
- CEYLAN, H.; BELL, M.G.H. [2004]: Traffic signal optimization based on genetic algorithm approach, including drivers' routing. Transportation Research Part B 38, S. 329-342.
- CEYLAN, H.; BELL, M.G.H. [2005]: Genetic algorithm solution for the stochastic equilibrium transportation networks under congestion. Transportation Research Part B 39, S. 169-185.

DIAKAKI, C., DINOPOULOU, V., ABOUDOLAS, K., PAPAGEORGIOU, M., BEN-SHABAT, E., SEIDER, E., LEIBOV, A. [2003]: Extensions and new applications of the traffic signal control strategy TUC. Transportation Research Record No. 1856, pp. 202-211.

DONATI, F.; MAURO, V., RONCOLINI, G.; VALLAURI, M. [1984]: A Hierarchical Decentralised Traffic Light Control System. The First Realisation: 'Progetto Torino'. IFAC 9th World Congress, Vol II, 11G/A-1.

DÜRR, P. [2001]: Integration des ÖPNV in die dynamische Fahrwegsteuerung des Straßenverkehrs – Steuerungsverfahren DARVIN (Dissertation). Veröffentlichung des Lehrstuhls für Verkehrs- und Stadtplanung, TU München.

DÜRR, P. [2002]: Evolution adaptiver Steuerungsverfahren für den städtischen Mischverkehr. Forschungsgesellschaft für Straßen- und Verkehrswesen (Hrsg.) Tagungsbericht der HEU-REKA '02, Optimierung in Verkehr und Transport, FGSV Verlag, Köln, S. 351-367.

FORSCHUNGSGESELLSCHAFT FÜR STRAßEN- UND VERKEHRSWESEN (Hrsg.) [FGSV, 1998]: RiLSA - Richtlinien für Lichtsignalanlagen - Lichtzeichenanlagen für den Straßenverkehr. Ausgabe 1992, Berichtiger Nachdruck 1998. FGSV-Nr. 321, FGSV Verlag, Köln.

FORSCHUNGSGESELLSCHAFT FÜR STRAßEN- UND VERKEHRSWESEN (Hrsg.) [FGSV, 2003]: RiLSA - Richtlinien für Lichtsignalanlagen - Lichtzeichenanlagen für den Straßenverkehr - Teilfortschreibung 2003. FGSV-Nr. 321/1, FGSV Verlag, Köln.

FORSCHUNGSGESELLSCHAFT FÜR STRAßEN- UND VERKEHRSWESEN (Hrsg.) [FGSV, 2005]: HBS 2001, Fassung 2005 - Handbuch für die Bemessung von Straßenverkehrsanlagen. FGSV-Nr. 299, FGSV Verlag, Köln.

FOY, M.D.; BENEKHAL, R.F.; GOLDBERG, D.E. [1992]: Signal timing determination using genetic algorithms. Transportation Research Record 1365, S. 108-115.

FRIEDRICH, B. [1999]: Ein verkehrsabhängiges Verfahren zur Steuerung von Lichtsignalanlagen (Dissertation). Veröffentlichung des Fachgebiets Verkehrstechnik und Verkehrsplanung, TU München.

FRIEDRICH, B. [2002A]: Verkehrsadaptive Steuerung von Lichtsignalanlagen - Ein Überblick. Festschrift zum Ehrenkolloquium für Univ.-Prof. Dr./UCB Hartmut Keller 01.03.2002 an der TU München, Veröffentlichungen des Fachgebiets Verkehrstechnik und Verkehrsplanung der Technischen Universität München, München.

FRIEDRICH, B. [2002B]: Adaptive Signal Control: an Overview. Extended Abstract for 9th Meeting of the Euro Working Group Transportation, 11.-14. Juni 2002, Bari, Italien.

FRIEDRICH, B.; KEMPER, C. [2006]: Betrieb von Straßenverkehrsanlagen – Vertiefungsstudium. Vorlesungsskript. Hannover, Institut für Verkehrswirtschaft, Straßenwesen und Städtebau (ivh), Universität Hannover.

FRIEDRICH, B.; MERTZ, J.; ERNHOFER, O.; CLARK, M.; TOOMEY, C.; MCLEAN, T.; BRADER, C.; FRAYNE, B. [1998]: Urban Traffic Control with PT Priority: Final Evaluation Report, Deliverable 9.2 EU-Project Tabasco, European Commission, DG XIII, Brüssel.

- FRIEDRICH, B.; SHAHIN, M. [2002]: Adaptive Traffic Control in Metropolitan Areas. Proc. of the 4th International Conference on the Role of Engineering Towards a Better Environment RETBE'02, Dec. 21 - 23, 2002, Alexandria, Egypt.
- GARTNER, N.H. [1983]: OPAC: A demand responsive strategy for traffic signal control. Transportation Research Record 906, National Research Council, Washington, D.C.
- GERDES, I.; KLAWONN, F.; KRUSE, R. [2004]: Evolutionäre Algorithmen. Vieweg Verlag, Wiesbaden.
- GREFENSTETTE, J.J. [1986]: Optimization of control parameters for genetic algorithms. IEEE Transactions on Systems, Man and Cybernetics, Vol. SMC-16, 1, S. 122-128.
- HADI, M.A.; WALLACE, C.E. [1993]: Hybrid genetic algorithm to optimize signal phasing and timing. Transportation Research Record 1421, S. 104-112.
- HALE, D.K. [2005]: Traffic Network Study Tool TRANSYT-7F, United States Version. McTrans Center, University of Florida, Gainesville, Florida.
- HANSEN, N.; A. OSTERMEIER [2001]: Completely Derandomized Self-Adaptation in Evolution Strategies. Evolutionary Computation, 9(2), pp. 159-195.
- HAUENSTEIN, A. [2002]: Simulation und autodynamische Störfolgenbeseitigung im Steuerungs- und Dispositionsverfahren SALTOS (Dissertation). Veröffentlichung des Lehrstuhls für Verkehrs- und Stadtplanung, TU München.
- HENRY, J.J.; FARGES, J.L.; TUFAL, J. [1983]: The PROLYN real time traffic algorithm. Proc. of the IFAC Symposium, Baden-Baden.
- HILDEBRAND, L. [2005]: Grundlagen und Anwendungen der Computational Intelligence II: Evolutionäre Algorithmen. Vorlesungsmaterialien Informatik I, Universität Dortmund. <http://ls1-www.cs.uni-dortmund.de/~hildebra/Vorlesungen/GACI%20II%20EA/GACI2-gr.html> (Stand: 18.1.2005).
- HOUNSELL, N.B.; McDONALD, M. [2001]: Urban network traffic control. Proceedings of the Institution of Mechanical Engineers Part I: Journal of Systems and Control Engineering, 215, (4), 325-334.
- HUNT, P.B.; ROBERTSON, D.I.; BRETHERTON, R.D.; WINTON, R.I. [1981]: SCOOT – a traffic responsive method of coordinating signals. TRRL Laboratory Report 1014.
- JAHN, P. [2006]: Informationen über Lichtsignalanlagen im Straßenverkehr. www.info-lsa.de (Stand 11.01.2006).
- KEIJZER, M.; MERELO, J. J.; ROMERO, G.; SCHOENAUER, M. [2001]: Evolving Objects: a general purpose evolutionary computation library. 5th International Conference on Artificial Evolution, Université de Bourgogne, France.
- KIMBER, R.M.; HOLLIS, E.M. [1979]: Traffic queues and delays at road junctions; TRRL Laboratory Report 909.

KOST, B. [2003]: Optimierung mit Evolutionsstrategien. Verlag Harri Deutsch, Frankfurt am Main.

LÄMMER, S. [2007]: Reglerentwurf zur dezentralen Online-Steuerung von Lichtsignalanlagen in Straßennetzwerken (Dissertation). Fakultät Verkehrswissenschaften, Technische Universität Dresden. <http://nbn-resolving.de/urn:nbn:de:swb:14-1194272623825-42598>.

LOWRIE, P.R. [1982]: The Sydney co-ordinated adaptive traffic system: principles, methodology, algorithms. In Proceedings of Institution of Electrical Engineers International Conference on Road Traffic Signalling, London.

MARTIN, P.; PERRIN, J.; CHILUKURI, B.; JHAVERI, C.; FENG, Y. [2003]: Adaptive Signal Control II. Department of Civil and Environmental Engineering, University of Utah.

McTRANS [2008]: Traffic Network Study Tool, TRANSYT-7F, United States Version. <http://mctrans.ce.ufl.edu/featured/TRANSYT-7F> (Stand 09.06.2008).

MENIG, C.; HILDEBRANDT, R.; BRAUN, R. [2008]: Der informierte Fahrer – Optimierung des Verkehrsablaufs durch LSA-Fahrzeug-Kommunikation. Forschungsgesellschaft für Straßen- und Verkehrswesen (Hrsg.), HEUREKA '08, Optimierung in Verkehr und Transport - Tagungsband, FGSV Verlag, Köln, ISBN 978-3-939715-48-1.

MERTZ, J. [1996]: Munich Comfort, Abschlussbericht, Arbeitsbereich 'Städtische Verkehrssteuerung', 444, Fachgebiet Verkehrstechnik und Verkehrsplanung, TU München.

MERTZ, J. [2001]: Ein mikroskopisches Verfahren zur verkehrsadaptiven Knotenpunktsteuerung mit Vorrang des öffentlichen Verkehrs (Dissertation). Veröffentlichung des Fachgebiets Verkehrstechnik und Verkehrsplanung, TU München.

MERTZ, J.; WEICHENMEIER, F.; BERGMANN, A.; HAFTMANN, A. [2000]: ManaH – Management von Hauptstraßennetzen, A1 Integrierte Lichtsignalsteuerung, Fachgebiet Verkehrstechnik und Verkehrsplanung, Technische Universität München.

MICHALEWICZ, Z. [1996]: Genetic Algorithms + Data Structures = Evolution Programs. Springer, Berlin, 3rd Edition.

MÜCK, J. [2008]: Schätz- und Optimierungsverfahren in der Adaptiven Netzsteuerung SI-TRAFFIC Motion MX, Proceedings of Heureka 2008, Stuttgart, Forschungsgesellschaft für Straßen- und Verkehrswesen, Köln.

PAPAGEORGIU, M. [1996]: Optimierung. Statische, dynamische, stochastische Verfahren für die Anwendung. 2. Auflage. R. Oldenbourg Verlag München Wien.

PAPAGEORGIU, M., DIAKAKI, C., DINOPOULOU, V., KOTSIALOS, A., WANG, Y. [2003]: Review of road traffic control strategies. Proceedings of the IEEE, Vol. 91, No. 12, S. 2043-2067.

PARK, B.; MESSER, C.; URBANIK II, T. [1999]: Traffic Signal Optimization Program for Oversaturated Conditions: A Genetic Algorithm Approach. Paper presented at the 78th Annual Meeting of the Transportation Research Board (Paper No. 991507), Washington, D.C. Jan. 1999.

POHLHEIM, H. [1999]: Evolutionäre Algorithmen. Springer Verlag, Berlin.

ROBERTSON, D.I. [1969]: TRANSYT: A Traffic Network Study Tool. RRL Report LR 253, Road Research Laboratory, Crowthorne, Berkshire.

SADEK, A.W.; SMITH, B.L.; DEMETSKY, M.J. [1997]: Dynamic Traffic Assignment: A Genetic Algorithm Approach. Paper presented at the 76th Annual Meeting of the Transportation Research Board, Washington, D.C. Jan. 1997.

SCHNABEL, W.; LOHSE, D. [1997]: Grundlagen der Straßenverkehrstechnik und der Verkehrsplanung. Bd. 1 Verkehrsplanung, 2. Auflage.

SCHNIEDER, E. (Hrsg.) [2007]: Verkehrsleittechnik. Automatisierung des Straßen- und Schienenverkehrs. Springer-Verlag, Berlin Heidelberg.

SCHWEFEL, H.-P. [1977]: Numerische Optimierung von Computer-Modellen mittels der Evolutionsstrategie. Birkhäuser Verlag, Basel und Stuttgart.

SCHWEFEL, H.-P. [1995]: Evolution and Optimum Seeking. John Wiley & Sons, Inc., New York.

SCOOT [2008]: Split Cycle Offset Optimisation Technique. Peek Traffic Limited, Siemens Traffic Controls, TRL Limited. www.scoot-utc.com. (Stand: 26.08.2008)

SEN, S.; HEAD, K.L. [1997]: Controlled optimization of phases at an intersection. Transport. Sci. 31, S. 5–17.

STEVANOVIC, A.; MARTIN, P.T.; STEVANOVIC, J. [2007]: VisSim-Based Genetic Algorithm Optimization of Signal Timings. Transportation Research Record: Journal of the Transportation Research Board. Volume 2035 / 2007, S. 59-68.

TRANSPORT RESEARCH LABORATORY, TRL [2008]: www.trl.co.uk, www.trlsoftware.co.uk (Stand: 26.08.2008).

WEICKER, K. [2007]: Evolutionäre Algorithmen. 2., überarbeitete und erweiterte Auflage. Teubner Verlag, Wiesbaden.

YUN, I.; PARK, B. [2006]: Stochastic Optimization Method for Coordinated Actuated Signal Systems. Research Report No. UVACTS-15-0-102, Center for Transportation Studies, University of Virginia.

Glossar

BALANCE	<u>B</u> alancing <u>A</u> daptive <u>N</u> etwork <u>C</u> ontrol <u>M</u> ethod
CAN-Bus	Controller Area Network-Bus
CORSIM	<u>C</u> orridor <u>S</u> imulation. Mikroskopisches Verkehrssimulationsprogramm der US-amerikanischen Federal Highway Administration.
CPU	Central Processing Unit. Hauptprozessor
EA	Evolutionärer Algorithmus
EO	Evolving Objects
ES	Evolutionsstrategie
FCD	Floating Car Data. Fahrzeuggenerierte Daten
FGSV	Forschungsgesellschaft für Straßen- und Verkehrswesen
FSA	Fußgängerschutzanlage
FWE	Fahrzeugwiedererkennung
GA	Genetischer Algorithmus
GALOP	<u>G</u> enetischer <u>A</u> lgorithmus zur netzweiten <u>O</u> ptimierung der Lichtsignalsteuerung <u>G</u> enetic <u>A</u> lgorithm for network-wide Traffic <u>L</u> ights <u>O</u> ptimization
GPS	Global Positioning System, Globales Positionsbestimmungssystem
HBS	Handbuch für die Bemessung von Straßenverkehrsanlagen der FGSV
HC	Hill-Climbing
Krad	Kraftrad
Lkw	Lastkraftwagen
LSA	Lichtsignalanlage(n) = Ampel(n)
MOTION	<u>M</u> ethode zur <u>O</u> ptimierung der Lichtsignalsteuerung in <u>o</u> nline gesteuerten <u>N</u> etzen
nbOffspring	Anzahl der Nachkommen (%)
OPAC	<u>O</u> ptimization <u>P</u> olicies for <u>A</u> daptive <u>C</u> ontrol
ÖPNV	Öffentlicher Personennahverkehr
pCross	Rekombinationswahrscheinlichkeit

Phasenübergang	Zeitdauer zwischen zwei Phasen (Phasenwechsel). Enthält alle Zwischenzeiten und ggf. Mindestfreigabezeiten.
Pkw	Personenkraftwagen
pMut	Mutationswahrscheinlichkeit
Population	Gruppe von Individuen
QSV	Qualitätsstufe des Verkehrsablaufs
Rahmensignalplan	Enthält für jeden Phasenübergang den frühesten Beginn und das späteste Ende
RiLSA	Richtlinien für Lichtsignalanlagen der FGSV
SCATS	<u>S</u> dney <u>c</u> o-ordinated <u>a</u> daptive <u>t</u> raffic <u>s</u> ystem
SG	Signalgruppe. Alle Signalgeber einer Signalgruppe steuern den gleichen Verkehrsstrom und zeigen zu jeder Zeit das gleiche Signalbild.
signalnit	Anfangsmutationsschrittweite
Signalplan	Signalzeitenplan
Signalprogramm	Enthält einen oder mehrere Signalpläne (Phasenfolgen) pro LSA
TL	Testlauf
TRAVOLUTION	<u>T</u> raffic and <u>E</u> volution. Ein Verkehrsforschungsprojekt gefördert vom Freistaat Bayern
t-Test	Statistischer Hypothesentest über Erwartungswerte einer oder zweier normalverteilter Grundgesamtheiten mit t-verteilter Testprüfgröße
T-Zeit	Startzeitpunkt eines Phasenübergangs (= Endzeitpunkt einer Phase)
T-Zeit-Grenzen	Zeitraumen, innerhalb dessen die lokale verkehrsabhängige Steuerung eine Phase vorzeitig abbrechen kann
UTOPIA	<u>U</u> rban <u>T</u> raffic <u>O</u> ptimization by <u>I</u> ntegrated <u>A</u> utomation
Versatzzeit	Zeitliche Differenz des Referenzpunkts des lokalen Signalplans zur globalen Zeitskala
VISSIM	<u>V</u> erkehr <u>i</u> n <u>S</u> tädten – <u>S</u> imulation. Mikroskopisches Verkehrssimulationsprogramm der PTV AG

Abbildungsverzeichnis

Abb. 1.1	Zwei Dimensionen eines 20-dimensionalen Lösungsraums.....	2
Abb. 2.1	SCOOT-Verkehrsmodell (aus FRIEDRICH und KEMPER [2006] in Anlehnung an HUNT et al. [1981]).....	20
Abb. 3.1	Allgemeiner Regelkreis	25
Abb. 3.2	Ablauf der Optimierung.....	25
Abb. 4.1	Allgemeiner Ablauf eines Evolutionären Algorithmus	32
Abb. 4.2	Entwurfsmethodik (in Anlehnung an WEICKER [2007]).....	34
Abb. 4.3	T-Zeit-Grenzen für die lokale verkehrsabhängige Steuerung	37
Abb. 4.4	Relative Kodierung von Umlaufzeit und Freigabezeiten	41
Abb. 4.5	Berechnung der Intervallgrenzen für T_i	43
Abb. 4.6	Zusammensetzung der Startpopulation	44
Abb. 4.7	Funktionsweise der Turnierselektion	46
Abb. 4.8	Prinzip der chromosomweisen Rekombination	48
Abb. 5.1	Drei Netze in Ingolstadt (Karte: Amt für Verkehrsmanagement und Geoinformation, Stadt Ingolstadt).....	57
Abb. 5.2	Histogramm der Fitnesswerte von 100 000 zufällig erzeugten Individuen	59
Abb. 5.3	Einfluss der Startzufallszahl	60
Abb. 5.4	Vergleich der besten erreichten Fitness.....	63
Abb. 5.5	Vergleich der durchschnittlichen Besten Fitness.....	63
Abb. 5.6	Vergleich der durchschnittlichen Fitness aller bewerteten Individuen.....	64
Abb. 5.7	Gesamte Rechenzeit eines Optimierungslaufs [min]	64
Abb. 5.8	Rechenzeit pro Individuum [s].....	65
Abb. 5.9	Vergleich unterschiedlicher Rekombinationsvarianten für drei Szenarien.....	67
Abb. 5.10	Einfluss Rekombinations- und Mutationswahrscheinlichkeit für drei Szenarien	69
Abb. 5.11	Einfluss der Anzahl der Nachkommen für drei Szenarien.....	71
Abb. 5.12	Verlauf der besten Fitness für eine unterschiedliche Anzahl an Nachkommen	72
Abb. 5.13	Verlauf der Standardabweichung für eine unterschiedliche Anzahl an Nachkommen.....	73
Abb. 5.14	Einfluss der adaptiven Startmutationsschrittweite für drei Szenarien	74
Abb. 5.15	Vergleich der adaptiven Mutationsschrittweiten der ersten und letzten Generation für drei Szenarien	75
Abb. 5.16	Ergebnisse der Bewertung (Qualität der Bestlösung).....	77
Abb. 5.17	Vergleich der Fitnessentwicklung von GALOP und HC	79
Abb. 6.1	Testgebiet Ingolstadt mit gemessenen Routen	82
Abb. 6.2	Ganglinienvergleich Routen NORD (FSA36)	83
Abb. 6.3	Ganglinienvergleich Routen WEST (LSA E5a)	84

Abb. 6.4	Ganglinienvergleich Routen OST (LSA F2b)	84
Abb. 6.5	Reisezeiten auf der Route NORD Richtung Osten.....	86
Abb. 6.6	Reisezeiten auf der Route NORD Richtung Westen	86
Abb. 6.7	Vergleich der Verlustzeiten aus der Fahrzeugwiedererkennung	87
Abb. 6.8	Vergleich der Verlustzeiten und der Zahl der Halte morgens (6:30-9:00 Uhr).....	90
Abb. 6.9	Vergleich der Verlustzeiten und der Zahl der Halte tagsüber (9:00-15:00 Uhr)....	90
Abb. 6.10	Vergleich der Verlustzeiten und der Zahl der Halte abends (15:00-19:00 Uhr)....	91
Abb. 6.11	Vergleich der Verlustzeiten und der Zahl der Halte aller Fahrten (6:00-19:00 Uhr).....	92
Abb. 6.12	Vergleich der Summen der Verlustzeiten und der Zahl der Halte	92
Abb. 6.13	Vergleich der durch Fahrzeugwiedererkennung (je 1 Tag) und Befahrung (je 2 Tage) ermittelten Reisezeiten auf der Route NORD in Richtung Osten	96
Abb. 6.14	Vergleich der durch Fahrzeugwiedererkennung (je 1 Tag) und Befahrung (je 2 Tage) ermittelten Reisezeiten auf der Route NORD in Richtung Westen.....	96

Tabellenverzeichnis

Tab. 1.1	Größe des Lösungsraums.....	3
Tab. 5.1	Testszenarios für Kalibrierung, Sensitivitätsuntersuchung und Bewertung	58
Tab. 5.2	Ergebnisse der Konsistenzuntersuchung	61
Tab. 5.3	Prozentuale Unterschiede der Optimierungsergebnisse zwischen GALOP und HC.....	77
Tab. 6.1	Übersicht Messtage.....	81
Tab. 6.2	Anzahl der ausgewerteten Fahrten	89
Tab. 6.3	Prozentuale Unterschiede der Verlustzeiten (VZ) und Zahl der Halte (ZH) (Befahrungsergebnisse)	93
Tab. 6.4	Qualitätsstufen des Verkehrsablaufs nach HBS	94
Tab. 6.5	Koordinierungsmaß auf den Routen NORD.....	94
Tab. 6.6	Koordinierungsmaß auf den Routen WEST	95
Tab. 6.7	Koordinierungsmaß auf den Routen OST.....	95
Tab. 6.8	Prozentuale Unterschiede der Verlustzeiten (Befahrungsergebnisse und Fahrzeugwiedererkennung auf Routen NORD).....	97

