

Ein Verfahren der sequentiellen, konvexen
Optimierung mit kombinierter
Trust-Region- und
Moving-Asymptotes-Stabilisierung zur
Lösung nichtlinearer, restringierter
Optimierungsprobleme

Axel Luthardt

UNIVERSITÄT BAYREUTH
FAKULTÄT FÜR MATHEMATIK, PHYSIK UND INFORMATIK

Ein Verfahren der sequentiellen, konvexen
Optimierung mit kombinierter
Trust-Region- und
Moving-Asymptotes-Stabilisierung zur
Lösung nichtlinearer, restringierter
Optimierungsprobleme

Von der Universität Bayreuth
zur Erlangung des akademischen Grades eines
Doktors der Naturwissenschaften (Dr. rer. nat.)
genehmigte Abhandlung

vorgelegt von
Axel Luthardt
geboren in Coburg

1. Gutachter: K. Schittkowski
2. Gutachter: F. Jarre
2. Gutachter: H. J. Pesch

Tag der Einreichung: 12. November 2012

Tag des Kolloquiums: 30. April 2013

Zusammenfassung

Ein neues Verfahren der sequentiellen konvexen Optimierung (SCP) zur Lösung allgemeiner kontinuierlicher und restringierter nichtlinearer Optimierungsprobleme (NLP) wird vorgestellt, das die Approximation der „Method of Moving Asymptotes“ (MMA) mit einer Trust-Region-Strategie kombiniert. Als Trust-Region für diese Methode wird das Gebiet zwischen den MMA-Asymptoten abzüglich einem festen Sicherheitsabstand definiert, um die Beschränktheit der Approximationen zu gewährleisten. Die Asymptoten, die notwendig für das Aufstellen der Approximationen sind, werden implizit über die Steuerung des Trust-Region-Radius im Sinne einer Trust-Region-Methode erzeugt. Der Algorithmus „Trust-Region Sequential Convex Programming - TRSCP“ wird vorgestellt und die globale Konvergenzeigenschaft des neuen Verfahrens wird nachgewiesen. TRSCP ist sowohl ein SCP-Verfahren, als auch eine Trust-Region-Methode. Im Unterschied zu anderen Ansätzen auf diesem Gebiet, benötigt TRSCP keine explizite Trust-Region zusätzlich zu den MMA-Asymptoten. Darüber hinaus behindert eine aktive Trust-Region nicht die globale Konvergenzeigenschaft des neuen Verfahrens. TRSCP ist im Programm TRSCP1.0 implementiert. Es wird gezeigt, daß TRSCP1.0 ein leistungsfähiges Programm zur Lösung von nichtlinearen Optimierungsproblemen ist und ein Vergleich mit einer Line-Search-Methode wird durchgeführt.

Abstract

Title: A method of sequential convex programming with a combined trust-region/move-limits stabilization for solving nonlinear constrained optimization problems

A new sequential convex programming (SCP) method for solving general continuous and constrained non-linear programs (NLP) is introduced combining the approximation of the method of moving asymptotes (MMA) with a trust-region strategy. The trust-region of this method is the area between the asymptotes defining the MMA-approximations, minus a fixed safety distance due to boundedness of the derivations of the approximations. The asymptotes necessary for the approximations are generated implicitly by controlling the trust-region in the sense of a trust-region method. A proof of global convergence of the new method is given and the algorithm "Trust-Region Sequential Convex Programming - TRSCP" is presented. Hence TRSCP is the implementation of a SCP- as well as a trust-region method. Different to other approaches in this area, TRSCP does not need an explicit trust-region and asymptotes combination. Furthermore, an active trust-region does not prevent convergence properties of the new method. TRSCP has been implemented in the program TRSCP1.0. It will be shown that TRSCP1.0 is a powerful tool for solving continuous nonlinear programs. A comparison between TRSCP1.0 and a line-search method is performed.

Symbolverzeichnis

SCP	Sequential Convex Programming	3
TRSCP	Trust-Region Sequential Convex Programming	3
LSSCP	Line-Search Sequential Convex Programming	81
NLP	Non-Linear Program	3
OP	OptimierungsProblem	22
EOP	Erweitertes OptimierungsProblem	83
CONLIN	CONvex LINearisation	4
MMA	Method of Moving Asymptotes	4
FMO	Free Material Optimization	5
SQP	Sequential Quadratic Programming	11
BCP	Box-Constraint Problem	12
CCSA	Conservative Convex Seperable Approximation	20
KKT	Karush-Kuhn-Tucker	17
BFGS	Broyden-Fletcher-Goldfarb-Shanno	41
LICQ	Linear Independant Constraint Qualification	27
TRSCP1.0	Programm mit Implementierung des TRSCP-Algorithmus	15
SCPIP3.0	Programm mit Implementierung des LSSCP-Algorithmus	15
x	Primale Variable	3
x^*	Primale Lösung	28
x^{Eingabe}	Eingabe ins Programm TRSCP 1.0 als Schätzung für einen stationären Punkt	84
u	Duale Variable (Lagrange-Multiplikator)	28
u^*	Duale Lösung	28
v_{max}	Schranke für Lagrange-Multiplikatoren	30
\bar{u}	spezieller Vektor aus dem Lagrange-Multiplikator-Anteil für aktive Nebenbedingungen	30

y	primale Teilproblemlösung	26
v	duale Teilproblemlösung	31
w	Anteil der dualen Teilproblemlösung durch Trust-Region-Nebenbedingungen	50
z	Entwicklungspunkt	4
s	Suchrichtung	31
δ	Abstand zwischen Entwicklungspunkt und primaler Teilproblemlösung	32
k	Iterationsindex	22
p	Teilterationsindex Penalty-Parameter Anpassung	5
q	Teilterationsindex innere Iteration	9
\bar{q}	Schranke für Anzahl an inneren Iterationen	85
n	Dimension der primalen Variablen	3
m	Anzahl der Nebenbedingungen	3
m_g	Anzahl der Gleichungsnebenbedingungen	3
m_v	Anzahl der verletzten Nebenbedingungen	87
M	Indexmenge der Nebenbedingungen	29
M_v	Indexmenge der verletzten Nebenbedingungen	87
$A(r, x, u)$	Indexmenge der aktiven Nebenbedingungen	30
$K(r, x, u)$	Indexmenge der inaktiven Nebenbedingungen	30
f	Zielfunktion	3
h	Nebenbedingungsfunktion	3
$\bar{h}(x)$	Vektor der aktiven Nebenbedingungsfunktionswerte	30
$\check{h}(r, x, u)$	erweiterter Vektor der aktiven Nebenbedingungsfunktionswerte	30
g	Schranken-Nebenbedingungsfunktion	83
X	Schranken	12
\underline{x}	untere Schranke für primale Variable	12
\bar{x}	obere Schranke für primale Variable	12
J	Jacobi-Matrix der Nebenbedingungsfunktionen	30
L	Untere Asymptote für die primale Variable	4
L^*	Untere Asymptote für Teilproblem in x^*	28
$L(z, \Delta, \xi)$	Untere Asymptote für TRSCP-Teilproblem	46
L_{min}	Untere Schranke für die untere Asymptote	22
U	Obere Asymptote für die primale Variable	4
U^*	Obere Asymptote für Teilproblem in x^*	28

$U(z, \Delta, \xi)$	Obere Asymptote für TRSCP-Teilproblem	46
U_{max}	Obere Schranke für die obere Asymptote	22
Δ	Trust-Region-Radius	12
Δ_i^{max}	maximaler Trust-Region-Radius im Koeffizienten i	84
Δ_{max}	maximaler Trust-Region-Radius	46
Δ_{min}^k	minimaler Trust-Region-Radius in Iteration k	46
Δ^*	Trust-Region-Radius für Teilproblem in x^*	54
T_1, T_2	Parameter zur Berechnung des Trust-Region-Radius im Programm TRSCP1.0	84
ξ	Mindestabstand	13
Ξ	Vektor mit dem Mindestabstand als Eintrag in jedem Koeffizienten	44
$I_0^+(z)$	Indexmenge der nicht-negativen Koeffizienten der partiellen Ableitung der Zielfunktion	24
$I_0^-(z)$	Indexmenge der negativen Koeffizienten der partiellen Ableitung der Zielfunktion	24
$I_j^+(z)$	Indexmenge der nicht-negativen Koeffizienten der partiellen Ableitung der Nebenbedingungsfunktion j	24
$I_j^-(z)$	Indexmenge der negativen Koeffizienten der partiellen Ableitung der Nebenbedingungsfunktion j	24
I_u	Indizes der primalen Variablen für die eine untere Schranke existiert	83
I_o	Indizes der primalen Variablen für die eine obere Schranke existiert	83
n_u	Anzahl der Indizes der primalen Variablen für die eine untere Schranke existiert	83
n_o	Anzahl der Indizes der primalen Variablen für die eine obere Schranke existiert	83
$SP(z, L, U)$	SCP-Teilproblem	24
$TRSP(z, \Delta)$	TRSCP-Teilproblem	49
$SP(z^*, L^*, U^*)$	SCP-Teilproblem in einem stationären Punkt	54
$TRSP(z^*, \Delta^*)$	TRSCP-Teilproblem in einem stationären Punkt	49
$ASP(z, \Delta, \rho)$	alternatives Teilproblem	88
ρ	Penalty-Parameter für alternatives Teilproblem	88
θ	Schlupfvariable im alternativen Teilproblem	88
$f^k(x, z^k, L^k, U^k, \tau)$	Approximation der Zielfunktion (SCP-Teilproblem)	24
$f^k(x, z^k, \Delta^k, \xi, \tau)$	Approximation der Zielfunktion (TRSCP-Teilproblem)	47
$h_j^k(x, z^k, L^k, U^k)$	Approximation der Nebenbedingungsfunktion j (SCP-Teilproblem)	24

VI

$h_j^k(x, z^k, \Delta^k, \xi)$	Approximation der Nebenbedingungsfunktion j (TRSCP-Teilproblem)	47
$\alpha_{0,0}(z^k, L^k, U^k)$	Bestandteil der SCP-Approximation	24
$\alpha_{0,0}(z^k, \Delta^k, \xi)$	Bestandteil der TRSCP-Approximation	48
$\alpha_{0,i}(x_i, z^k, L_i^k, U_i^k, \tau)$	Bestandteil der SCP-Approximation	24
$\alpha_{0,i}(x_i, z^k, \Delta^k, \xi, \tau)$	Bestandteil der TRSCP-Approximation	48
$\alpha_{j,0}(z^k, L^k, U^k)$	Bestandteil der SCP-Approximation	24
$\alpha_{j,0}(z^k, \Delta^k, \xi)$	Bestandteil der TRSCP-Approximation	48
$\alpha_{j,i}(z^k, L_i^k, U_i^k)$	Bestandteil der SCP-Approximation	24
$\alpha_{j,i}(z^k, \Delta^k, \xi)$	Bestandteil der TRSCP-Approximation	48
$R_{f^k}(x, z^k, \Delta^k, \xi, \tau)$	Nichtlineares Restglied der Taylorentwicklung der approximierten Zielfunktion	56
$R_{h_j^k}(x, z^k, \Delta^k, \xi)$	Nichtlineares Restglied der Taylorentwicklung der approximierten Nebenbedingungsfunktion j	56
$R_{h^k}(x, z^k, \Delta^k, \xi)$	Vektor, gebildet aus den nichtlinearen Restgliedern der Taylorentwicklungen der approximierten Nebenbedingungsfunktionen	59
$\tilde{h}(x, z, L, U)$	MMA-Approximation	4
X^k	Schranken im Teilproblem als Menge	24
$\Omega(z^k, L^k, U^k, \underline{x}, \bar{x})$	zulässiger Bereich (SCP-Teilproblem)	26
$\Omega(z^k, \Delta^k, \xi, \underline{x}, \bar{x})$	zulässiger Bereich (TRSCP-Teilproblem)	50
$\Phi_r \begin{pmatrix} x \\ u \end{pmatrix}$	erweiterte Lagrange-Funktion	29
$L(x, u)$	Lagrange-Funktion	85
r	Penalty-Parameter	29
P	Penalty-Parameter als Diagonalmatrix	30
λ	Parameter zur Erzeugung der Asymptoten bzw. der Trust-Region	23
μ	Parameter zur Bewertung einer Teilproblemlösung	90
ν	Parameter zur Änderung der Trust-Region	90
τ	Parameter zur Erzeugung strikt konvexer Zielfunktionsapproximationen	24
η^k	Maß des Krümmungsverhaltens	31
$\eta(z^k, y_i^k, L_i^k, u_i^k, \tau)$	Einzelmaß des Krümmungsverhaltens	31
γ	Faktor für Bestimmung eines hinreichenden Abstiegs	32
$\beta(z, y, u, v)$	Quotient	68
$\bar{\beta}$	Schranke für den Quotienten $\beta(z, y, u, v)$	68
ϑ	Faktor zur Bestimmung der Schrittweite	32
Q	Quotient aus erreichter Verbesserung und prognostizierter Verbesserung	40

κ	Faktor für Erhöhung des Penalty-Parameters	32
$P(x, \zeta)$	Penalty-Funktion im Alg. 3.1	39
ζ	Penalty-Parameter im Alg. 3.1	39
D	Matrix bei Alg. 3.1	40
E_n	Rang-n-Einheitsmatrix	41
$TP(z, \Delta, \zeta)$	Teilproblem für Alg. 3.1	40
$S(x, z, \zeta)$	Zielfunktion im Alg. 3.1	40
ϖ	Faktor im Alg. 3.1	40
ϱ	Vektor für Update der Matrix im Alg. 3.1	41
ω	Vektor für Update der Matrix im Alg. 3.1	41
\underline{d}	untere Schranke der primalen Variablen (BCP-Teilproblem)	13
\bar{d}	obere Schranke der primalen Variablen (BCP-Teilproblem)	13
b	Schranke	26
p	einwirkende Kraft	5
l	Stablänge	6
ψ	Winkel	6
φ	Winkel	6
$u(x)$	Verformungsnebenbedingung	6
\check{u}	Verformungsschranke	6
$K(x)$	Steifigkeitsmatrix	6
$K_e(x)$	Element-Steifigkeitsmatrix	6
q	Anzahl der Freiheitsgrade	9
∇	Kennzeichnung der partiellen Ableitung einer Funktion	27
F_{\max}	Schranke für die Anzahl der Funktionsauswertungen im Programm TRSCP1.0	93
Γ	Abkürzung für einen Ausdruck, der abgeschätzt werden soll	65
MS	Motorschiff	8

Inhaltsverzeichnis

Zusammenfassung	I
Abstract	II
Symbolverzeichnis	III
1 Einleitung	3
2 SCP-Verfahren	17
2.1 Übersicht	17
2.2 Problemstellung	20
2.3 Konvexe Approximationen	21
2.4 KKT-Punkte	27
2.5 Ein Algorithmus mit konvexen Approximationen	31
2.6 Globale Konvergenz von LSSCP	34
3 Trust-Region-Methoden	37
3.1 Ein Trust-Region-Modell-Algorithmus	38
3.2 Der Algorithmus TRSCP	43
3.3 Globale Konvergenz von TRSCP	53
4 Numerische Resultate	81
4.1 Implementierungsdetails	82
4.2 Vergleich von TRSCP mit LSSCP	92
4.3 Weitere Resultate von TRSCP	99
5 Fazit und Ausblick	111
Literaturverzeichnis	a
Danksagung	g

Kapitel 1

Einleitung

In der vorliegenden Arbeit wird ein neuer Ansatz zur Lösung von kontinuierlichen, nichtlinearen und restringierten Optimierungsproblemen vorgestellt. Der Lösungsansatz basiert auf einem Verfahren der sequentiellen konvexen Optimierung, der aufgrund einer Trust-Region-Stabilisierung globale Konvergenzeigenschaften besitzt. Das neue Optimierungsverfahren heißt TRSCP¹.

TRSCP gehört zu der Klasse der SCP²- bzw. Trust-Region-Verfahren und kann zur Lösung von folgendem NLP³ eingesetzt werden.

Definition 1.0.1 (Nichtlineares Optimierungsproblem)

Das Problem

$$\begin{aligned} & \text{Minimiere } f(x), \\ \text{(NLP)} \quad & \text{unter } h_j(x) = 0, \quad j = 1, \dots, m_g, \\ & h_j(x) \leq 0, \quad j = m_g + 1, \dots, m, \\ & x \in \mathbb{R}^n, \end{aligned} \tag{1.1}$$

heißt nichtlineares Optimierungsproblem (NLP).

Für die Funktionen f und h_j , $j = 1, \dots, m$, wird zweimalige stetige Differenzierbarkeit gefordert bzw. vorausgesetzt.

Bei sequentiellen Optimierungsverfahren wird iterativ, ausgehend von einer willkürlichen Schätzung für die Lösung von NLP, ein Teilproblem aufgestellt und anschließend gelöst. Die Lösung des Teilproblems sollte einfacher zu berechnen sein als die des Gesamtproblems und sollte in einem gewissen Sinn eine Verbesserung der ursprünglichen Iterierten sein. Mit dem Ergebnis des Teilproblems wird ein neues Teilproblem aufgestellt und so eine Folge von Näherungen für die Lösung des NLPs erzeugt. Das heißt, ein iteratives SCP-Verfahren erzeugt eine Folge von Vektoren,

¹steht für Trust-Region Sequential Convex Programming

²steht für Sequential Convex Programming

³steht für Non-Linear Program

die Entwicklungspunkte für die Teilprobleme sind. Die Iteration endet, wenn eine Lösung oder eine hinreichend genaue Näherung für eine Lösung von NLP gefunden wurde.

Die SCP-Verfahren haben ihren Ursprung in den achtziger Jahren des 20. Jahrhunderts. In Fleury und Braibant [20] wurde die Methode CONLIN⁴ zur Lösung von Aufgaben aus dem Bereich der mechanischen Strukturoptimierung vorgestellt. Durch konvexe Approximation der Funktionen eines NLPs werden einfach zu lösende Teilprobleme erzeugt und effizient gelöst. Die entstehende Iterationsfolge nähert eine Lösung des NLPs an.

Moderne SCP-Verfahren basieren auf MMA⁵, einem Vertreter dieser Optimierungsverfahren. Hier werden konvexe Teilprobleme erzeugt, indem die Zielfunktion f und die Ungleichungsnebenbedingungen h_j , $j = m_g + 1, \dots, m$, bezüglich reziproker Variablen linearisiert werden. Die Gleichungsnebenbedingungen h_j , $j = 1, \dots, m_g$, werden mittels einer gewöhnlichen Linearisierung in diese Teilprobleme aufgenommen.

Bei MMA-Methoden müssen zwei Schranken $L \in \mathbb{R}^n$ und $U \in \mathbb{R}^n$ gewählt werden, zwischen denen sich die Variable bewegen kann. L und U sind Asymptoten für die konvexe Approximation.

Definition 1.0.2 (MMA-Approximation)

Sei $x \in \mathbb{R}^n$ und $h : \mathbb{R}^n \rightarrow \mathbb{R}$. Weiter sei $z \in \mathbb{R}^n$, $L \in \mathbb{R}^n$ und $U \in \mathbb{R}^n$ mit $L_i < z_i < U_i$, $i = 1, \dots, n$. Dann ist die Funktion

$$\begin{aligned} \tilde{h}(x, z, L, U) := & h(z) + \sum_{\{i=1, \dots, n: \frac{\partial h(z)}{\partial x_i} \geq 0\}} \frac{\partial h(z)}{\partial x_i} (U_i - z_i)^2 \left(\frac{1}{U_i - x_i} - \frac{1}{U_i - z_i} \right) \\ & - \sum_{\{i=1, \dots, n: \frac{\partial h(z)}{\partial x_i} < 0\}} \frac{\partial h(z)}{\partial x_i} (z_i - L_i)^2 \left(\frac{1}{x_i - L_i} - \frac{1}{z_i - L_i} \right) \end{aligned} \quad (1.2)$$

eine MMA-Approximation von $h(x)$ im Entwicklungspunkt z .

Eine MMA-Approximation ist eine Näherung 1. Ordnung der zugrundeliegenden Funktion. Außerdem ist sie konvex und separabel, d.h. die zugehörige Hessematrix ist diagonal.

Ein Optimierungsverfahren heißt global konvergent, wenn es von jedem beliebigen zulässigen Startwert aus konvergiert. SCP-Verfahren benötigen eine Stabilisierungstechnik, um globale Konvergenz zu erreichen. Dies kann beispielsweise durch eine Schrittweitensteuerung erfolgen. Als Schrittweite wird der Abstand zwischen dem Entwicklungspunkt und der Teilproblemlösung bezeichnet.

In dieser Arbeit wird die Methode TRSCP vorgestellt. TRSCP ist ein SCP-Verfahren mit kombiniertem Trust-Region-/Move-Limits-Ansatz, d.h. dieses SCP-Verfahren

⁴steht für CONVex LINearization

⁵steht für Method of Moving Asymptotes

wird mittels einer Trust-Region-Strategie stabilisiert. Hierbei wird der Bereich zwischen den MMA-Asymptoten L und U als Trust-Region angesehen. Durch Steuerung der Asymptotenerzeugung im Sinne einer Trust-Region-Methode wird globale Konvergenz des Verfahrens erreicht.

Svanberg [57] hat in der Methode MMA die grundlegende Approximationsidee der SCP-Verfahren vorgestellt. Zillober [69] präsentierte das erste global konvergente SCP-Verfahren, indem ein auf der MMA-Approximation basierendes iteratives Optimierungsverfahren mit einer Line-Search-Methode kombiniert wurde. Weiter folgten in Svanberg [59] und Zillober [74] global konvergente SCP-Methoden für allgemeine, kontinuierliche und restringierte, nichtlineare Optimierungsprobleme. Diese zwei SCP-Methoden sind Standardverfahren zur Lösung von Strukturoptimierungsproblemen. In Svanberg [58] und Zillober et al. [78] wurde die Effizienz der beiden Verfahren zur Lösung von hochdimensionierten strukturmechanischen Optimierungsproblemen gezeigt. Darüber hinaus wurde das SCP-Verfahren von Zillober [74] zur Lösung von hochdimensionierten Topologieoptimierungsproblemen und Steuerungsproblemen basierend auf semilinearen, elliptischen, partiellen Differentialgleichungssystemen angewandt. Ertel [11] kombiniert die MMA-Approximation mit einer Filter-Methode zu einem neuen SCP-Verfahren. Filter sind eine weitere Möglichkeit zur Stabilisierung von sequentiellen Optimierungsverfahren. Durch einen Filter wird einerseits die Zulässigkeit von Iterationspunkten und andererseits der Wert der Zielfunktion bewertet. Filter-Methoden erzeugen bezogen auf den Wert der Zielfunktion zumeist nicht-monotone Folgen. Bei dem SCP-Filter-Verfahren aus Ertel [11] werden im Allgemeinen mehr Teilproblemlösungen als neue Entwicklungspunkte bzw. Iterationspunkte akzeptiert als beim Einsatz einer Line-Search-Methode. In Stingl et al. [56] wurde der Algorithmus PENS-SCP vorgestellt, der auf dem SCP-Verfahren aus Zillober [74] aufbaut und der Lösung von FMO⁶-Problemen mit multiplen Lastfällen dient. Mit diesem SCP-Verfahren können semidefinite Probleme gelöst werden. Für semidefinite Nebenbedingungen kommt hierbei eine speziell angepasste Approximation zur Erstellung von Teilproblemen zum Einsatz. In Ertel et al. [12, 13] werden ebenfalls FMO-Probleme mittels einer SCP-Methode gelöst. Das in Ertel et al. [13] vorgestellte SCP-Verfahren basiert auf strikter Zulässigkeit, wobei in jeder Iteration die Zulässigkeit einer gegebenen Teilmenge der Nebenbedingungen von NLP garantiert ist.

Ein Beispiel für ein nichtlineares, kontinuierliches Optimierungsproblem, das mittels eines SCP-Verfahrens effizient gelöst werden kann, ist die mathematische Formulierung folgender Aufgabe.

„Bestimme für das statisch bestimmte Stabtragwerk aus Abbildung 1.1 den minimalen Durchmesser der Stäbe, wenn die auf das Tragwerk wirkende Kraft p und das Material der Stäbe bekannt sind.“

Ein statisch bestimmtes Stabtragwerk ist eine Struktur aus geraden Stäben, die zum Teil an ihren Enden miteinander durch Gelenke verbunden sind und bei dem die

⁶steht für Free Material Optimization

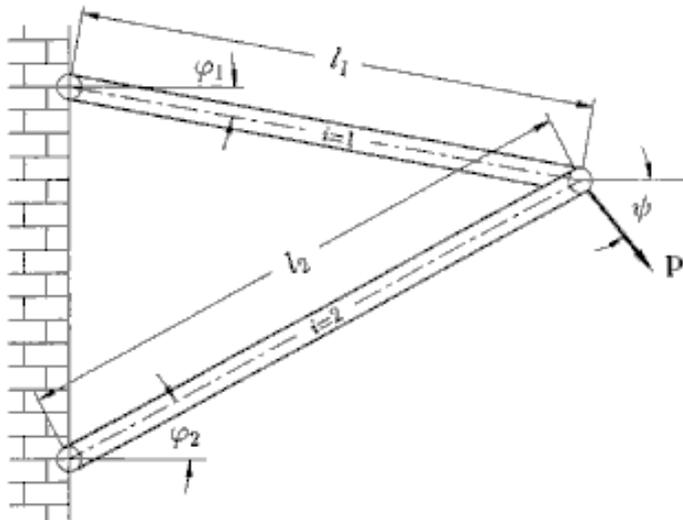


Abbildung 1.1: Statisch bestimmtes Stabtragwerk mit einwirkender Kraft p

Lager- und Stabkräfte aus den Gleichgewichtsbedingungen der Statik bestimmbar sind.

Ein einzelner Stab ist ein Bauteil mit einer wesentlich größeren Länge als Breite, das ausschließlich in seiner Längsrichtung durch Zug bzw. Druck belastet werden kann. Eine Einführung in die Theorie der statisch bestimmten Stabtragwerke bietet Krätzig et al. [29].

Optimierungsaufgaben wie in Abbildung 1.1 stammen aus dem Bereich der Mechanik und werden unter dem Begriff Auslegungsoptimierung bzw. unter dem Überbegriff Strukturoptimierung zusammengefasst. Einen Einblick in die Strukturoptimierung aus Ingenieurssicht bieten die Bücher von Harzheim [24], Bendsøe und Sigmund [3], Papalambros und Wilde [38], Haug und Arora [25], Haftka und Gürdal [22] und Vanderplaats [63].

SCP-Methoden haben sich als besonders effektiv bei der Lösung von Strukturoptimierungsaufgaben herausgestellt. Ein Blick auf das mathematische Modell für die Aufgabenstellung zu Abbildung 1.1 zeigt den Grund hierfür.

Aufgabe 1.0.3 (Two-bar-truss)

Seien die konstanten Längen $l_i > 0$ der beiden Stäbe $i := 1$ bzw. $i := 2$ gegeben und das Stabtragwerk an zwei Stellen fixiert⁷. Weiter seien die Winkel ψ , φ_i und die Kraft $p \in \mathbb{R}^2$ festgelegt. Dann wird die minimale Querschnittsfläche x_i der Stäbe

⁷Dargestellt in Abbildung 1.1 durch eine Verbindung der Stäbe mit dem Mauerwerk.

durch folgende Minimierungsaufgabe bestimmt:

$$\begin{aligned}
 \text{Minimiere} \quad & \sum_{i=1}^2 x_i, \\
 \text{unter} \quad & x_i > 0, \\
 & u_1(x) \leq \check{u}_1 > 0, \\
 & u_2(x) \geq \check{u}_2 < 0.
 \end{aligned} \tag{1.3}$$

Die vertikalen und horizontalen Verformungen $u_1(x)$ und $u_2(x)$ ergeben sich als Lösung des Gleichungssystems $K(x)u = p$, wobei p der Vektor der äußeren Kräfte ist und $K(x)$ die globale Steifigkeitsmatrix. Für die Koeffizienten von u gilt

$$\begin{aligned}
 u_1(x) &:= \frac{|p| \left[\cos \psi \left(\frac{\sin^2 \varphi_2}{x_1} + \frac{\sin^2 \varphi_1}{x_2} \right) - \sin \psi \left(\frac{\sin \varphi_2 \cos \varphi_2}{x_1} + \frac{\sin \varphi_1 \cos \varphi_1}{x_2} \right) \right]}{\sin^2(\varphi_1 - \varphi_2)}, \\
 u_2(x) &:= \frac{|p| \left[\sin \psi \left(\frac{\cos^2 \varphi_2}{x_1} + \frac{\cos^2 \varphi_1}{x_2} \right) - \sin \psi \left(\frac{\sin \varphi_2 \cos \varphi_2}{x_1} + \frac{\cos \varphi_1 \cos \varphi_1}{x_2} \right) \right]}{\sin^2(\varphi_1 - \varphi_2)}.
 \end{aligned} \tag{1.4}$$

Die globale Steifigkeitsmatrix $K(x)$ ist die Summe der Elementsteifigkeitsmatrizen $K_e(x_i)$ mit

$$K_e(x_i) := x_i \begin{pmatrix} \cos^2 \varphi_i & \sin \varphi_i \cos \varphi_i \\ \sin \varphi_i \cos \varphi_i & \sin^2 \varphi_i \end{pmatrix}. \tag{1.5}$$

Eine Ersetzung der reziproken Variablen mit $y_1 := \frac{1}{x_1}$ und $y_2 := \frac{1}{x_2}$ führt zu linearen Restriktionen, d.h., insgesamt auf ein lineares Optimierungsproblem.

In der vorliegenden Arbeit werden Fragestellungen aus dem Bereich der Strukturoptimierung behandelt, die die Auslegung und die Topologie einer Struktur betrachten. Weitere Themen, wie zum Beispiel die Wahl der Geometrie oder des Materials, werden in dieser Arbeit nicht behandelt. Für das Thema FMO, das zu semidefiniten, nichtkonvexen Optimierungsproblemen führt, wird auf Bendsøe et al. [2], Ben-Tal et al. [4], Zowe et al. [79] und Kočvara et al. [28] verwiesen.

Einfache Strukturen lassen sich mit wenigen Parametern analytisch beschreiben und Strukturoptimierungsaufgaben über diese Parameter direkt formulieren. Allgemeine mechanische Strukturen werden durch eine Idealisierung mittels einer Finiten-Elemente-Methode, d.h. über eine Diskretisierung des Problems mathematisch modellierbar. Hierbei wird die zu diskretisierende Struktur mit einem sogenannten Netz überzogen. Mit Hilfe dieses Netzes wird die Struktur in Teilkörper aufgeteilt. Die Form der Teilkörper ist so zu wählen, daß ihre Eigenschaften, wie zum Beispiel Steifigkeit, bekannt sind. So kann die Ausgangsstruktur durch ein Netz einfacher Elemente mit bekannten Eigenschaften angenähert werden.

Abbildung 1.2 zeigt ein Bild der MS „Radiance of the Seas“. Die Struktur dieses Schiffes wurde mittels des SCP-Verfahrens von Zillober [74] gewichtsreduziert. In Abbildung 1.3 ist das hierbei verwendete Finite-Elemente-Modell dargestellt.⁸



Abbildung 1.2: MS „Radiance of the Seas“

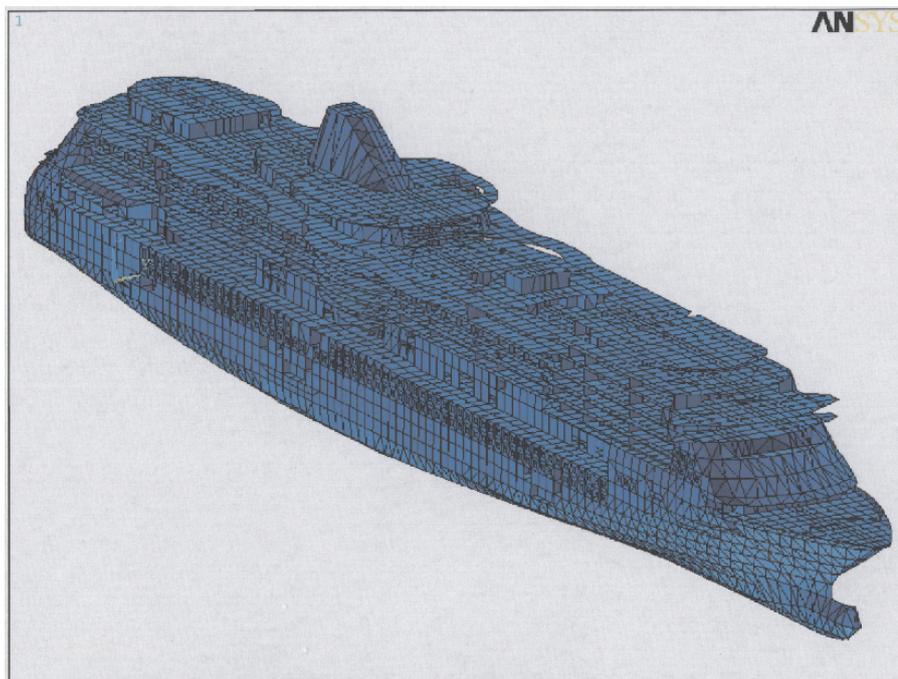


Abbildung 1.3: Finite-Elemente-Netz der „Radiance of the Seas“

Mit einer Finite-Elemente-Diskretisierung wird eine Approximation eines mathematischen Modells einer Struktur beschrieben. Jedes diskrete Element der Struktur sollte eine einfachere mathematische Beschreibung haben als das Gesamtproblem.

⁸Abbildung 1.2 und 1.3 sind aus Zillober und Vogel [72] mit freundlicher Genehmigung.

Zur Erläuterung des Einsatzes der Finiten-Elemente-Methode sei eine Aufgabe betrachtet, die eine bestimmte Steifigkeit von einer Struktur fordert. Die Steifigkeit ist eine Größe in der Mechanik, die den Widerstand einer Struktur gegen Verformung durch eine Kraft beschreibt. Die Steifigkeit einer Struktur ist von deren Material sowie ihrer Geometrie abhängig. Die Dimension der Steifigkeitsmatrix $K(x) \in \mathbb{R}^{q \times q}$ der Struktur bestimmt sich aus der Anzahl der Freiheitsgrade $q \in \mathbb{N}$ der Problemstellung. Die Struktursteifigkeitsmatrix $K(x)$ ergibt sich aus der Summe der Element-Steifigkeitsmatrizen $K_e(x_i)$, $i = 1, \dots, n$, d.h.

$$K(x) := \sum_{i=1}^n K_e(x_i). \quad (1.6)$$

Jedes der n Elemente der Finite-Elemente-Diskretisierung besitzt eine Element-Steifigkeitsmatrix $K_e(x_i) \in \mathbb{R}^{q \times q}$. Eine Element-Steifigkeitsmatrix hat für jeden Freiheitsgrad q der Modellierung eine Zeile bzw. Spalte. Beeinflussen sich zwei Freiheitsgrade in diesem Element gegenseitig, dann werden die Einträge in der Elementsteifigkeitsmatrix durch die Richtung der insgesamt wirkenden Kräfte bzw. Verschiebungen bestimmt. Weiter beeinflussen Größe, Form und Material des Elements die Element-Steifigkeitsmatrix. Als Beispiel siehe die Element-Steifigkeitsmatrizen (1.5).

Ist die Summe aller auf einen Körper einwirkenden Kräfte gleich Null, dann befindet sich dieser in einem statischen Gleichgewicht. Ist eine Struktur in einem statischen Gleichgewicht, wird über die Gleichgewichtsbeziehung

$$K(x)u = p \quad (1.7)$$

die Steifigkeitsmatrix mit dem Vektor $u \in \mathbb{R}^q$ der globalen Verschiebungen und mit dem Vektor $p \in \mathbb{R}^q$ der in Richtung der globalen Verschiebungen auf das Bauteil einwirkenden Kräfte verknüpft. Die Beziehung (1.7) ist grundlegend, um ein Strukturoptimierungsproblem aufstellen zu können.

Je feiner ein Finite-Elemente-Netz gewählt wird, desto genauer ist die mathematische Beschreibung der Struktur. Im Gegenzug wächst durch die Verfeinerung des Netzes die Dimension des mathematischen Modells.

Eine Einführung und einen Überblick über Finite-Elemente-Methoden bieten die Bücher von Bathe [1], Hahn [23] und Zienkiewicz und Taylor [67, 68].

Ein mathematisches Modell einer Struktur soll wirksam und zuverlässig sein. Hierbei wird unter wirksam verstanden, daß ein solches Modell bei einer gegebenen Genauigkeit den Aufwand zur Lösung des Problems gering hält. Ein mathematisches Modell heißt zuverlässig, wenn die zu erreichende Genauigkeit vorhergesagt werden kann.

Wird mittels einer Topologieoptimierung ein Stabtragwerk mit minimalem Gewicht gesucht, das einer bestimmten Belastungsanforderung genügt, ist die Frage zu beantworten, welche Stäbe in dieser Struktur benötigt werden. Das heißt, es ist zu entscheiden, welche Knoten des Tragwerks miteinander verbunden sind und welche nicht. Die Änderung der Verbindungen der Knoten hat einen grundlegenden Einfluß

auf die mathematische Formulierung des Problems. Dies führt zu Unstetigkeiten, aufgrund der Wechsel der Problemformulierung. Im allgemeineren Fall wird mit einer Topologieoptimierung bei einem vorgegebenen Volumen die optimale Materialverteilung für eine Struktur gesucht, d.h. es wird die benötigte Dicke des Materials in der Struktur bestimmt. Um Unstetigkeiten zu vermeiden wird für jede Stelle, die potentiell Material enthalten kann, eine minimale Dicke gefordert. Deshalb kann die gesamte Optimierung mit ein und derselben Problemformulierung durchgeführt werden. Nach der Optimierung können Elemente, die genau die minimal geforderte Dicke haben aus der Struktur entfernt werden. Erst dadurch wird die Topologie der Struktur verändert.

Abbildung 1.4 zeigt die Skizze einer Topologieoptimierungsaufgabe. Innerhalb des Rechtecks darf Material verteilt werden. Der Pfeil definiert die auf die Struktur einwirkende Kraft. Die Struktur ist an der oberen und unteren linken Ecke fixiert, d.h. die Knoten an diesen Stellen dürfen nicht verschoben werden. Die Fixierung ist durch die schwarzen Dreiecke dargestellt.

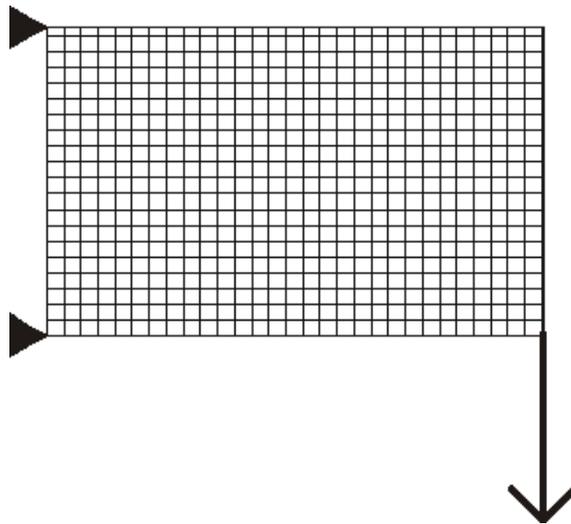


Abbildung 1.4: Skizze einer Topologieoptimierungsaufgabe

Eine mittels einer SCP-Methode bestimmte Lösung der skizzierten Topologieoptimierungsaufgabe ist in Abbildung 1.5 dargestellt.

In den siebziger Jahren des 20. Jahrhunderts wurden die ersten Verfahren zur Lösung von Optimierungsproblemen vorgestellt, deren Konvergenz mit einer Trust-Region-Methode stabilisiert wurde. Die Einführung einer Trust-Region ist eine Möglichkeit, die Konvergenzeigenschaften sequentieller Optimierungsmethoden zu unterstützen. Für einen Entwicklungspunkt wird eine Umgebung definiert, in der davon ausgegangen wird, daß das Optimierungsproblem hinreichend genau abgebildet wird. Durch eine zusätzliche, künstliche Nebenbedingung im Teilproblem wird die Menge der möglichen Lösungen auf die sogenannte Trust-Region eingeschränkt. Diese künstliche Nebenbedingung bewirkt, daß die Schrittweite, mit der sich ein solches Verfahren auf einer Abstiegsrichtung bewegt, kontrolliert wird. Von Iteration zu Iteration kann durch geeignetes Vergrößern oder Verkleinern der Trust-Region globale Konvergenz



Abbildung 1.5: Lösung der Topologieoptimierungsaufgabe aus Abbildung 1.4

des Optimierungsverfahrens erreicht werden. Ein Verfahren heißt global konvergent, wenn ein beliebiger Startpunkt gewählt werden kann und das Verfahren eine Näherungslösung des Optimierungsproblems berechnet.

Mit Powell [40] wurde ein Trust-Region-Verfahren vorgestellt und der Begriff Trust-Region eingeführt. Fletcher [17] präsentiert einen Trust-Region-Modellalgorithmus, der als Grundlage für diese Art der Optimierung gilt. Conn et al. [9] bietet vorrangig für unrestringierte Optimierungsprobleme eine intensive Einführung in die Trust-Region-Methoden. Vorwiegend SQP⁹-Verfahren werden mit Hilfe der Trust-Region-Methode stabilisiert. Für diese Trust-Region-Verfahren ist globale Konvergenz und im Allgemeinen lokale superlineare bis quadratische Konvergenz nachweisbar. Trust-Region-Methoden für unrestringierte Probleme sind ausführlich in z.B. Morè [34], Powell [41, 43] und Sorensen [55] diskutiert. Die Kombination von Trust-Region-Technik und Line-Search-Methode wird in Nocedal und Yuan [36] untersucht. Hier wird eine Line-Search zurück zur Näherungslösung aus der vorhergehenden Iteration durchgeführt, sollte die Prognose für die Lösung des aktuellen Teilproblems über die erwartete Verbesserung des Zielfunktionswerts schlecht sein. Ziel ist es, einen Entwicklungspunkt für ein Teilproblem zu bestimmen, für dessen Lösung diese Prognose hinreichend gut ausfällt. Es existiert umfangreiche Literatur über Trust-Region-Methoden für restringierte nichtlineare Optimierungsprobleme, z.B. mit Gleichungsnebenbedingungen in Dennis et al. [10], Byrd et al. [6], Celis et al. [7], Powell und Yuan [44] und für allgemein restringierte nichtlineare Optimierungsprobleme z.B. in Yuan [65]. Um lokal quadratische Konvergenz zu erhalten, bedarf es Ableitungen 2. Ordnung. Benötigen frühere Algorithmen die Auswertung einer Penalty-Funktion, kommen neuere Trust-Region-Verfahren ohne dieses Hilfsmittel aus. Fletcher und Leyffer [18] präsentieren ein Trust-Region-Verfahren ohne Strafkostenfunktion, das mit einem Filter die Aufgabe einer Penalty-Parameter-Merit-Funktion ersetzt. Die globale Konvergenz dieser Methode wird in Fletcher et

⁹steht für Sequential Quadratic Programming

al. [19] bewiesen. In Ulbrich und Ulbrich [62] wird ein nichtmonotones, ebenfalls Penalty-Funktion-freies Trust-Region-Verfahren ohne den Filter von Fletcher und Leyffer [18] vorgestellt. Diese Methode ist global konvergent und konvergiert lokal quadratisch. Das Verfahren von Ulbrich und Ulbrich [62] benötigt hierfür keinen Korrekturschritt basierend auf Informationen 2. Ordnung.

In Ni [35] wird ein unrestringiertes SCP-Verfahren vorgestellt, das durch das Einführen einer Trust-Region stabilisiert wird. Zusätzlich zu den bereits vorhandenen MMA-Asymptoten $L \in \mathbb{R}^n$ und $U \in \mathbb{R}^n$ wird hier eine Trust-Region mit dem Trust-Region-Radius $\Delta \in \mathbb{R}$ eingeführt. Dies unterscheidet sich grundlegend von dem in dieser Arbeit vorgestellten Ansatz, bei dem der Bereich zwischen den Asymptoten L und U als Trust-Region angesehen bzw. genutzt wird. Weiter können bei TRSCP Gleichungs- und Ungleichungsnebenbedingungen in der Problemstellung betrachtet werden. In Ni [35] wird ein Trust-Region-Algorithmus für Optimierungsprobleme ausschließlich mit Schranken als Nebenbedingungen aufgestellt. Dieser Ansatz soll im Folgenden kurz ausgeführt werden.

Definition 1.0.4

Das Problem

$$\begin{aligned} \text{(BCP)} \quad & \text{Minimiere } f(x), \\ & \text{unter } x \in X \end{aligned} \tag{1.8}$$

mit

$$X := \{x \in \mathbb{R}^n : \underline{x}_i \leq x_i \leq \bar{x}_i, i = 1, \dots, n\}, \tag{1.9}$$

heißt box-constrained¹⁰ Optimierungsproblem (BCP).

Ein box-constrained Optimierungsproblem ist ein NLP mit $m_g := 0$ und $m := 2n$, wobei die Nebenbedingungen speziell durch Schranken gewählt sind.

Ausgehend von einer Näherungslösung von BCP soll der Algorithmus aus Ni [35] eine Folge von Iterationen erzeugen, die gegen eine Lösung von BCP konvergiert. Unter einer Lösung wird hierbei ein Element aus X verstanden, von dem aus keine Abstiegsrichtung zu einem anderen Element aus X gefunden werden kann.

Für eine Näherung $z \in \mathbb{R}^n$ einer Lösung von BCP wird ein konvexes Teilproblem aufgestellt und anschließend gelöst. Die Lösung dieses Teilproblems ist wiederum eine im Regelfall bessere Näherungslösung von BCP. Der Algorithmus aus Ni [35] erzeugt eine Folge von Iterationspunkten, die gegen eine Lösung konvergiert. Diese Konvergenz wird durch eine Trust-Region-Strategie unterstützt, d.h. von Iteration zu Iteration wird der Trust-Region-Radius Δ angepaßt. Um die Art dieser Anpassung, d.h. Verkleinerung oder Vergrößerung der Trust-Region, und deren Güte zu bestimmen, wird für die aktuelle Lösung des Teilproblems eine Prognose erstellt, inwiefern sich der Wert der Zielfunktion von BCP mit diesem Vektor verbessert. Diese Prognose wird mit der tatsächlichen Änderung des Zielfunktionswerts verglichen. Basierend auf dem Ergebnis dieses Vergleichs wird der Trust-Region-Radius angepaßt.

¹⁰steht für „Schachtelnebenbedingungen“, d.h. für untere und obere Schranken

Im konvexen Teilproblem der Methode aus Ni [35] wird die Menge an zulässigen Lösungen entweder durch die Trust-Region oder durch die Schranken aus BCP eingeschränkt. Um die Trust-Region mit den Box-Constraints abzustimmen, wird

$$\begin{aligned} \underline{d}_i &:= \max \{-\Delta, \underline{x}_i - z_i\}, \\ \bar{d}_i &:= \min \{\Delta, \bar{x}_i - z_i\} \end{aligned} \quad (1.10)$$

für $i = 1, \dots, n$ festgelegt. Das heißt die Nebenbedingungen von BCP werden für das Teilproblem in

$$\underline{d}_i \leq x \leq \bar{d}_i, \quad \forall i = 1, \dots, n, \quad (1.11)$$

transformiert, siehe Abbildung 1.6.

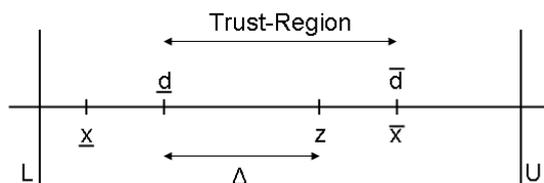


Abbildung 1.6: Trust-Region Definition in Ni [35]

Sei $z \in \mathbb{R}^n$ der Entwicklungspunkt eines Teilproblems von BCP. Dann fordert Ni [35] für jeden Koeffizienten $i = 1, \dots, n$ der Asymptoten $L, U \in \mathbb{R}^n$,

$$\begin{aligned} L_i &< z_i - \Delta, \\ U_i &> z_i + \Delta. \end{aligned} \quad (1.12)$$

Das heißt, die Trust-Region muß immer kleiner sein als der Abstand zwischen dem Entwicklungspunkt und den Asymptoten.

Im Unterschied zu der SCP-Methode aus Ni [35] kann mit TRSCP nicht nur ein BCP, sondern ein allgemeines NLP gelöst werden. TRSCP wird ebenfalls durch eine Trust-Region-Strategie stabilisiert und erreicht globale Konvergenz.

Bei TRSCP wird ein fester Zusammenhang zwischen MMA-Asymptoten und Trust-Region-Radius festgelegt, siehe Abbildung 3.2. Die Asymptoten werden aus einem Entwicklungspunkt $z \in \mathbb{R}^n$ für ein Teilproblem und einem gegebenen Trust-Region-Radius $\Delta \in \mathbb{R}^n$, $\Delta_i > 0$, $i = 1, \dots, n$, berechnet, d.h.

$$\begin{aligned} L(z, \Delta, \Xi) &:= z - \Delta - \Xi, \\ U(z, \Delta, \Xi) &:= z + \Delta + \Xi, \end{aligned} \quad (1.13)$$

mit $\Xi := (\xi, \dots, \xi)$, $\xi > 0$ konstant. Die Konstante ξ garantiert hierbei, daß die Asymptoten nicht mit dem Entwicklungspunkt $z \in \mathbb{R}^n$ zusammenfallen. Würden Asymptoten und Entwicklungspunkt zusammenfallen, gäbe es neben anderen Problemen keine Trust-Region.

Im Unterschied zu TRSCP besteht bei Ni [35] eine gewisse Freiheit, die Asymptoten zu wählen. TRSCP wird durch die Festlegung der Asymptotenerzeugung in keiner

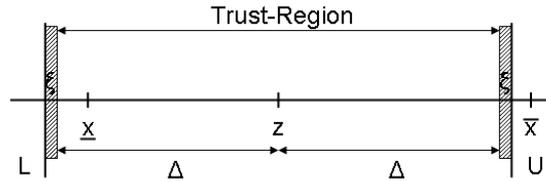


Abbildung 1.7: Trust-Region Definition von TRSCP für ein X definiert durch 1.9

Weise bedeutend eingeschränkt. Die Forderung (1.12) wird durch den in dieser Arbeit vorgestellten Ansatz mit (1.13) automatisch erfüllt.

Für die Formulierung der Teilprobleme von TRSCP findet die MMA-Approximation aus Zillober [74] Anwendung. Durch Ausnutzen von (1.13) werden die Asymptoten aus der Darstellung eliminiert und ein SCP-Teilproblem ohne explizite Asymptoten definiert. Der Trust-Region-Radius wird direkt in das Teilproblem aufgenommen und dadurch die Formulierung durch Ausnutzen der symmetrischen Definition von (1.13) vereinfacht.

Liegt in einer Iteration die Lösung eines Teilproblems einer Trust-Region-Methode auf dem Rand des Vertrauensbereichs, so spricht man von einer aktiven Trust-Region. Diese Situation kann ein Problem für das Trust-Region-Verfahren sein, falls eine erweiterte Lagrange-Funktion verwendet wird. Die Konvergenz dieses Verfahrens kann durch den Anteil des Lagrange-Multiplikators für die an sich künstliche Schranke gestört werden. Die Konvergenz des in dieser Arbeit vorgestellten Algorithmus wird allerdings nicht gestört.

In einem klassischen Trust-Region-Verfahren ist der Trust-Region-Radius eindimensional und es wird über eine eindimensionale Bedingung festgelegt, wie die Trust-Region verändert wird. Das heißt, der Trust-Region-Radius wird für jeden Koeffizienten gleich angepaßt, es wird also in einer Iteration für alle Koeffizienten der Variablen der Trust-Region-Radius gleichermaßen verändert. Mit der Strategie der Asymptotenerzeugung aus Svanberg [60] wird jeder Koeffizient der Asymptoten individuell angepaßt. In dieser Arbeit wird eine Kombination aus Trust-Region-Strategie und der Strategie der Asymptotenerzeugung aus Svanberg [60] vorgestellt, die es ohne zusätzlichen Aufwand ermöglicht, jeden Koeffizienten eines mehrdimensionalen Trust-Region-Radius individuell geeignet anzupassen. Hieraus folgt, daß in einer Iteration ein Koeffizient des Trust-Region-Radius größer, ein anderer aber kleiner werden kann.

Der Inhalt dieser Arbeit gliedert sich im Weiteren wie folgt:

Im zweiten Kapitel werden SCP-Verfahren eingeführt. Das SCP-Verfahren aus Zillober [74] und die zugehörige MMA-Approximation werden vorgestellt. Diese Approximation wird später für TRSCP geeignet angepaßt und kommt in dem neuen Verfahren zum Einsatz. Weiter werden grundlegende Begriffe wie z.B. stationäre Punkte und erweiterte Lagrange-Funktion definiert. Zum Abschluß dieses Kapitels wird der Beweis der globalen Konvergenz der SCP-Methode aus Zillober [74] skizziert. Dieser Beweisweg dient als Basis für den Beweis der globalen Konvergenz des

Verfahrens TRSCP.

Das dritte Kapitel gliedert sich in drei Teile. Zuerst wird ein Trust-Region-Algorithmus aus Yuan [65] vorgestellt und ebenfalls der Beweis der globalen Konvergenz dieser Methode skizziert. Hierbei werden Grundbegriffe der Trust-Region-Methoden eingeführt. Der zweite Teil enthält das SCP-Verfahren mit kombiniertem Trust-Region-/Move-Limits-Ansatz. Die für Trust-Region-Stabilisierung angepaßte MMA-Approximation aus Zillober [74] wird vorgestellt. Der Algorithmus Trust-Region Sequential Convex Programming (TRSCP) wird aufgestellt. Im letzten Teil dieses Kapitels wird bewiesen, daß diese SCP-Methode mit Trust-Region-Stabilisierung global konvergiert.

Das vierte Kapitel beschreibt, wie der Algorithmus TRSCP im Programm TRSCP1.0 umgesetzt wurde. Diese Umsetzung basiert auf dem Programm SCIP3.0 von Zillober [76, 77]. Verschiedene Erweiterungen für diese Trust-Region-SCP-Methode werden vorgestellt, unter anderem eine Kombination der MMA-Strategie zur Asymptotenerzeugung mit einer Trust-Region-Strategie. Den Abschluß dieses Kapitels und dieser Arbeit bilden numerische Resultate von TRSCP1.0.

Kapitel 2

SCP-Verfahren

In diesem Kapitel wird in die Theorie der sequentiellen konvexen Optimierung eingeführt. Methoden dieser Klasse von Optimierungsverfahren dienen der Lösung von kontinuierlichen, nichtlinearen und restringierten Problemen. Es wird ein Lösungsansatz vorgestellt, der mit Hilfe von Teilproblemen arbeitet, die mittels konvexer Approximation erzeugt werden. Methoden dieser Art werden unter dem Begriff SCP¹-Verfahren zusammengefaßt. Der in dieser Arbeit im Abschnitt 3.2 vorgestellte neue Algorithmus TRSCP² gehört zu dieser Klasse von Verfahren. Neben einer Einführung in die Theorie der Optimierung mittels konvexer Approximation werden grundlegende Definitionen für diese Arbeit getroffen. Unter anderem werden KKT³-Punkte vorgestellt. Die Optimierungsverfahren in dieser Arbeit liefern im Normalfall einen KKT-Punkt als Ergebnis. Als Abschluß dieses Kapitels wird der SCP-Algorithmus aus Zillober [74] vorgestellt und der Beweis der globalen Konvergenz dieses Verfahrens skizziert. Das konvexe Teilproblem aus Zillober [74] ist Grundlage für TRSCP. Die globale Konvergenzeigenschaft von TRSCP wird analog der in diesem Kapitel vorgestellten Beweisskizze nachgewiesen.

Die wichtigsten Arbeiten zum Thema SCP-Verfahren sind Fleury und Braibant [20], Svanberg [57, 59] und Zillober [69, 74].

2.1 Übersicht

SCP-Verfahren sind iterative Methoden zur Lösung von nichtlinearen, kontinuierlichen und restringierten Optimierungsproblemen.

¹steht für Sequential Convex Programming

²steht für Trust-Region Sequential Convex Programming

³steht für Karush-Kuhn-Tucker

Definition 2.1.1 (Nichtlineares Optimierungsproblem)

Das Problem

$$\begin{aligned}
 & \text{Minimiere } f(x), \\
 \text{(NLP)} \quad & \text{unter } h_j(x) = 0, \quad j = 1, \dots, m_g, \\
 & h_j(x) \leq 0, \quad j = m_g + 1, \dots, m, \\
 & x \in \mathbb{R}^n,
 \end{aligned} \tag{2.1}$$

heißt nichtlineares Optimierungsproblem (NLP).

Um ein NLP mit einem SCP-Verfahren lösen zu können, muß das Optimierungsproblem eine Stetigkeits- und Differenzierbarkeitsanforderung erfüllen.

Voraussetzung 2.1

Die Funktionen $f : \mathbb{R}^n \rightarrow \mathbb{R}$ und $h_j : \mathbb{R}^n \rightarrow \mathbb{R}$, $j = 1, \dots, m$, von NLP sind zweimal stetig differenzierbar.

In Zillober [74] ist ein Teil der Ungleichungsnebenbedingungen mit Schranken für die Variable gesetzt (analog 1.9). Durch diese Schranken ist der zulässige Bereich des Optimierungsproblems sicher eine kompakte Menge. Solche Schranken sind üblich in Strukturoptimierungsproblemen. Diese Schrankenbedingungen werden in den Beweisen zum Nachweis der globalen Konvergenzeigenschaft anders behandelt als die restlichen (siehe z.B. Zillober [74] oder Lehmann [30]), werden aber in dieser Arbeit nicht näher betrachtet, um die Aussagen und deren Beweise übersichtlicher zu gestalten. Die Kompaktheit des zulässigen Bereichs wird in dieser Arbeit nur an wenigen Stellen benötigt, bei denen dann explizit darauf eingegangen wird.

Eine unbeschränkte Folge von Iterationspunkten soll in dieser Arbeit nicht betrachtet werden. Dies würde unter anderem bei der Lösung zu numerischen Problemen führen.

Voraussetzung 2.2

Für jede Folge $\{z^k\}_{k=0,1,2,\dots} \subset \mathbb{R}^n$ existiert $0 < b < \infty$ mit $|z_i^k| \leq b$, $i = 1, \dots, n$ und $k \in \mathbb{N}_0$.

Die Voraussetzung ist erfüllt, sobald Schranken analog (1.9) für das Optimierungsproblem existieren. Da dies für TRSCP nicht allgemein gilt, wird Voraussetzung 2.2 gefordert.

SCP-Verfahren sind iterativ, d.h. in jeder Iteration wird durch konvexe Approximation für einen Vektor $z \in \mathbb{R}^n$ aus NLP ein Teilproblem erstellt und gelöst. Dafür werden $f(z)$ und $h_j(z)$, $j = 1, \dots, m$, berechnet und mit diesen Ergebnissen konvexe Funktionen aufgestellt, die in einer Umgebung von z die Funktionen von NLP approximieren. Diese Approximationen erster Ordnung definieren ein konvexes Optimierungsproblem, das sogenannte Teilproblem dieser Iteration. Der Vektor z heißt hierbei Entwicklungspunkt. Mit der Lösung des Teilproblems wird der Entwicklungspunkt der nächsten Iteration bestimmt. Die Entwicklungspunkte sind Näherungen

für eine später noch zu charakterisierende lokale Lösung von NLP. Die Iteration endet entweder mit einer lokalen Lösung von NLP oder eine unendliche Folge von Entwicklungspunkten wird erzeugt. In diesem Fall besitzt diese Folge mindestens einen Häufungspunkt und mindestens einer der Häufungspunkte dieser Folge ist eine lokale Lösung von NLP.

Im Unterschied zu SCP-Verfahren, die auf einer konvexen Approximation von NLP basieren, werden bei den häufig verwendeten SQP⁴-Verfahren quadratische Teilprobleme erzeugt, die mit Hilfe der Lagrange-Funktion des Optimierungsproblems aufgestellt werden.

Mitte der achtziger Jahre wurde von Fleury und Braibant [20] der Algorithmus CONLIN⁵ vorgestellt, der zu den SCP-Verfahren gehört. Dieser ist speziell für Strukturoptimierungsprobleme entwickelt worden. Das Verfahren konvergiert allerdings in vielen Fällen nicht. Eine allgemeinere Form dieses Algorithmus mit der Bezeichnung MMA⁶ wurde 1987 von Svanberg [57] veröffentlicht. MMA ist in der Praxis stabiler als sein Vorgänger, aber auch hier ist globale Konvergenz nur in speziellen und für die Praxis unrealistischen Fällen beweisbar. Svanberg [60] und Zillober [74] präsentierten in den neunziger Jahren zwei erweiterte MMA-Algorithmen, für die globale Konvergenz bewiesen wurde.

Das Teilproblem bei MMA-Methoden entsteht durch eine lokale konvexe Approximation erster Ordnung in einem Entwicklungspunkt $z \in \mathbb{R}^n$. Hierfür werden zwei Schranken $L \in \mathbb{R}^n$ und $U \in \mathbb{R}^n$ gewählt, so daß $L_i < z_i < U_i$ für $i = 1, \dots, n$ gilt. Um das Teilproblem zu formulieren werden die Zielfunktion f und die Ungleichungsnebenbedingungen h_j , $j = m_g + 1, \dots, m$, unter Verwendung der Terme

$$\frac{1}{x_i - L_i} \quad \text{und} \quad \frac{1}{U_i - x_i}$$

für alle $i = 1, \dots, n$ linearisiert. L und U sind Asymptoten für diese Approximationen. Die Anpassung der Asymptoten in jeder Iteration gibt MMA den Namen.

Um die globale Konvergenzeigenschaft der Verfahren beweisen zu können, beschritten Svanberg und Zillober zwei unterschiedliche Wege.

In Svanberg [58, 59, 60] werden erweiterte Versionen des MMA-Algorithmus aus Svanberg [57] vorgestellt. Durch eine Erweiterung wird sichergestellt, daß der zulässige Bereich eines Teilproblems eine Teilmenge aller zulässigen Lösungen, also des zulässigen Bereichs von NLP ist. Der zulässige Bereich ist die Menge von Vektoren, für die alle Nebenbedingungen des Problems erfüllt werden. Mit einer anderen Erweiterung wird erreicht, daß lokal für einen Vektor der Wert der Approximation größer oder gleich dem Funktionswert der Funktion von NLP ist. Diese zwei Eigenschaften zusammen mit einer speziellen Strategie der Asymptotenerzeugung liefern die globale Konvergenzeigenschaft für diese MMA-Verfahren, d.h. eine durch diese

⁴steht für Sequential Quadratic Programming

⁵steht für CONvex LINearization

⁶steht für Method of Moving Asymptotes

Verfahren erzeugte Folge von Entwicklungspunkten endet mit einer lokalen Lösung von NLP oder konvergiert gegen eine solche.

Um die Asymptoten zu bestimmen, wird beim MMA-Verfahren aus Svanberg [59] in jedem Koeffizienten des Lösungsvektors des Teilproblems die Richtung betrachtet, in die sich dieser Koeffizient im Lauf der Iteration entwickelt hat. Ist diese Richtung bei den letzten zwei Iterationen gleich, dann wird der Abstand zwischen Asymptote und Entwicklungspunkt vergrößert. Anderenfalls wird er verkleinert.

Ein Verfahren basierend auf Approximationen, die den tatsächlichen Wert der Funktionen überschätzen, wird konservativ genannt. Der Begriff lokale Lösung wird im Abschnitt 2.4 charakterisiert. Für einen aktuelleren Überblick über Svanbergs global konvergenten MMA-Algorithmus, den er in die Gruppe der CCSA⁷-Methoden einordnet, wird auf Svanberg [61] verwiesen.

Bei Zillober [69, 74] wird die globale Konvergenzeigenschaft des SCP-Verfahrens durch das Hinzufügen einer Line-Search-Methode zum MMA-Verfahren aus Svanberg [57] erreicht. Dieser Ansatz ist analog zu einigen SQP-Verfahren, wie zum Beispiel in Schittkowski [46]. Hieraus folgt, daß in Zillober [69, 74] die globale Konvergenzeigenschaft unabhängig von der Asymptotenerzeugung ist. Für die Verfahren aus Zillober [69, 74] müssen nur sehr schwache Voraussetzungen von den Asymptoten erfüllt werden.

In diesem Kapitel werden Grundlagen für das neue im Kapitel 3 vorgestellte Optimierungsverfahren TRSCP definiert. Weiter wird das SCP-Verfahren von Zillober [74] vorgestellt und Teile des globalen Konvergenzbeweises skizziert, soweit sie für die Ausführungen im nächsten Kapitel relevant sind. Im Gegensatz zu Svanberg [59] wird bei Zillober [74] nur eine allgemein gehaltene Forderung an die Asymptotenerzeugung gestellt und nicht eine explizite Berechnungsvorschrift vorgegeben. Deshalb ist das SCP-Verfahren von Zillober [74] und seine Beweisidee besser als Grundlage für diese Arbeit geeignet als das Verfahren von Svanberg [59]. Bei TRSCP ist eine spezielle Asymptotenerzeugung ein wesentlicher Bestandteil. Eine Implementierung des SCP-Algorithmus von Zillober [74] ist das Programm SCPIP3.0 [76, 77].

2.2 Problemstellung

Bei bestimmten Aufgaben der Strukturoptimierung haben SCP-Verfahren einen besonderen Vorteil. Werden Problemstellungen für statisch bestimmte Strukturen in ihren inversen Variablen betrachtet, ergeben sich lineare Probleme. Die MMA-Approximation bedient sich genau dieser Strategie, was in diesem Fall zu einer Vereinfachung der Teilprobleme führt. In dieser Arbeit wird allerdings nicht auf spezielle Aufgaben der Strukturmechanik eingegangen, sondern ein allgemeines nichtlineares Optimierungsproblem betrachtet. Solange es nicht explizit anders erwähnt ist, ist im weiteren Verlauf dieser Arbeit stets NLP (Definition 2.1.1) die Basis für die Aufgabenstellung.

⁷steht für Conservative Convex Seperable Approximation

Jede Voraussetzung oder zusätzliche Forderung einer Methode an die Problemstellung schränkt den Anwendungsbereich ein. Voraussetzung 2.1 ist ausreichend für diese Arbeit. Die Stetigkeits- und Differenzierbarkeitsforderung ist Grundlage für diese Form der gradientenbasierten Optimierung. In Zillober [74] wird in der Problemstellung $x \in X$ mit X analog (1.9) gefordert. Dies wird benötigt, um ein garantiert beschränktes Problem zu betrachten. Es wird jeweils eine untere und eine obere Schranke für jeden Koeffizienten von $x \in \mathbb{R}^n$ festgelegt bzw. gefordert. Für praktische Anwendungen macht die Forderung $x \in X$ analog (1.9) keine Schwierigkeiten, da einerseits Schranken in Strukturoptimierungsproblemen vorkommen und andererseits solche Schranken mit hinreichend großen Werten einfach angenommen werden können. Um die Notation in dieser Arbeit einfach zu halten, wird Voraussetzung 2.2 gefordert. Dadurch kann auf Schranken für die Variable verzichtet werden.

2.3 Konvexe Approximationen

Der Hauptunterschied zwischen SCP-Verfahren und den häufig benutzten sequentiellen, quadratischen Optimierungsverfahren (SQP-Verfahren) ist die Approximation, mit der ein Teilproblem erzeugt wird. SQP-Verfahren erzeugen mit Hilfe der Lagrange-Funktion quadratische Teilprobleme, d.h. die Zielfunktion ist quadratisch und die Nebenbedingungsfunktionen werden linearisiert. Bei SCP-Verfahren werden durch eine inverse Approximation konvexe, separable Teilprobleme aufgestellt.

Die konvexe Approximation aus Svanberg [57] ist die Grundlage für die Teilprobleme der SCP-Verfahren in dieser Arbeit. Sie basiert auf Funktionen, die durch Linearisierung der Zielfunktion f und der Ungleichungsnebenbedingungen h_j , $j = m_g + 1, \dots, m$, für einen Entwicklungspunkt $z \in \mathbb{R}^n$ bezüglich der reziproken Terme

$$\frac{1}{x_i - L_i} \quad \text{und} \quad \frac{1}{U_i - x_i}, \quad (2.2)$$

$i = 1, \dots, n$, gebildet werden. Die Schranken $L \in \mathbb{R}^n$ und $U \in \mathbb{R}^n$ sind hierbei so zu wählen, daß für jeden Koeffizienten i eines Entwicklungspunkts $z \in \mathbb{R}^n$

$$L_i < z_i < U_i, \quad (2.3)$$

gilt. Dadurch folgt insgesamt, daß L eine untere und U eine obere Asymptote für diese Approximationen ist.

Die Gleichungsnebenbedingungen werden durch eine gewöhnliche Linearisierung in das Teilproblem aufgenommen. Es wird im Weiteren

$$\text{ohne Beschränkung der Allgemeinheit } m_g := 0 \quad (2.4)$$

immer dann angenommen, wenn für eine einfachere Darstellung auf Gleichungsrestriktionen verzichtet werden kann.

Definition 2.3.1

$$\begin{array}{ll}
 \text{(OP)} & \begin{array}{l}
 \text{Minimiere } f(x), \\
 \text{unter } h_j(x) \leq 0, \quad j = 1, \dots, m, \\
 x \in \mathbb{R}^n.
 \end{array}
 \end{array} \tag{2.5}$$

Ein OP ist ein NLP mit $m_g := 0$.

Beim SCP-Verfahren in Zillober [74] wird die konvexe Approximation aus Svanberg [57] erweitert. Notwendig für den Beweis der globalen Konvergenz dieses Verfahrens ist eine streng konvexe Approximation der Zielfunktion f .

Um die globale Konvergenzeigenschaft dieser SCP-Methode sicherzustellen, müssen die Asymptoten bestimmte Bedingungen erfüllen. Benötigt werden Asymptoten, die der folgenden Anforderung genügen.

Definition 2.3.2 (Folge zulässiger Asymptoten)

Sei $\{z^k\}_{k=0,1,2,\dots}$ eine Folge von Entwicklungspunkten. Eine Folge von Asymptoten $\{L^k, U^k\}_{k=0,1,2,\dots}$ heißt zulässig, falls für vorgegebene Konstanten $\xi > 0$, $-\infty < L_{min} < U_{max} < \infty$,

$$\begin{array}{ll}
 L_i^k & \leq z_i^k - \xi, \\
 U_i^k & \geq z_i^k + \xi,
 \end{array} \tag{2.6}$$

und

$$\begin{array}{ll}
 L_i^k & \geq L_{min}, \\
 U_i^k & \leq U_{max},
 \end{array} \tag{2.7}$$

für $i = 1, \dots, n$ und $k \geq 0$ erfüllt sind.

Seien $z, L, U \in \mathbb{R}^n$ und $\xi > 0$. Speziell heißen L, U für z zulässige Asymptoten, wenn (2.6) und (2.7) für die Vektoren z, L und U erfüllt sind. Wenn der Bezug klar ist kann im Weiteren von „zulässigen Asymptoten“ gesprochen werden, ohne den zugehörigen Entwicklungspunkt zu nennen.

Die Definition 2.3.2 fordert einen konstanten Mindestabstand $\xi > 0$ zwischen den Asymptoten und dem Entwicklungspunkt. Wird eine Funktion in $z \in \mathbb{R}^n$ für zulässige Asymptoten $L \in \mathbb{R}^n$ und $U \in \mathbb{R}^n$ mit den reziproken Variablen

$$\frac{1}{x_i - L_i} \quad \text{oder} \quad \frac{1}{U_i - x_i} \tag{2.8}$$

für $i = 1, \dots, n$ approximiert, dann verhindert der konstante Mindestabstand ξ ein unbeschränktes Ansteigen der partiellen Ableitung der Approximation.

Die Schranken L_{min}, U_{max} mit

$$-\infty < L_{min} \leq L_i < U_i \leq U_{max} < \infty, \tag{2.9}$$

$i = 1, \dots, n$, bewirken, daß sich die Approximation in jedem Fall von einer gewöhnlichen Linearisierung unterscheidet. Ausführungen hierzu finden sich in Zillober [69].

Die Asymptotenwahl wird in den Arbeiten von Svanberg [59, 60], Zillober [73] und Schenk [45] näher behandelt. Bei Svanberg [59, 60] ist eine spezielle Asymptotenwahl notwendig, um globale Konvergenz beweisen zu können. Bei Zillober [73] und Schenk [45] wird die Auswirkung der Asymptotenwahl auf die Konvergenzgeschwindigkeit betrachtet. Die Wahl der Asymptoten spielt in Abschnitt 3.2 eine besondere Rolle und wird dort benutzt, um ein Trust-Region SCP-Verfahren zu entwickeln.

Zillober [74] verwendet in seinem Programm SCPIP3.0 [76, 77] folgenden Algorithmus zur Asymptotenerzeugung.

Algorithmus 2.3 (Asymptotenerzeugung)

Sei OP , eine Folge $\{z^k\}_{k=0,1,2,\dots} \subset \mathbb{R}^n$ von Entwicklungspunkten und Schranken $\underline{x}_i < \bar{x}_i$, $i = 1, \dots, n$, analog (1.9) gegeben. Ferner seien $0 < \lambda_1 < 1$, $0 < \lambda_2 < 1$ und $1 < \lambda_3$ Konstanten. Für $i = 1, \dots, n$ berechne:

für $k = 0, 1$

$$\begin{aligned} L_i^k &:= z_i^k - \lambda_1 (\bar{x}_i - \underline{x}_i), \\ U_i^k &:= z_i^k + \lambda_1 (\bar{x}_i - \underline{x}_i), \end{aligned}$$

und für $k = 2, 3, 4, \dots$

falls $\text{sign}(z_i^k - z_i^{k-1}) \neq \text{sign}(z_i^{k-1} - z_i^{k-2})$

$$\begin{aligned} L_i^k &:= z_i^k - \lambda_2 (z_i^{k-1} - L_i^{k-1}), \\ U_i^k &:= z_i^k + \lambda_2 (U_i^{k-1} - z_i^{k-1}), \end{aligned}$$

sonst

$$\begin{aligned} L_i^k &:= z_i^k - \lambda_3 (z_i^{k-1} - L_i^{k-1}), \\ U_i^k &:= z_i^k + \lambda_3 (U_i^{k-1} - z_i^{k-1}). \end{aligned}$$

Die Parameter für diesen Algorithmus werden im Allgemeinen mit $\lambda_1 := 0.5$, $\lambda_2 := 0.7$ und $\lambda_3 := 1.15$ gesetzt, siehe z.B. Zillober [74], Svanberg [59]. Weitere Möglichkeiten der Asymptotenerzeugung, die im Programm SCPIP3.0 [76, 77] implementiert wurden, werden in dieser Arbeit nicht betrachtet. Sind durch die Problemstellung keine Schranken gegeben, dann kann für die ersten zwei Schritte des Algorithmus eine positive, für die Problemstellung geeignete Konstante gewählt werden.

Für theoretische Betrachtungen des SCP-Verfahrens von Zillober [74] ist die Asymptotenerzeugung nach Algorithmus 2.3 nicht geeignet, da hierdurch Asymptotenfolgen von nach Definition 2.3.2 unzulässigen Asymptoten gebildet werden können.

Um das konvexe Teilproblem des SCP-Verfahrens von Zillober [74] aufstellen zu können, müssen die Vorzeichen der partiellen Ableitungen der Zielfunktion und Ungleichungsnebenbedingungsfunktionen

$$\frac{\partial f(x)}{\partial x_i} \quad \text{und} \quad \frac{\partial h_j(x)}{\partial x_i}, \quad j = 1, \dots, m,$$

für $i = 1, \dots, n$ betrachtet werden. Für einen Entwicklungspunkt $z \in \mathbb{R}^n$ werden die folgenden Indextmengen für die Zielfunktion

$$\begin{aligned} I_0^+(z) &:= \left\{ i \in \{1, \dots, n\} : \frac{\partial f(z)}{\partial x_i} \geq 0 \right\}, \\ I_0^-(z) &:= \left\{ i \in \{1, \dots, n\} : \frac{\partial f(z)}{\partial x_i} < 0 \right\} \end{aligned} \quad (2.10)$$

und für die Ungleichungsnebenbedingungsfunktionen

$$\begin{aligned} I_j^+(z) &:= \left\{ i \in \{1, \dots, n\} : \frac{\partial h_j(z)}{\partial x_i} \geq 0 \right\}, \\ I_j^-(z) &:= \left\{ i \in \{1, \dots, n\} : \frac{\partial h_j(z)}{\partial x_i} < 0 \right\}, \end{aligned} \quad (2.11)$$

für alle $j = 1, \dots, m$, bestimmt.

Definition 2.3.3 (Konvexes Teilproblem)

Gegeben sei OP , eine Folge von Entwicklungspunkten $\{z^k\}_{k=0,1,2,\dots} \subset \mathbb{R}^n$ und eine Folge von zulässigen Asymptoten $\{L^k, U^k\}_{k=0,1,2,\dots} \subset \mathbb{R}^{2n}$. Ferner sei $\tau > 0$ konstant. Dann ist das Problem $\mathbf{SP}(z^k, L^k, U^k)$ definiert durch

$$\begin{aligned} \text{Minimiere } & \tilde{f}^k(x, z^k, L^k, U^k, \tau), \\ \text{unter } & \tilde{h}_j^k(x, z^k, L^k, U^k) \leq 0, \quad j = 1, \dots, m, \\ & x \in \mathbb{R}^n, \end{aligned} \quad (2.12)$$

mit

$$\begin{aligned} \tilde{f}^k(x, z^k, L^k, U^k, \tau) &:= \alpha_{0,0}^k(z^k, L^k, U^k) + \sum_{i \in I_0^+(z^k)} \frac{\alpha_{0,i}^k(x_i, z^k, L_i^k, U_i^k, \tau)}{U_i^k - x_i} \\ &\quad - \sum_{i \in I_0^-(z^k)} \frac{\alpha_{0,i}^k(x_i, z^k, L_i^k, U_i^k, \tau)}{x_i - L_i^k}, \end{aligned} \quad (2.13)$$

$$\begin{aligned} \tilde{h}_j^k(x, z^k, L^k, U^k) &:= \alpha_{j,0}^k(z^k, L^k, U^k) + \sum_{i \in I_j^+(z^k)} \frac{\alpha_{j,i}^k(z^k, L_i^k, U_i^k)}{U_i^k - x_i} \\ &\quad - \sum_{i \in I_j^-(z^k)} \frac{\alpha_{j,i}^k(z^k, L_i^k, U_i^k)}{x_i - L_i^k}, \end{aligned} \quad (2.14)$$

$$j = 1, \dots, m,$$

und

$$\begin{aligned} \alpha_{0,0}^k(z^k, L^k, U^k) &:= f(z^k) - \sum_{i \in I_0^+(z^k)} \frac{\partial f(z^k)}{\partial x_i} (U_i^k - z_i^k) \\ &\quad + \sum_{i \in I_0^-(z^k)} \frac{\partial f(z^k)}{\partial x_i} (z_i^k - L_i^k), \end{aligned} \quad (2.15)$$

$$\alpha_{0,i}^k(x_i, z^k, L_i^k, U_i^k, \tau) := \begin{cases} \frac{\partial f(z^k)}{\partial x_i} (U_i^k - z_i^k)^2 + \tau(x_i - z_i^k)^2, & \text{falls } i \in I_0^+(z^k), \\ \frac{\partial f(z^k)}{\partial x_i} (z_i^k - L_i^k)^2 - \tau(x_i - z_i^k)^2, & \text{falls } i \in I_0^-(z^k), \end{cases} \quad (2.16)$$

und

$$\alpha_{j,0}^k(z^k, L^k, U^k) := h_j(z^k) - \sum_{i \in I_j^+(z^k)} \frac{\partial h_j(z^k)}{\partial x_i} (U_i^k - z_i^k) + \sum_{i \in I_j^-(z^k)} \frac{\partial h_j(z^k)}{\partial x_i} (z_i^k - L_i^k), \quad (2.17)$$

$$\alpha_{j,i}^k(z^k, L_i^k, U_i^k) := \begin{cases} \frac{\partial h_j(z^k)}{\partial x_i} (U_i^k - z_i^k)^2, & \text{falls } i \in I_j^+(z^k), \\ \frac{\partial h_j(z^k)}{\partial x_i} (z_i^k - L_i^k)^2, & \text{falls } i \in I_j^-(z^k), \end{cases} \quad (2.18)$$

für $j = 1, \dots, m$, und heißt konvexes Teilproblem in z^k des nichtlinearen Optimierungsproblems OP.

Mit $\tau > 0$ wird garantiert, daß die Approximation $\tilde{f}^k : \mathbb{R}^n \rightarrow \mathbb{R}$ der Zielfunktion streng konvex ist. Anderenfalls wäre für $\frac{\partial f(z^k)}{\partial x_i} = 0$ die strenge Konvexität verletzt, die für den Nachweis der globalen Konvergenzeigenschaft benötigt wird.

Im Teilproblem ist die Variable $x \in \mathbb{R}^n$ ein Element einer kompakten Teilmenge des \mathbb{R}^n . Dies folgt aus der Zulässigkeit der Asymptoten $(L^k, U^k) \in \mathbb{R}^{2n}$, denn es gilt

$$L_i^k < x_i < U_i^k, \quad i = 1, \dots, n, \quad (2.19)$$

für zulässige Asymptoten nach Definition 2.3.2. Genauer ist für $k \in \mathbb{N}_0$ und konstantes $\xi > 0$ die Variable ein Element der Menge

$$\tilde{X}^k := \{x \in \mathbb{R}^n : L_i^k + \xi \leq x_i \leq U_i^k - \xi, \quad i = 1, \dots, n\}. \quad (2.20)$$

Falls in der Aufgabenstellung Schranken für $x \in \mathbb{R}^n$ vorgegeben sind, müssen diese im Teilproblem mit den Asymptoten abgestimmt werden. Eine Möglichkeit dieser Abstimmung kann in Zillober [74] bzw. im Kapitel 4 nachgelesen werden.

Durch die Konstruktion der Funktionen des Teilproblems mittels reziproker Variablen wird erreicht, daß

- $\tilde{f}^k : \mathbb{R}^n \rightarrow \mathbb{R}$ und $\tilde{h}_j^k : \mathbb{R}^n \rightarrow \mathbb{R}$, $j = 1, \dots, m$, Approximationen erster Ordnung von $f : \mathbb{R}^n \rightarrow \mathbb{R}$ bzw. $h_j : \mathbb{R}^n \rightarrow \mathbb{R}$ sind,
- die Funktion \tilde{f}^k streng konvex ist,
- die Funktionen \tilde{h}_j^k , $j = 1, \dots, m$, konvex sind,

- \tilde{f}^k und \tilde{h}_j^k , $j = 1, \dots, m$, separabel sind.

Die Konvexität und die Approximation erster Ordnung sind in erster Linie Ziel der Konstruktion der Teilprobleme $SP(z^k, L^k, U^k)$. Die Separabilität kann in geeigneten Verfahren zur Lösung eines Teilproblems effizient genutzt werden, siehe z.B. Zillober [69].

Es folgt eine allgemein bekannte Tatsache für konvexe Optimierungsprobleme.

Bemerkung 2.3.4

Gegeben sei OP , eine Folge von Entwicklungspunkten $\{z^k\}_{k=0,1,2,\dots}$ und eine Folge von zulässigen Asymptoten $\{L^k, U^k\}_{k=0,1,2,\dots}$. Die Menge

$$\Omega(z^k, L^k, U^k) := \left\{ x \in \tilde{X}^k : \tilde{h}_j^k(x, z^k, L^k, U^k) \leq 0, j = 1, \dots, m \right\} \quad (2.21)$$

heißt zulässiger Bereich des Teilproblems $SP(z^k, L^k, U^k)$. Der zulässige Bereich ist konvex und kompakt. Ist der zulässige Bereich nicht leer, hat das Teilproblem aufgrund der streng konvexen Zielfunktion $\tilde{f}^k : \mathbb{R}^n \rightarrow \mathbb{R}$ eine eindeutige Lösung y^k .

Der zulässige Bereich $\Omega(z^k, L^k, U^k)$ eines Teilproblems $SP(z^k, L^k, U^k)$ kann leer sein, obwohl der zulässige Bereich des zugrundeliegenden Problems OP nicht leer ist. In diesem Fall kann durch Einführen von zusätzlichen Variablen, den sogenannten Schlupfvariablen, ein geeignetes Ersatzproblem mit einem nichtleeren und konvexen zulässigen Bereich aufgestellt werden. Die Lösung des Ersatzproblems ist ein Vektor bei dem der Betrag der Schlupfvariablen minimiert ist. In Kapitel 4 wird dieser Ansatz im Detail beschrieben.

Für die Konvergenzbeweise in dieser Arbeit wird die gleichmäßige Beschränktheit von Funktionenfolgen benötigt. Diese wird für diese Arbeit wie folgt definiert.

Definition 2.3.5 (Gleichmäßige Beschränktheit)

Sei $\{D^k\}_{k=0,1,2,\dots} \subset \mathbb{R}^n$ eine Folge von Teilmengen. Eine Folge skalarer Funktionen $\{\tilde{g}^k : D^k \rightarrow \mathbb{R}\}_{k=0,1,2,\dots}$ heißt auf D^k gleichmäßig nach unten (oben) beschränkt bzgl. k , wenn ein $b \in \mathbb{R}$ existiert mit $\tilde{g}^k(x) \geq b$ ($\tilde{g}^k(x) \leq b$) für alle $x \in D^k$ und für alle $k \in \mathbb{N}_0$. \tilde{g}^k heißt gleichmäßig beschränkt bzgl. k , wenn die Folge gleichzeitig gleichmäßig nach unten und nach oben beschränkt ist.

Das anschließende Lemma zeigt die gleichmäßige Beschränktheit der Funktionen einer Folge von konvexen Teilproblemen $\{SP(z^k, L^k, U^k)\}_{k=0,1,2,\dots}$, d.h. die Folgen der Approximationen der Zielfunktion bzw. der Ungleichungsnebenbedingungen sind unabhängig von $k \in \mathbb{N}_0$ beschränkt. Hierfür wird benötigt, daß die Variable $x \in \mathbb{R}^n$ einen gewissen Abstand zu den Asymptoten einhält. In Zillober [74] wird dies durch eine spezielle Transformation der Schranken in das Teilproblem erreicht. Allerdings ist für zulässige Asymptoten diese Voraussetzung automatisch erfüllt, d.h. im Verfahren TRSCP werden keine vorgegebenen Schranken benötigt. Hier wird ein Mindestabstand zwischen Variable und den Asymptoten durch eine spezielle Wahl der Asymptoten erreicht, d.h. das folgende Lemma ist für TRSCP gültig.

Lemma 2.3.6

Gegeben sei OP , eine Folge von Entwicklungspunkten $\{z^k\}_{k=0,1,2,\dots}$ und eine Folge von zulässigen Asymptoten $\{L^k, U^k\}_{k=0,1,2,\dots}$. Die Mengen $\{\tilde{X}^k\}_{k=0,1,2,\dots}$ seien für ein $\xi > 0$ bestimmt und es sei $\tau > 0$ konstant. Die Folgen

- $\left\{ \tilde{f}^k : \tilde{X}^k \rightarrow \mathbb{R} \right\}_{k=0,1,2,\dots}$,
- $\left\{ \tilde{h}_j^k : \tilde{X}^k \rightarrow \mathbb{R} \right\}_{k=0,1,2,\dots}$, für $j = 1, \dots, m$,
- $\left\{ \nabla_x \tilde{f}^k : \tilde{X}^k \rightarrow \mathbb{R} \right\}_{k=0,1,2,\dots}$,
- $\left\{ \nabla_x \tilde{h}_j^k : \tilde{X}^k \rightarrow \mathbb{R} \right\}_{k=0,1,2,\dots}$, für $j = 1, \dots, m$,

der konvexen Teilprobleme $SP(z^k, L^k, U^k)$ sind auf $\{\tilde{X}^k\}_{k=0,1,2,\dots}$ gleichmäßig beschränkt bezüglich k .

Beweis:

Zillober [69]

Die gleichmäßige Beschränktheit der Approximationen wird für die Beweise benötigt, die die globale Konvergenzeigenschaft des SCP-Verfahrens von Zillober [74] und von TRSCP zeigen. Zu beachten ist hierbei, daß die Beschränktheit dieser Funktionen nicht auf ganz \mathbb{R}^n gilt.

Um sicherzustellen, daß die Nebenbedingungen in OP regulär sind, ist eine 'constraint qualification' zu erfüllen.

Definition 2.3.7 (LICQ)

Sei OP gegeben. Die Linear Independence Constraint Qualification (**LICQ**) ist für ein bzgl. OP zulässiges $x \in \mathbb{R}^n$ erfüllt, wenn die Gradienten aktiver Nebenbedingungen in x linear unabhängig sind.

2.4 KKT-Punkte

Die Karush-Kuhn-Tucker-Bedingung definiert einen stationären Punkt für das nicht-lineare Optimierungsproblem. Mit dem Karush-Kuhn-Tucker Punkt wird eine lokale Lösung von OP definiert.

Definition 2.4.1 (KKT-Bedingung)

Sei OP gegeben. Für $x \in \mathbb{R}^n$ und $u \in \mathbb{R}^m$ heißt

$$\nabla_x f(x) + \sum_{j=1}^m u_j \nabla_x h_j(x) = 0, \quad (2.22)$$

$$h_j(x) \leq 0, \quad j = 1, \dots, m, \quad (2.23)$$

$$u_j h_j(x) = 0, \quad j = 1, \dots, m, \quad (2.24)$$

$$u_j \geq 0, \quad j = 1, \dots, m, \quad (2.25)$$

Karush-Kuhn-Tucker-Bedingung. Ist die KKT-Bedingung für (x, u) erfüllt, dann heißt (x, u) Karush-Kuhn-Tucker-Punkt.

Der Vektor x wird auch als primale Variable des nichtlinearen Problems, der Vektor u als duale Variable oder Lagrange-Multiplikator bezeichnet. Die Begriffe „KKT-Punkt“ und „stationärer Punkt“ werden in dieser Arbeit synonym verwendet.

Für den Nachweis der globalen Konvergenzeigenschaft des SCP-Verfahrens aus Zillober [74] wird ein konvexes Teilproblem für einen stationären Punkt als Entwicklungspunkt aufgestellt und betrachtet. Für dieses kann o.B.d.A. ein beliebiges Paar von zulässigen Asymptoten gewählt werden.

Bezeichnung 2.4.2

Gegeben seien OP mit einem stationären Punkt $(x^*, u^*) \in \mathbb{R}^{n+m}$ und für x^* zulässige Asymptoten (L^*, U^*) . Dann bezeichnet $SP(x^*, L^*, U^*)$ ein konvexes Teilproblem von OP im Entwicklungspunkt x^* .

Den Zusammenhang zwischen einem stationären Punkt eines nichtlinearen Optimierungsproblems und der Lösungen eines konvexen Teilproblems liefert das folgende Lemma.

Lemma 2.4.3

Gegeben sei OP und der Entwicklungspunkt $x^* \in \mathbb{R}^n$ mit zulässigen Asymptoten $(L^*, U^*) \in \mathbb{R}^{2n}$. Es existiert genau dann ein Lagrange-Multiplikator $u^* \in \mathbb{R}^m$, so daß (x^*, u^*) ein stationärer Punkt von OP ist, wenn x^* der primale Anteil des stationären Punktes des Teilproblems $SP(x^*, L^*, U^*)$ ist.

Beweis:

Zillober [69]

Dies bedeutet, daß x^* eine Lösung für OP ist, wenn x^* die Lösung eines Teilproblems mit dem Entwicklungspunkt $z := x^*$ ist.

Das SCP-Verfahren von Zillober kombiniert die Approximationsidee von MMA mit einer Line-Search-Methode bezüglich einer Merit-Funktion und erhält dadurch globale Konvergenzeigenschaften. Als Merit-Funktion kommt die erweiterte Lagrange-Funktion zum Einsatz. Mit diesem Ansatz wurden schon SQP-Methoden stabilisiert,

siehe z.B. Schittkowski [46]. Zur Vereinfachung der Notation wird hierfür die Indexmenge

$$M := \{j : j = 1, \dots, m\} \quad (2.26)$$

definiert.

Definition 2.4.4 (Erweiterte Lagrange-Funktion)

Die zu OP gehörende erweiterte Lagrange-Funktion $\Phi_r : \mathbb{R}^{n+m} \rightarrow \mathbb{R}$ ist für $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$ und $r \in \mathbb{R}^m$ mit $u_j \geq 0$, $r_j > 0$, für $j \in M$, definiert durch

$$\Phi_r \begin{pmatrix} x \\ u \end{pmatrix} := f(x) + \sum_{j \in M} \begin{cases} u_j h_j(x) + \frac{r_j}{2} h_j^2(x), & \text{für } -\frac{u_j}{r_j} \leq h_j(x) \\ -\frac{u_j^2}{2r_j}, & \text{sonst.} \end{cases} \quad (2.27)$$

u heißt Lagrange-Multiplikator, r Penalty-Parameter.

Bei Zillober [74] wird eine kompakte Menge X in OP über untere und obere Schranken für die primale Variable definiert (analog (1.9)). In diesem Fall müssen diese Nebenbedingungen in die erweiterte Lagrange-Funktion aufgenommen werden. Die Dimension der dualen Variablen ist dann $m + 2n$.

Die ersten erweiterten Lagrange-Funktionen wurden von Hestenes [26] und Powell [39] vorgestellt. Allgemein gehören diese Funktionen zu der Klasse der Penalty-Funktionen. Im Gegensatz zu anderen Penalty-Methoden, können mit der erweiterten Lagrange-Funktion Konvergenzaussagen für endliche Penalty-Parameter getroffen werden. Eine Einführung in das Thema erweiterte Lagrange-Funktion findet sich in Bertsekas [5], einen Überblick über grundlegende Arbeiten gibt Conn et al. [8].

Die Minimierung der erweiterten Lagrange-Funktion bezüglich $(x, u) \in \mathbb{R}^{n+m}$ für ein festes $r \in \mathbb{R}^m$ mit $r_j > 1$, $j \in M$, ist ein unrestringiertes Optimierungsproblem. Die erweiterte Lagrange-Funktion $\Phi_r : \mathbb{R}^{n+m} \rightarrow \mathbb{R}$ mit $r \in \mathbb{R}^m$ ist exakt, d.h. ein stationärer Punkt (x^*, u^*) von OP ist ein stationärer Punkt des unrestringierten Optimierungsproblems, das mit einem hinreichend großen Penalty-Parameter gebildet wird, und umgekehrt.

Für den Nachweis der globalen Konvergenzeigenschaft des SCP-Verfahrens aus Zillober [74] werden die zwei folgenden Ergebnisse aus Fletcher [15], [16] benötigt.

- Es ist genau dann $\nabla \Phi_r \begin{pmatrix} x^* \\ u^* \end{pmatrix} = 0$ für alle $r > 0$, wenn (x^*, u^*) stationär für OP ist.
- Sei $(x^*, u^*) \in \mathbb{R}^{n+m}$ ein stationärer Punkt von OP. Falls die LICQ in x^* erfüllt ist, existiert ein $\bar{r} \in \mathbb{R}^m$ mit $\bar{r} > 0$, so daß x^* die erweiterte Lagrange-Funktion $\Phi_r : \mathbb{R}^{n+m} \rightarrow \mathbb{R}$ minimiert für alle $r \geq \bar{r}$.

Die Beschränktheit der erweiterten Lagrange-Funktion nach unten ist für beschränkte Variable unabhängig vom Penalty-Parameter gegeben.

Bemerkung 2.4.5

Sei $x \in \mathbb{R}^n$, mit $|x_i| \leq b < \infty$, $i = 1, \dots, n$, und $u \in \mathbb{R}^m$, mit $0 \leq u_j \leq v^{\max} < \infty$, $j \in M$. Es existiert $\bar{r} \in \mathbb{R}^m$, mit $\bar{r}_j \geq 1$, $j \in M$, so daß die erweiterte Lagrange-Funktion $\Phi_r : \mathbb{R}^{n+m} \rightarrow \mathbb{R}$ nach unten beschränkt ist für alle x , u und $r \in \mathbb{R}^m$ mit $r_j \geq \bar{r}_j$, $j \in M$.

Es werden Bezeichnungen festgelegt, die im Weiteren die Notation vereinfachen.

Bezeichnung 2.4.6

Sei OP gegeben. Weiter sei $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$ mit $u_j \geq 0$ für $j \in M$ und $r \in \mathbb{R}^m$ mit $r_j > 0$ für $j \in M$. Es ist

1. $A(r, x, u) := \left\{ j \in M : h_j(x) \geq -\frac{u_j}{r_j} \right\}$,
2. $K(r, x, u) := M \setminus A(r, x, u)$,
3. $\bar{u} := (\bar{u}_1, \dots, \bar{u}_m)^T$ mit $\bar{u}_j := \begin{cases} u_j, & \text{für } j \in A(r, x, u), \\ 0, & \text{für } j \in K(r, x, u), \end{cases}$
4. $h(x) := (h_1(x), \dots, h_m(x))^T$,
5. $\bar{h}(x) := (\bar{h}_1(x), \dots, \bar{h}_m(x))^T$ mit $\bar{h}_j(x) := \begin{cases} h_j(x), & \text{für } j \in A(r, x, u), \\ 0, & \text{für } j \in K(r, x, u), \end{cases}$
6. $\hat{h}(r, x, u) := \left(\hat{h}_1(r, x, u), \dots, \hat{h}_m(r, x, u) \right)^T$
mit $\hat{h}_j(r, x, u) := \begin{cases} h_j(x), & \text{für } j \in A(r, x, u), \\ -\frac{u_j}{r_j}, & \text{für } j \in K(r, x, u), \end{cases}$
7. $J_h(x) := (\nabla_x h_1(x), \dots, \nabla_x h_m(x))$,
8. $P := \text{diag}(r_1, \dots, r_m) \in \mathbb{R}^{m \times m}$.

Bemerkung 2.4.7

Mit der in Bezeichnung 2.4.6 eingeführten Notation ist die partielle Ableitung der erweiterten Lagrange-Funktion von OP gegeben durch

$$\nabla \Phi_r \begin{pmatrix} x \\ u \end{pmatrix} := \begin{pmatrix} \nabla_x f(x) + J_h(x) (\bar{u} + P\bar{h}(x)) \\ \hat{h}(r, x, u) \end{pmatrix}. \quad (2.28)$$

2.5 Ein Algorithmus mit konvexen Approximationen

Bevor der Algorithmus aus Zillober [74] vorgestellt wird, wird ein Parameter eingeführt, der bei der Aktualisierung der Penalty-Parameter der erweiterten Lagrange-Funktion eingesetzt wird.

Bezeichnung 2.5.1

Gegeben sei OP , eine Folge von Entwicklungspunkten $\{z^k\}_{k=0,1,2,\dots}$ und eine Folge von zulässigen Asymptoten $\{L^k, U^k\}_{k=0,1,2,\dots}$. Ferner sei $\tau > 0$ konstant und $y^k \in \tilde{X}^k$ der primale Anteil der Lösung des Teilproblems $SP(z^k, L^k, U^k)$. Dann ist

$$\eta^k := \min_{i=1,\dots,n} \eta(z^k, y_i^k, L_i^k, U_i^k, \tau) \quad (2.29)$$

mit

$$\eta(z^k, y_i^k, L_i^k, U_i^k, \tau) := \begin{cases} \left(\frac{\partial f(z^k)}{\partial x_i} + \tau \right) \frac{2U_i^k - y_i^k - z_i^k}{(U_i^k - y_i^k)^2} & , \text{ falls } i \in I_0^+(z^k), \\ \left(\tau - \frac{\partial f(z^k)}{\partial x_i} \right) \frac{-2L_i^k + y_i^k + z_i^k}{(y_i^k - L_i^k)^2} & , \text{ falls } i \in I_0^-(z^k), \end{cases} \quad (2.30)$$

für $i = 1, \dots, n$.

Zu beachten ist, daß $\eta(z^k, y_i^k, L_i^k, U_i^k, \tau) > 0$ für alle $i = 1, \dots, n$.

Es wird eine Suchrichtung s^k aus der Differenz von der aktuellen Näherungslösung $z^k \in \mathbb{R}^n$ mit dem zugehörigen Lagrange-Multiplikator $u^k \in \mathbb{R}^m$ zum stationären Punkt (y^k, v^k) des konvexen Teilproblems $SP(z^k, L^k, U^k)$, $k \in \mathbb{N}_0$, mit $y^k \in \mathbb{R}^n$ und $v^k \in \mathbb{R}^m$ gebildet, d.h.

$$s^k := \begin{pmatrix} y^k - z^k \\ v^k - u^k \end{pmatrix}. \quad (2.31)$$

Für hinreichend große Penalty-Parameter wird die Suchrichtung zur Abstiegsrichtung für die erweiterte Lagrange-Funktion.

Es folgt eine kurze Beschreibung des SCP-Algorithmus aus Zillober [74]. Er besteht aus fünf Schritten und einem Initialisierungsschritt. Im ersten Schritt werden zulässige Asymptoten gebildet. Im zweiten Schritt wird das konvexe Teilproblem aufgestellt und gelöst. Ist die Lösung des Teilproblems identisch mit dem aktuellen Entwicklungspunkt, ist ein stationärer Punkt von OP gefunden worden und die Iteration wird beendet. Im dritten Schritt wird der Penalty-Parameter der erweiterten Lagrange-Funktion angepaßt, so daß die Suchrichtung zu einer Abstiegsrichtung bezüglich dieser Funktion wird. Diese Anpassung erfolgt durch eine Teiliteration. Im vierten Schritt wird entlang der Suchrichtung eine Schrittweite bestimmt, so daß ein hinreichender Abstieg realisiert wird. Dies wird als Line-Search bezeichnet. Im Anschluß an die Line-Search wird im Schritt 5 die neue Näherung bestimmt und mit der nächsten Iteration begonnen.

Des Verfahren aus Zillober [74] kombiniert die Optimierung mittels konvexer Approximation mit einer Line-Search-Stabilisierung und wird in dieser Arbeit mit LSSCP (Line-Search Sequential Convex Programming) bezeichnet.

Algorithmus 2.4 (LSSCP)

Sei OP gegeben.

Schritt 0:

Wähle einen Entwicklungspunkt $z^0 \in \mathbb{R}^n$ und einen Lagrange-Multiplikator $u^0 \in \mathbb{R}^m$ mit $u_j^0 \geq 0$ für $j \in M$. Wähle einen Penalty-Parameter $r_0^{-1} \in \mathbb{R}^m$ mit $r_{0,j}^{-1} \geq 1$ für $j \in M$ und Konstanten $\gamma \in (0; 1)$, $\vartheta \in (0; 1)$, $\tau > 0$ und $1 < \kappa_1 < \kappa_2$. Berechne $f(z^0)$, $\nabla_x f(z^0)$ und $h_j(z^0)$, $\nabla_x h_j(z^0)$ für $j \in M$. Initialisiere $k := 0$.

Schritt 1:

Wähle für z^k beliebige zulässige Asymptoten L^k und U^k .

Schritt 2:

Bestimme den stationären Punkt (y^k, v^k) des Teilproblems $SP(z^k, L^k, U^k)$. Ist $z^k = y^k$, dann ist $(x^*, u^*) := (z^k, u^k)$ stationärer Punkt von OP , STOP.

Anderenfalls bestimme die Suchrichtung $s^k := \begin{pmatrix} y^k - z^k \\ v^k - u^k \end{pmatrix}$, $\delta^k := \|y^k - z^k\|_2$ und η^k nach 2.5.1 und initialisiere $p := 0$.

Schritt 3:

Falls $\nabla \Phi_{r_p^{k-1}} \begin{pmatrix} z^k \\ u^k \end{pmatrix}^T s^k > -\frac{\eta^k (\delta^k)^2}{2}$, bestimme

$$r_{p+1,j}^{k-1} := \min \left\{ \kappa_2 r_{p,j}^{k-1}; \max \left\{ \kappa_1 r_{p,j}^{k-1}; \left| \frac{2(v_j^k - u_j^k)}{h_j(z^k)} \right| \right\} \right\}, \quad (2.32)$$

für alle $j \in A(r_p^{k-1}, y^k, v^k)$, falls $h_j(z^k) > 0$ und $\nabla_x h_j(z^k)^T (y^k - z^k) \neq 0$ bzw. falls $h_j(z^k) < 0$ und $\nabla_x h_j(z^k)^T (y^k - z^k) > 0$.

Anderenfalls setze

$$r_{p+1,j}^{k-1} := \kappa_1 r_{p,j}^{k-1}. \quad (2.33)$$

Weiter bestimme

$$r_{p+1,j}^{k-1} := \min \left\{ \kappa_2 r_{p,j}^{k-1}; \max \left\{ \kappa_1 r_{p,j}^{k-1}; \left| \frac{u_j^k (v_j^k - u_j^k) 4m}{\eta^k (\delta^k)^2} \right| \right\} \right\}, \quad (2.34)$$

für alle $j \in K(r_p^{k-1}, y^k, v^k)$, falls $v_j^k - u_j^k < 0$.

Anderenfalls setze

$$r_{p+1,j} := \kappa_1 r_{p,j}. \quad (2.35)$$

Weiter setze $p := p + 1$ und wiederhole SCHRITT 3.

Anderenfalls setze $r_0^k := r_p^{k-1}$ und initialisiere $a := 0$, $\sigma_a := 1$.

Schritt 4:

Berechne $f(z^k + \sigma_a(y^k - z^k))$ und $h_j(z^k + \sigma_a(y^k - z^k))$ für $j \in M$.

Falls

$$\Phi_{r_0^k} \begin{pmatrix} z^k \\ u^k \end{pmatrix} - \Phi_{r_0^k} \left(\begin{pmatrix} z^k \\ u^k \end{pmatrix} + \sigma_a s^k \right) < -\gamma \sigma_a \nabla \Phi_{r_0^k} \begin{pmatrix} z^k \\ u^k \end{pmatrix}^T s^k, \quad (2.36)$$

setze $\sigma_{a+1} := \vartheta \sigma_a$, $a := a + 1$ und wiederhole *SCHRITT 4*,
andernfalls setze $\sigma^k := \sigma_a$.

Schritt 5:

Setze

$$\begin{pmatrix} z^{k+1} \\ u^{k+1} \end{pmatrix} := \begin{pmatrix} z^k \\ u^k \end{pmatrix} + \sigma^k s^k, \quad (2.37)$$

berechne $\nabla_x f(z^{k+1})$, $\nabla_x h_j(z^{k+1})$, $j \in M$,
setze $k := k + 1$ und gehe zu *SCHRITT 1*.

Bemerkung 2.5.2

1. Schritt 4 beruht auf der Armijo-Bedingung. Es kann gezeigt werden, daß diese Teiliteration endlich ist (siehe Schittkowski [47]). Ausgehend von der Schrittweite Eins wird diese sukzessiv reduziert, bis erstmals ein hinreichender Abstieg erreicht wird.
2. Eine Kombination der Armijo-Bedingung mit einer quadratischen Interpolation ist möglich (siehe Schittkowski [47]).
3. In jedem Teiliterationsschritt der Line-Search-Methode werden die Funktionswerte einmal berechnet, in jedem Schritt der Hauptiteration werden die Gradientenwerte einmal berechnet.
4. LSSCP ist ein Verfahren 1. Ordnung.

In Schritt 0 wird der Algorithmus LSSCP initialisiert. Ein Vektor $z^0 \in \mathbb{R}^n$ und ein Lagrange-Multiplikator $u^0 \in \mathbb{R}^m$ werden als Startwerte gewählt. z^0 ist der erste Entwicklungspunkt für die konvexen Teilprobleme. Sind Schranken für die primale Variable gegeben, muß z^0 so angepaßt werden, daß die Schranken nicht verletzt werden. Eine Möglichkeit hierfür ist in Kapitel 4 ausgeführt. Ferner werden alle benötigten Konstanten festgelegt. Die Zielfunktion und Nebenbedingungsfunktionen, wie auch deren Gradienten, werden für z^0 ausgewertet. Der Iterationsparameter wird zu Beginn mit $k := 0$ initialisiert.

In Schritt 1 sind beliebige für z^k zulässige Asymptoten zu wählen. Die Asymptoten aus Definition 2.3 sind für die folgenden theoretischen Betrachtungen nicht geeignet, da ein konstanter Mindestabstand $\xi > 0$ für unendliche Folgen von Asymptoten nicht gewährleistet werden kann. Das heißt, eine durch Definition 2.3 erzeugte unendliche Folge von Asymptoten kann unzulässig sein.

In Schritt 2 wird das konvexe Teilproblem $SP(z^k, L^k, U^k)$ aufgestellt und gelöst. Unter der Annahme, daß der zugehörige zulässige Bereich nicht leer ist, hat das

strikt konvexe Teilproblem die eindeutige Lösung (y^k, v^k) . Ist (y^k, v^k) kein stationärer Punkt von OP, wird die aktuelle Suchrichtung gebildet, sowie das Maß für einen hinreichenden Abstieg festgelegt. Anschließend wird der Iterationsparameter der Teiliteration zur Anpassung des Penalty-Parameters der erweiterten Lagrange-Funktion durch $p := 0$ initialisiert.

In Schritt 3 wird der Penalty-Parameter der erweiterten Lagrange-Funktion so angepaßt, daß die Suchrichtung zur Abstiegsrichtung für diese Penalty-Funktion wird. Dies erfolgt durch eine endliche Teiliteration, was im Abschnitt 2.6 ersichtlich wird. Der Parameter κ_1 verhindert, daß der Penalty-Parameter gegen eine obere Schranke konvergiert, κ_2 sorgt dafür, daß der Wert nicht zu schnell ansteigt. Für den Algorithmus LSSCP ist es dabei nicht notwendig, den kleinsten Penalty-Parameter zu bestimmen, für den die Abstiegseigenschaft erfüllt ist. Es ist ausreichend, wenn der Penalty-Parameter hinreichend groß ist. Im Abschnitt 2.6 wird die Abstiegseigenschaft der erweiterten Lagrange-Funktion nachgewiesen. Im Anschluß an diese Aktualisierung des Penalty-Parameters wird der Iterationsparameter für die nächste Teiliteration zur Erfüllung der Armijo-Bedingung mit $a := 0$ initialisiert. Über $\sigma_a := 1$ wird die maximale Schrittweite längs der Suchrichtung für diese Iteration gesetzt.

In Schritt 4 wird mit einer weiteren Teiliteration eine hinreichende Abstiegsbedingung erfüllt. Hierfür wird die Schrittweite entlang der Suchrichtung so lange reduziert, bis der tatsächlich erreichte Abstieg der erweiterten Lagrange-Funktion hinreichend für eine Konvergenz ist. In jeder Teiliteration ist es notwendig, Zielfunktion und Nebenbedingungsfunktionen neu zu berechnen. Diese Teiliteration ist ebenfalls endlich.

In Schritt 5 wird mit der durch Schritt 4 berechneten Schrittweite der nächste Iterationspunkt bestimmt. Für die nachfolgende Iteration müssen die Gradienten der Zielfunktion und der Nebenbedingungsfunktionen neu berechnet werden. Anschließend wird der Iterationsindex erhöht und mit Schritt 1 fortgefahren.

2.6 Globale Konvergenz von LSSCP

In diesem Abschnitt wird kurz die Idee für den globalen Konvergenzbeweis des SCP-Algorithmus aus Zillober [74] vorgestellt und die wichtigsten Sätze hierzu aufgeführt. Für die Beweise wird auf Zillober [69, 74] verwiesen. Im Abschnitt 3.3 wird eine ähnliche Vorgehensweise für TRSCP ausführlich dargestellt, bei der entsprechende Beweise für das SCP-Verfahren mit Trust-Region-Stabilisierung ausgeführt werden.

Um die globale Konvergenzeigenschaft von LSSCP nachzuweisen, wird gezeigt, daß die von Algorithmus 2.4 erzeugte Folge $\{u^k\}_{k=0,1,2,\dots} \in \mathbb{R}^m$ von Lagrange-Multiplikatoren beschränkt ist. Weiter wird nachgewiesen, daß die berechnete Suchrichtung $s^k \in \mathbb{R}^{n+m}$ eine Abstiegsrichtung der erweiterten Lagrange-Funktion $\Phi_{r_0^k} \begin{pmatrix} z^k \\ u^k \end{pmatrix}$ ist.

Ist der Penalty-Parameter $r_0^k \in \mathbb{R}^m$ unabhängig vom Iterationsparameter k beschränkt, dann wird gezeigt, daß die Folge $\{(z^k, u^k)\}_{k=0,1,2,\dots}$ mindestens einen Häufungspunkt besitzt und, daß alle Häufungspunkte dieser Folge stationär sind. Weiter wird für unbeschränkte Penalty-Parameter nachgewiesen, daß die Folge mindestens einen stationären Häufungspunkt besitzt.

Als erstes wird im folgenden Satz gezeigt, daß die Lagrange-Multiplikatoren u^k , d.h. die dualen Variablen, für alle $k \in \mathbb{N}_0$ beschränkt sind.

Satz 2.5

Sei OP und $\tau > 0$ konstant gegeben. Weiter sei $\{(z^k, u^k)\}_{k=0,1,2,\dots}$ vom Algorithmus 2.4 erzeugt, die Asymptoten $\{L^k, U^k\}_{k=0,1,2,\dots}$ seien für $\{z^k\}_{k=0,1,2,\dots}$ zulässig, alle Teilprobleme $SP(z^k, L^k, U^k)$ seien lösbar und LICQ sei in jedem Iterationspunkt erfüllt. Dann ist $\{u^k\}_{k=0,1,2,\dots} \subset \mathbb{R}^m$ gleichmäßig beschränkt bzgl. $k \in \mathbb{N}_0$.

Beweis:

Zillober [74]

In der Praxis kann die Lösbarkeit der Teilprobleme Schwierigkeiten bereiten, da ihr zulässiger Bereich leer sein kann. Dies kann, wie unter Bemerkung 2.3.4 angesprochen, durch Einführen einer künstlichen Variablen gelöst werden. Ausführungen hierzu finden sich im Kapitel 4.

Der anschließende Satz zeigt die Beschränktheit der Penalty-Parameter r_0^k der erweiterten Lagrange-Funktion $\Phi_{r_0^k} : \mathbb{R}^{n+m} \rightarrow \mathbb{R}$. Weiter wird die Abstiegeigenschaft der Suchrichtung s^k für Glieder der Folge $\{(z^k, u^k)\}_{k=0,1,2,\dots}$ nachgewiesen, deren Abstand durch einen Mindestabstand von einer lokalen Lösung (x^*, u^*) nach unten beschränkt ist.

Satz 2.6

Sei OP gegeben und $\tau > 0$ konstant. Weiter sei $\{(z^k, u^k)\}_{k=0,1,2,\dots}$ vom Algorithmus 2.4 erzeugt, die Asymptoten $\{L^k, U^k\}_{k=0,1,2,\dots}$ für $\{z^k\}_{k=0,1,2,\dots}$ zulässig und alle Teilprobleme $SP(z^k, L^k, U^k)$ lösbar. LICQ sei in jedem Iterationspunkt erfüllt und die Parameter s^k, η^k, δ^k seien wie in Algorithmus 2.4 definiert. Dann existieren Penalty-Parameter $\bar{r} \in \mathbb{R}^m$ mit $\bar{r}_j^k > 0, j \in M$, so daß s^k eine Abstiegsrichtung der erweiterten Lagrange-Funktion ist, d.h.

$$\nabla \Phi_r \begin{pmatrix} z^k \\ u^k \end{pmatrix}^T s^k \leq -\frac{\eta^k (\delta^k)^2}{2}, \quad (2.38)$$

für alle $r \in \mathbb{R}^m$ mit $r_j \geq \bar{r}_j^k, j \in M$. Ferner gilt, daß für $\delta > 0$ ein $\bar{r}^\delta \in \mathbb{R}^m$ mit $\bar{r}_j^\delta > 0, j \in M$, und ein $\eta > 0$ existiert, so daß für alle (z^k, u^k) mit $\delta^k \geq \delta$

$$\nabla \Phi_r \begin{pmatrix} z^k \\ u^k \end{pmatrix}^T s^k \leq -\frac{\eta^k (\delta^k)^2}{2} \leq -\frac{\eta \delta^2}{2} \quad (2.39)$$

erfüllt ist für alle $r_j \geq \bar{r}_j^\delta, j \in M$.

Beweis:

Zillober [74]

Der zweite Teil der Aussage dieses Satzes besagt, daß für hinreichend große Penalty-Parameter das Produkt aus der Ableitung der erweiterten Lagrange-Funktion und der Suchrichtung s^k für alle $k \in \mathbb{N}_0$ nach oben beschränkt ist, sofern ein konstanter Abstand $\delta > 0$ zwischen (z^k, u^k) und (y^k, v^k) gehalten wird. Das heißt in Fällen, bei denen dieser Abstand besteht, kann für hinreichend große Penalty-Parameter ein vom Iterationsparameter unabhängiger Abstieg der Lagrange-Funktion erzielt werden.

Als eine Voraussetzung in diesem Satz wird die in Definition 2.3.2 eingeführte Zulässigkeit der Asymptoten gefordert. Für den nachfolgenden Algorithmus TRSCP muß diese Zulässigkeit erzwungen werden, da ansonsten der Beweis der globalen Konvergenz im Abschnitt 3.2 nicht entsprechend der hier vorgestellten Weise geführt werden kann.

Um schwache Konvergenz von LSSCP zu beweisen, genügt die Aussage von Satz 2.6. Das heißt, man kann für unbeschränkte Penalty-Parameter zeigen, daß die erzeugte Folge von Entwicklungspunkten mindestens einen Häufungspunkt besitzt und, daß mindestens ein Häufungspunkt dieser Folge stationär ist.

Die Beschränktheit der Penalty-Parameter in Algorithmus 2.4 bedeutet, daß bedingt durch das gewählte Penalty-Parameter-Update ein $\bar{r} \in \mathbb{R}^m$ mit $\bar{r}_j < \infty$, $j \in M$, und ein $\bar{k} < \infty$ existieren, so daß $r_0^k = \bar{r}$ für alle $k \geq \bar{k}$. Für Penalty-Parameter mit dieser Eigenschaft läßt sich die globale Konvergenz von LSSCP mit folgender starken Aussage beweisen.

Satz 2.7

Sei OP gegeben und $\tau > 0$ konstant. Weiter sei $\{(z^k, u^k)\}_{k=0,1,2,\dots}$ vom Algorithmus 2.4 erzeugt, die Asymptoten $\{L^k, U^k\}_{k=0,1,2,\dots}$ für $\{z^k\}_{k=0,1,2,\dots}$ zulässig und alle Teilprobleme $SP(z^k, L^k, U^k)$ lösbar. Die LICQ sei in jedem Iterationspunkt, als auch jeden Häufungspunkt der Folge erfüllt. Die Parameter s^k , η^k , δ^k seien wie in Algorithmus 2.4 definiert. Es existiert $\bar{r} \in \mathbb{R}^m$ mit $1 < \bar{r}_j < \infty$ und $r_j^k \leq \bar{r}$ für alle $j \in M$ und $k = 0, 1, 2, \dots$. Dann besitzt $\{(z^k, u^k)\}_{k=0,1,2,\dots}$ mindestens einen Häufungspunkt, und jeder Häufungspunkt dieser Folge ist stationär.

Beweis:

Zillober [74]

Kapitel 3

Trust-Region-Methoden

Dieses Kapitel führt die Trust-Region-Methode zur Stabilisierung von sequentiellen Optimierungsverfahren ein. Es werden zwei verschiedene Optimierungsalgorithmen vorgestellt. Der Hauptteil dieses Kapitels enthält das neue SCP-Optimierungsverfahren TRSCP, das aufgrund einer Trust-Region-Methode eine globale Konvergenzeigenschaft besitzt. TRSCP steht für 'Trust-Region Sequential Convex Programming'.

Als erster Teil dieses Kapitels wird ein global konvergenter Trust-Region-Modell-Algorithmus aus Yuan [65] vorgestellt, der das nichtlineare Optimierungsproblem OP (2.5) löst. Anhand dieses Modell-Algorithmus soll das Konzept der Trust-Region-Methode veranschaulicht werden. Der vorgestellte Algorithmus aus Yuan [65] wurde für diese Arbeit ausgewählt, da er einerseits die Vorgehensweise eines Optimierungsverfahrens mit Trust-Region-Stabilisierung gut zeigt und andererseits die gleiche Problemklasse behandelt wie das in Kapitel 2 besprochene SCP-Verfahren von Zill-ober [74] und der anschließend vorgestellte Algorithmus TRSCP.

Im zweiten Teil dieses Kapitels wird das sequentielle Optimierungsverfahren TRSCP vorgestellt. Bei TRSCP wird in jeder Iteration mittels einer MMA-Approximation (Definition 1.0.2) ein konvexes Teilproblem aufgestellt. Hierfür wird die im Kapitel 2 vorgestellte Approximation angepaßt und eingesetzt. Die Konvergenz dieses SCP-Verfahrens wird durch eine Trust-Region-Methode stabilisiert. Als Trust-Region dient ein fest definierter Teilbereich zwischen den MMA-Asymptoten. Den Abschluß des zweiten Teils bildet der Beweis der globalen Konvergenz von TRSCP.

Eine umfassende Einführung in die Trust-Region-Methoden findet sich in Conn et al. [9]. Ferner wird für einen Überblick über Trust-Region-Methoden auf Yuan [66] verwiesen. Eine kurze, allgemeine Beschreibung von Stabilisierungsmethoden von iterativen Optimierungsverfahren gibt Schittkowski und Yuan [51].

3.1 Ein Trust-Region-Modell-Algorithmus

Eine Trust-Region-Methode stabilisiert die Konvergenz eines iterativen Verfahrens zur Lösung eines nichtlinearen Optimierungsproblems (OP), d.h. durch diese Schrittweitenkontrolle erhält das iterative Optimierungsverfahren globale Konvergenzeigenschaften. In jeder Iteration wird ausgehend von einem aktuellen Iterationspunkt $z \in \mathbb{R}^n$ ein Schritt $s \in \mathbb{R}^n$ zu einem möglichen nächsten Iterationspunkt berechnet. Der neue Iterationspunkt muß in einer bestimmten Umgebung des aktuellen Iterationspunkts liegen. Diese Umgebung ist die sogenannte Trust-Region. Die Schrittweite, d.h. der Abstand zwischen aktuellem Iterationspunkt und dem Kandidaten für den nächsten Iterationspunkt, wird hierbei durch den Trust-Region-Radius nach oben beschränkt. Der Schritt wird mittels einer Merit-Funktion bewertet, d.h. es wird eine Prognose erstellt, inwieweit sich durch den Schritt der neue Iterationspunkt im Hinblick auf die Optimierung verbessert. Fällt diese Prognose hinreichend gut aus, werden die Funktionen des OPs für den neuen Punkt berechnet. Die Trust-Region wird anschließend basierend auf dem Ergebnis der Prognose und der tatsächlich erreichten Änderung angepaßt und es wird mit der nächsten Iteration fortgefahren.

Powell [40] hat in den 70er Jahren eine Trust-Region-Methode für unrestringierte Probleme vorgestellt. Weiter präsentierte zu dieser Zeit Fletcher einen Modell-Algorithmus für eine Optimierung mit beschränkter Schrittweite¹. Dieser beschreibt ein Trust-Region-stabilisiertes Optimierungsverfahren und ist, obwohl die Trust-Region-Methode im weitesten Sinn der klassischen Levenberg-Marquardt-Methode (siehe Levenberg [31] bzw. Marquardt [32]) entspricht (siehe Yuan [65]), eine weitere grundlegende Arbeit in diesem Bereich. Der Modell-Algorithmus kann in Fletcher [17] nachgelesen werden. Eine ausführliche Diskussion der Levenberg-Marquardt-Methode findet sich in Morè [33].

Es lassen sich folgende unterschiedliche Ansätze von Trust-Region-Methoden unterscheiden. In der sogenannten null-space-Technik besteht der Schritt zwischen den Iterationspunkten aus einem horizontalen und einem vertikalen Anteil. Der vertikale Anteil dient der Reduzierung der Verletzung der Nebenbedingungen, der horizontale der Reduzierung der Zielfunktion. Wichtige Arbeiten zu diesem Ansatz sind z.B. Vardi [64], Byrd et al. [6] und Omojokun [37]. Ein weiterer Ansatz für eine Trust-Region-Methode basiert auf der Arbeit von Celis et al. [7]. Hier werden im Teilproblem die Nebenbedingungen nicht einzeln linearisiert, sondern es wird eine einzelne quadratische Bedingung eingeführt. Neben Celis et al. [7] ist Powell und Yuan [44] eine weitere wichtige Arbeit zu diesem Ansatz. Hier wird diese quadratische Bedingung näher betrachtet. Eine weitere Möglichkeit basiert auf der Verwendung einer Penalty-Funktion. Der in diesem Abschnitt vorgestellte Algorithmus aus Yuan [65] folgt diesem Ansatz.

Ein aktuellerer Ansatz ist die Kombination von einer Trust-Region mit einem Filter. Dieser Ansatz ermöglicht nicht-monotone Iterationsfolgen, indem die Verbesserung der Zielfunktion und die Verbesserung der Verletzung der Nebenbedingungen separat betrachtet werden. Mit Hilfe eines sogenannten Dominanz-Prinzips erfolgt eine

¹im englischen: restricted step method

Bewertung eines Iterationspunkts. Eine durch eine Filter-Methode erzeugte Iterationsfolge ist monoton im Bezug auf diese Beurteilung. Arbeiten hierzu sind Fletcher und Leyffer [18], Ulbrich und Ulbrich [62] und Fletcher et al. [19]. Eine Kombination von Trust-Region-Methode und Line-Search-Methode findet sich in Nocedal und Yuan [36]. In Gould und Toint [21] wird ein Verfahren der nichtlinearen Optimierung vorgestellt, das ohne Penalty-Funktion und ohne Filter eine globale Konvergenzeigenschaft besitzt. Bei diesem Verfahren kommen Trust-Regions zum Einsatz.

Der hier vorgestellte Modell-Algorithmus aus Yuan [65] entspricht schematisch dem Modell-Algorithmus von Fletcher [17]. Es wird iterativ ein stationärer Punkt des nichtlinearen Optimierungsproblems OP (2.5) gesucht. Hierfür wird in jeder Iteration ein Teilproblem für den aktuellen Iterationspunkt aufgestellt. Ein Iterationspunkt ist eine Näherung für einen stationären Punkt von OP. Im Teilproblem wird eine L_∞ -Meritfunktion mit linearisierten Restriktionen und einer Trust-Region minimiert. Eine Trust-Region ist eine speziell gewählte Menge von Vektoren, in der die Lösung des Teilproblems gesucht wird. Eine Trust-Region wird im Allgemeinen mit Hilfe des sogenannten Trust-Region-Radius um den Iterationspunkt aufgespannt. Die Lösung des Teilproblems wird mit Hilfe einer ebenfalls auf der L_∞ -Norm basierenden Penalty-Funktion bewertet. Mit dem Ergebnis und der Bewertung wird eine neue, in einem gewissen Sinn bessere Näherung für einen stationären Punkt und ein neuer Trust-Region-Radius festgelegt.

Der Modell-Algorithmus in diesem Abschnitt entspricht dem Algorithmus 2.1 in Yuan [65] und ist global konvergent, d.h. die vom Modell-Algorithmus erzeugte Folge von Iterationspunkten konvergiert unabhängig vom Startpunkt gegen einen stationären Punkt von OP.

Die folgende Definition der Abbildung $h^+(x)$ vereinfacht die Darstellung der im Modell-Algorithmus vorkommenden Nebenbedingungen, der zugehörigen Penalty-Funktion und des zu lösenden Teilproblems.

Definition 3.1.1

Sei OP gegeben. $h^+ : \mathbb{R}^n \rightarrow \mathbb{R}^m$ ist definiert, durch

$$h_j^+(x) := \max \{0, h_j(x)\}, \quad j \in M.$$

Die Definition kann ohne Probleme auf Gleichungsrestriktionen erweitert werden. Die Nebenbedingungen des Optimierungsproblems OP lassen sich mit Hilfe der L_∞ -Norm und dieser Definition zu einer einzigen skalaren Restriktion

$$\|h^+(x)\|_\infty = 0 \tag{3.1}$$

umformulieren. Die in diesem Abschnitt verwendete exakte Penalty-Funktion ist definiert durch

$$P(x, \zeta) := f(x) + \zeta \|h^+(x)\|_\infty \tag{3.2}$$

mit dem Penalty-Parameter $\zeta \in \mathbb{R}^+$.

Im hier vorgestellten Trust-Region-Verfahren von Yuan [65] wird in jeder Iteration ein Teilproblem gelöst, das auf einer L_∞ -Meritfunktion basiert. Die Nebenbedingungen von OP werden hierfür linearisiert.

Definition 3.1.2 (Teilproblem mit L_∞ -Meritfunktion)

Gegeben sei OP und ein Entwicklungspunkt $z \in \mathbb{R}^n$. Ferner sei $D \in \mathbb{R}^{n \times n}$ positiv definit, $\zeta > 0$ ein Penalty-Parameter und $\Delta > 0$ ein Trust-Region-Radius. Dann ist das Problem $\mathbf{TP}(z, \Delta, D, \zeta)$ definiert durch

$$\begin{aligned} & \text{Minimiere } S(x, z, D, \zeta), \\ & \text{unter } \|x\|_\infty \leq \Delta, \\ & \quad x \in \mathbb{R}^n, \end{aligned} \tag{3.3}$$

mit

$$S(x, z, D, \zeta) := \nabla_x f(z)^T x + \frac{1}{2} x^T D x + \zeta \|[h(z) + \nabla_x h(z)^T x]^+\|_\infty \tag{3.4}$$

und heißt Teilproblem in z mit L_∞ -Meritfunktion des nichtlinearen Optimierungsproblems OP .

Die Lösung des Teilproblems wird folglich ausschließlich innerhalb des Trust-Region-Radius $\Delta \in \mathbb{R}$ gesucht. Nach diesen Vorbereitungen läßt sich der Algorithmus 2.1 aus Yuan [65] wie folgt darstellen.

Algorithmus 3.1 (Trust-Region-Verfahren mit L_∞ -Meritfunktion)

Sei OP gegeben.

Schritt 0:

Wähle $z^0 \in \mathbb{R}^n$, $\Delta^0 > 0$, $D^0 \in \mathbb{R}^{n \times n}$ positiv definit, $\varpi^0 > 0$, $\zeta^0 > 0$ und initialisiere $k := 0$.

Schritt 1:

Bestimme s^k als Lösung des Teilproblems $TP(z^k, \Delta^k, D^k, \zeta^k)$ und u^k , den zugehörigen Lagrange-Multiplikator. Ist $s^k = 0$, dann ist z^k Lösung von OP , STOP.

Schritt 2:

Berechne den Quotienten

$$Q^k := \frac{P(z^k, \zeta^k) - P(z^k + s^k, \zeta^k)}{S(0, z^k, D^k, \zeta^k) - S(s^k, z^k, D^k, \zeta^k)}. \tag{3.5}$$

Falls $Q^k \geq 0$ gehe zu **SCHRITT 3**, anderenfalls setze $\Delta^{k+1} := \frac{\|s^k\|_\infty}{4}$, $z^{k+1} := z^k$, $k := k + 1$ und gehe zu **SCHRITT 1**.

Schritt 3:

Setze $z^{k+1} := z^k + s^k$ und

$$\Delta^{k+1} := \begin{cases} \max \{2\Delta^k, 4\|s^k\|_\infty\}, & \text{für } Q^k > 0.9, \\ \Delta^k, & \text{für } 0.1 \leq Q^k \leq 0.9, \\ \min \left\{ \frac{\Delta^k}{4}, \frac{\|s^k\|_\infty}{2} \right\}, & \text{für } Q^k < 0.1. \end{cases} \tag{3.6}$$

Erzeuge D^{k+1} aus D^k mit Hilfe eines Quasi-Newton-Verfahrens.

Schritt 4:*Falls*

$$S(0, z^k, D^k, \zeta^k) - S(s^k, z^k, D^k, \zeta^k) < \varpi^k \zeta^k \min \{ \Delta^k, \|h^+(z^k)\|_\infty \}, \quad (3.7)$$

dann setze $\zeta^{k+1} := 2\zeta^k$ und $\varpi^{k+1} := \frac{\varpi^k}{4}$, anderenfalls $\zeta^{k+1} := \zeta^k$ und

$\varpi^{k+1} := \varpi^k$. Setze $k := k + 1$ und gehe zu *SCHRITT 1*.

Die Matrix D^{k+1} kann aus D^k mit Hilfe eines BFGS²-Updateverfahrens bestimmt werden. Yuan [65] schlägt hierfür die Safeguard-BFGS-Aktualisierung von Powell [42] vor, d.h.

$$D^{k+1} := D^k - \frac{D^k s^k (D^k s^k)^T}{(s^k)^T D^k s^k} + \frac{\varrho^k (\varrho^k)^T}{(\varrho^k)^T s^k}. \quad (3.8)$$

Hierbei ist $\varrho^k \in \mathbb{R}^n$ eine Konvexkombination aus einem $\omega^k \in \mathbb{R}^n$ und $D^k s^k$. Der Vektor ϱ^k muß der Bedingung

$$(\varrho^k)^T s^k > 0.1 (s^k)^T D^k s^k \quad (3.9)$$

genügen. Der Vektor ω^k ist die Differenz der Gradienten der Lagrange-Funktion bei z^k und $z^k + s^k$, d.h.

$$\omega^k := \nabla_x f(z^k + s^k) - \nabla_x f(z^k) - \sum_{j=1}^m u_j^k (\nabla_x h(z^k + s^k) - \nabla_x h(z^k)), \quad (3.10)$$

mit dem Lagrange-Multiplikator $u^k \in \mathbb{R}^m$, $u_j^k \geq 0$ für $j \in M$. D^0 kann mit der Rang-n-Einheitsmatrix E_n initialisiert werden, d.h.

$$D^0 := E_n. \quad (3.11)$$

Im Schritt 2 des Modell-Algorithmus wird das Maß Q^k berechnet, mit dessen Hilfe die Größe des Trust-Region-Radius Δ^{k+1} für die folgende Iteration bestimmt wird. Q^k ist der Quotient der erwarteten Verbesserung und der erreichten Verbesserung der Merit-Funktion. Die erwartete Verbesserung ergibt sich aus der Veränderung der Zielfunktion $S : \mathbb{R}^n \rightarrow \mathbb{R}$ des Teilproblems und ist nach Konstruktion immer größer Null, ansonsten würde der Modell-Algorithmus im Schritt 1 terminieren. Die erreichte Verbesserung wird mit Hilfe der exakten Penalty-Funktion $P : \mathbb{R}^n \rightarrow \mathbb{R}$ bestimmt. Tritt eine Verschlechterung ein, d.h. $Q^k < 0$, dann wird der Trust-Region-Radius Δ^k verkleinert und das Teilproblem aus Definition 3.1.2 erneut gelöst. Für $Q^k \geq 0$ wird mit dem neuen Vektor $z^{k+1} := z^k + s^k$ fortgefahren. Der Wert von Q^k legt fest, wie Δ^k angepaßt wird.

Unter den beiden Voraussetzungen, daß f , h stetig differenzierbar und $\{x^k\}_{k=0,1,2,\dots}$, $\{D^k\}_{k=0,1,2,\dots}$ beschränkt sind, zeigt Yuan [65], daß dieser Modell-Algorithmus global konvergiert. Bleibt der Penalty-Parameter ζ^k beschränkt, dann konvergiert die

²steht für Broyden-Fletcher-Goldfarb-Shanno

Iterationsfolge $\{z^k\}_{k=0,1,2,\dots}$ gegen einen KKT-Punkt. Anderenfalls, d.h. für einen unbeschränkten Penalty-Parameter, zeigt Yuan [65], daß die Iterationsfolge gegen einen Fritz-John-Punkt konvergiert. In diesem Fall kann gezeigt werden, daß die Verletzung der Nebenbedingungen von OP beschränkt ist.

Definition 3.1.3 (Fritz-John-Bedingung)

Sei OP gegeben. Für $x \in \mathbb{R}^n$, $u_0 \in \mathbb{R}$ und $u \in \mathbb{R}^m$ heißt

$$u_0 \nabla_x f(x) + \sum_{j=1}^m u_j \nabla_x h_j(x) = 0, \quad (3.12)$$

$$u_0 \geq 0, \quad (3.13)$$

$$h_j(x) \leq 0, \quad j \in M, \quad (3.14)$$

$$u_j h_j(x) = 0, \quad j \in M, \quad (3.15)$$

$$u_j \geq 0, \quad j \in M, \quad (3.16)$$

Fritz-John-Bedingung. Ist die Fritz-John-Bedingung für (x, u_0, u) erfüllt, dann heißt (x, u_0, u) Fritz-John-Punkt.

Die Erfüllung der Fritz-John-Bedingung ist notwendig für einen stationären Punkt. Die Karush-Kuhn-Tucker-Bedingung hingegen ist hinreichend für den Nachweis eines stationären Punkts. Die Fritz-John-Bedingung entspricht für $u_0 \neq 0$ der Karush-Kuhn-Tucker-Bedingung.

Im folgenden Abschnitt wird die Trust-Region-Methode aus Algorithmus 3.1 mit der Trust-Region-Stabilisierung von TRSCP verglichen und Unterschiede und Parallelen aufgezeigt.

3.2 Der Algorithmus TRSCP

Der Algorithmus TRSCP basiert auf einem global konvergenten Lösungsverfahren für kontinuierliche, nichtlineare und restringierte Optimierungsprobleme, das durch eine Trust-Region-Methode stabilisiert wird. Dem Algorithmus liegt ein SCP-Verfahren zu Grunde, das eine MMA-Approximation (Definition 1.0.2) mit einer Trust-Region-Strategie kombiniert.

In diesem Abschnitt wird der Algorithmus TRSCP vorgestellt und im nächsten Abschnitt seine globale Konvergenzeigenschaft nachgewiesen. Die vom Algorithmus erzeugte Iterationsfolge besitzt mindestens einen Häufungspunkt und mindestens ein Häufungspunkt dieser Folge ist stationär. Diese Konvergenz ist unabhängig vom Startvektor. Bei TRSCP kommt eine Penalty-Funktion zum Einsatz. Ist der Penalty-Parameter dieser Funktion für diese Folge beschränkt, dann endet die Iterationsfolge in einem KKT-Punkt (Definition 2.4.1), oder mindestens ein Häufungspunkt der Folge erfüllt die KKT-Bedingung.

Für SCP-Verfahren werden, wie in Kapitel 2 vorgestellt, in jeder Iteration die Zielfunktion und die Restriktionsfunktionen von OP (Definition 2.3.1) in einem Vektor $z \in \mathbb{R}^n$ konvex approximiert. Mit Hilfe dieser Approximationen wird ein konvexes Teilproblem aufgestellt.

Der Vektor z heißt Entwicklungspunkt des Teilproblems. Wie beim SCP-Verfahren von Zillober [74] wird die Zielfunktion für das Teilproblem streng konvex approximiert. Hieraus folgt, daß das konvexe Teilproblem eine eindeutige Lösung $y \in \mathbb{R}^n$ besitzt, solange es einen nicht leeren zulässigen Bereich hat. In Kapitel 4 wird ausgeführt, wie selbst für ein Teilproblem mit leerem zulässigen Bereich ein eindeutiger Vektor gefunden wird, der als Iterationspunkt herangezogen werden kann. Deshalb wird als Vereinfachung im Weiteren davon ausgegangen, daß immer eine eindeutige Lösung des Teilproblems vorliegt.

Mit der Lösung $y \in \mathbb{R}^n$ des Teilproblems soll der nächste Iterationspunkt, d.h. der Entwicklungspunkt des nächsten Teilproblems, bestimmt werden. Das bedeutet, der Algorithmus erzeugt eine Folge von Entwicklungspunkten. Hierfür muß y hinsichtlich des verfolgten Ziels, einen stationären Punkt von OP zu approximieren, bewertet werden. Analog dem Verfahren von Zillober [74] kommt die erweiterte Lagrange-Funktion $\Phi_r : \mathbb{R}^{n+m} \rightarrow \mathbb{R}$ (Definition 2.4.4) zum Einsatz, um eine Abstiegsrichtung zu definieren. Die erweiterte Lagrange-Funktion Φ_r ordnet jeder primalen Variablen $x \in \mathbb{R}^n$ für einen Penalty-Parameter $r \in \mathbb{R}^m$ und einer dualen Variablen $u \in \mathbb{R}^m$ einen Wert zu. Die dualen Variablen werden auch Lagrange-Multiplikatoren genannt. Die Anzahl der Nebenbedingungsfunktionen legt die Dimension m des Penalty-Parameters fest.

Nicht jede Iteration eines Optimierungsverfahrens mit Trust-Region-Stabilisierung führt zu einem Wechsel des Entwicklungspunkts. Es werden hier innere und äußere Iterationen unterschieden. Die Lösung des Teilproblems muß gewissen Anforderungen genügen. Erfüllt diese Lösung die Anforderungen nicht, wird eine innere Iteration durchgeführt. Bei einer inneren Iteration wird der Entwicklungspunkt nicht

geändert, sondern die aktuelle Lösung des Teilproblems verworfen. Es wird nur der Trust-Region-Radius verkleinert und das Teilproblem anschließend erneut gelöst. Bei einer äußeren Iteration wird die aktuelle Lösung des Teilproblems als Entwicklungspunkt für das nächste Teilproblem akzeptiert, d.h. es findet ein Wechsel des Entwicklungspunkts statt. Hier kann im Anschluß der Trust-Region-Radius angepaßt werden, indem der Trust-Region-Radius vergrößert oder verkleinert wird. Für eine unendliche von TRSCP erzeugte Folge von Iterationspunkten ist zu zeigen, daß jede Teilfolge von inneren Iterationen endlich ist. Das heißt, für den Nachweis der globalen Konvergenzeigenschaft von TRSCP wird es danach ausreichend sein, die unendliche Teilfolge von äußeren Iterationen zu betrachten.

Bei MMA-Verfahren werden für einen Entwicklungspunkt $z \in \mathbb{R}^n$, für den das Teilproblem aufgestellt werden soll, eine untere Asymptote $L \in \mathbb{R}^n$ und eine obere Asymptote $U \in \mathbb{R}^n$ festgelegt. Mit ihrer Hilfe werden die Approximationen für das Teilproblem bestimmt. Es gilt

$$L_i < z_i < U_i, \quad \forall i = 1, \dots, n. \quad (3.17)$$

Jeder Koeffizient der Lösung $y \in \mathbb{R}^n$ des Teilproblems wird ebenfalls von dem entsprechenden Paar von Asymptotenkoeffizienten eingeschlossen, d.h. es gilt

$$L_i < y_i < U_i, \quad \forall i = 1, \dots, n. \quad (3.18)$$

Ein Teilbereich zwischen diesen Asymptoten wird bei TRSCP als Trust-Region angesehen. Durch Steuerung der Asymptotenerzeugung im Sinne einer Trust-Region-Methode wird globale Konvergenz des Algorithmus erreicht.

Würde der gesamte Bereich zwischen den Asymptoten als Trust-Region angesehen und erfolgt die Asymptotenerzeugung durch Addition eines Trust-Region-Radius $\Delta \in \mathbb{R}^n$ mit $\Delta_i > 0$, $i = 1, \dots, n$, zu $z \in \mathbb{R}^n$, könnte kein konstanter Mindestabstand zwischen z und den Asymptoten garantiert werden. Folglich wären diese Asymptoten nach Definition 2.3.2 nicht zulässig. Um die Verletzung der Zulässigkeit zu vermeiden wird ein konstanter Mindestabstand $\xi > 0$ zwischen dem Entwicklungspunkt und den Asymptoten eingeführt. Durch den Mindestabstand zwischen Entwicklungspunkt und Asymptote wird vermieden, daß die Werte der Approximationen bzw. die Werte ihrer Ableitungen unbeschränkt ansteigen. Abbildung 3.2 zeigt den Zusammenhang von Funktion f , Entwicklungspunkt z , Trust-Region-Radius Δ , Mindestabstand ξ , den Asymptoten $L(z, \Delta, \xi)$ und $U(z, \Delta, \xi)$ und der daraus resultierenden Approximation f^k , mit dem Iterationsindex $k \in \mathbb{N}_0$.

Es seien ein Entwicklungspunkt $z \in \mathbb{R}^n$, ein Trust-Region-Radius $\Delta \in \mathbb{R}^n$ mit $\Delta_i > 0$ für $i = 1, \dots, n$ und die Konstante $\xi > 0$ als Mindestabstand gegeben. Für den Mindestabstand ξ wird der Vektor

$$\Xi := (\xi, \dots, \xi) \in \mathbb{R}^n \quad (3.19)$$

definiert. Die untere und die obere Asymptote werden im Algorithmus TRSCP durch

$$\begin{aligned} L(z, \Delta, \xi) &:= z - (\Delta + \Xi), \\ U(z, \Delta, \xi) &:= z + (\Delta + \Xi) \end{aligned} \quad (3.20)$$

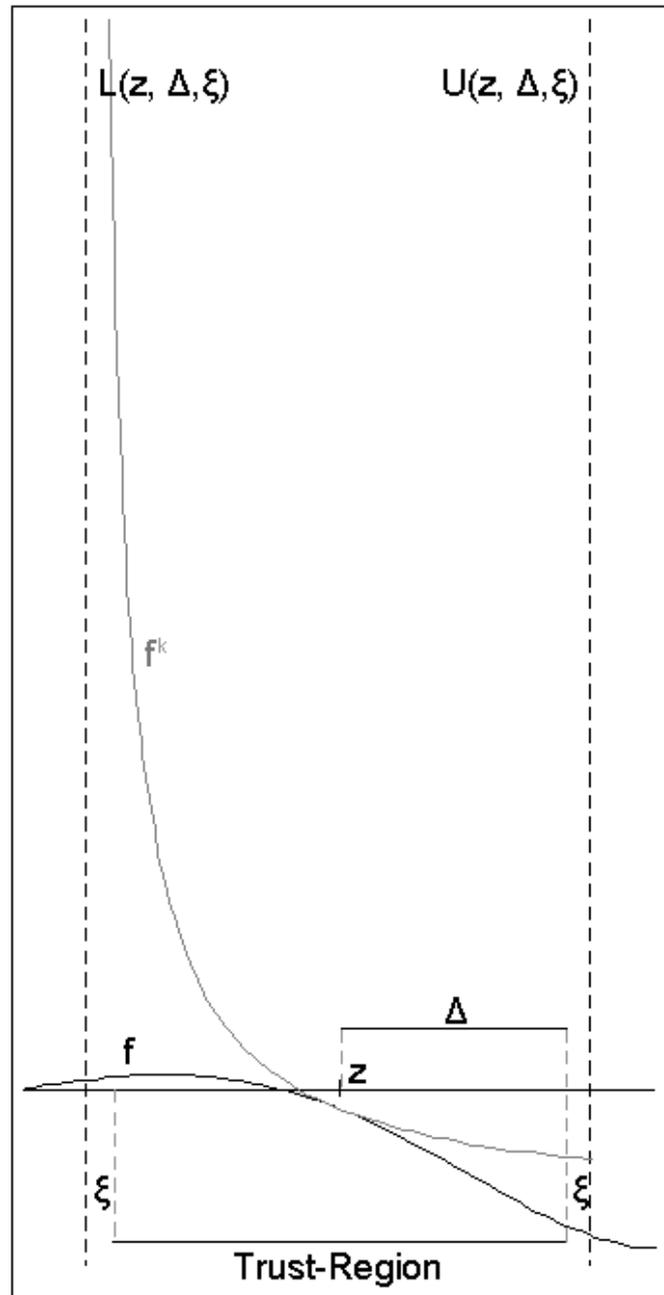


Abbildung 3.1: Trust-Region, zulässige Asymptoten, Zielfunktion und Approximation für einen Entwicklungspunkt z

festgelegt. Der Vektor $\Delta \in \mathbb{R}^n$ ist der Trust-Region-Radius im Algorithmus TRSCP.

Im Unterschied zum Modell-Algorithmus 3.1 von Yuan, ist der Trust-Region-Radius bei TRSCP mehrdimensional. Hieraus folgt, daß jeder Koeffizient des Trust-Region-Radius individuell geeignet angepaßt werden kann.

Lemma 3.2.1

Es seien $\Delta_{\max} > 0$, $\xi > 0$ und $b > 0$ gegeben. Ferner sei eine Folge von Entwicklungspunkten $\{z^k\}_{k=0,1,2,\dots}$ mit $|z_i^k| \leq b$, $i = 1, \dots, n$, und eine Folge von Trust-Region-Radien $\{\Delta^k\}_{k=0,1,2,\dots}$ mit $0 < \Delta_i^k \leq \Delta_{\max}$, $i = 1, \dots, n$, gegeben. Dann ist $\{L(z^k, \Delta^k, \xi), U(z^k, \Delta^k, \xi)\}_{k=0,1,2,\dots}$ eine Folge von zulässigen Asymptoten.

Beweis:

Die Einhaltung eines Mindestabstands ξ von den Asymptoten analog Definition 2.3.2 folgt aus $\Delta_i^k > 0$ für $i = 1, \dots, n$, $\xi > 0$ und (3.20). Aufgrund von (3.20) und der Beschränktheit von z^k ist die Beschränktheit der Folge von Asymptoten mit den Konstanten

$$\begin{aligned} L_{\min} &:= -(b + \Delta_{\max} + \xi) > -\infty, \\ U_{\max} &:= b + \Delta_{\max} + \xi < \infty, \end{aligned} \quad (3.21)$$

gegeben. Damit gilt

$$L_{\min} \leq L(z^k, \Delta^k, \xi)_i < z_i^k - \xi < z_i^k < z_i^k + \xi < U(z^k, \Delta^k, \xi)_i \leq U_{\max}, \quad (3.22)$$

für alle $k \in \mathbb{N}_0$ und $i = 1, \dots, n$.

q.e.d.

Die Konstante Δ_{\max} ist der maximale Betrag eines Koeffizienten eines Trust-Region-Radius, und wird im weiteren vereinfachend als maximaler Trust-Region-Radius bezeichnet. $\Delta_{\max} > 0$ kann willkürlich gewählt werden. Sind Schranken im Optimierungsproblem analog (1.9) gegeben, ist

$$\Delta_{\max} := \|\bar{x} - \underline{x}\|_{\infty} \quad (3.23)$$

eine Möglichkeit diese Konstante zu setzen.

Nicht alle Koeffizienten Δ_i eines Trust-Region-Radius $\Delta \in \mathbb{R}^n$ müssen den gleichen Wert besitzen. Hilfreich ist es, den kleinsten Betrag der Koeffizienten Δ_i zu bezeichnen. Es sei

$$\Delta_{\min} := \min_{i=1,\dots,n} \Delta_i \quad (3.24)$$

der kleinste Betrag der Koeffizienten des Trust-Region-Radius Δ .

Die Konstante Δ_{\max} ist unabhängig vom Iterationsparameter k . Im Gegensatz hierzu ist der kleinste Betrag der Koeffizienten des Trust-Region-Radius abhängig von k . Das heißt zu einer Folge $\{\Delta^k\}_{k=0,1,2,\dots}$ von Trust-Region-Radien gehört die Folge $\{\Delta_{\min}^k\}_{k=0,1,2,\dots} \subset \mathbb{R}$.

Wird ein konvexes Teilproblem mit den Asymptoten aus (3.20) aufgestellt und gelöst, liegt die Lösung y des Teilproblems im offenen Intervall $(L(z, \Delta, \xi); U(z, \Delta, \xi))$, aber nicht notwendigerweise in $[z - \Delta; z + \Delta]$. Das bedeutet, die Trust-Region-Bedingung kann durch die Einführung des konstanten Mindestabstands $\xi > 0$ verletzt werden. Bei TRSCP wird daher sicherzustellen sein, daß die Lösung eines Teilproblems trotz dieser Asymptotenerzeugung innerhalb des Trust-Region-Radius liegt. Das heißt,

$$|y_i - z_i| \leq \Delta_i, \quad i = 1, \dots, n, \quad (3.25)$$

muß erfüllt sein.

Durch die Asymptotenerzeugung (3.20) erhalten die Funktionen eines konvexen Teilproblems $SP(z^k, L(z^k, \Delta, \xi), U(z^k, \Delta, \xi))$ folgende Form.

Lemma 3.2.2

Sei OP , eine Folge von Entwicklungspunkten $\{z^k\}_{k=0,1,2,\dots}$, eine Folge von Trust-Region-Radien $\{\Delta^k\}_{k=0,1,2,\dots}$ mit $\Delta_i^k > 0$, $i = 1, \dots, n$, und die Konstanten $\xi > 0$ und $\tau > 0$ gegeben. Dann gilt für die Funktionswerte der konvexen Teilprobleme $SP(z^k, L(z^k, \Delta, \xi), U(z^k, \Delta, \xi))$ (siehe Definition 2.3.3),

$$\tilde{f}^k(x, z^k, L(z^k, \Delta^k, \xi), U(z^k, \Delta^k, \xi), \tau) = f^k(x, z^k, \Delta^k, \xi, \tau), \quad (3.26)$$

$$\begin{aligned} \tilde{h}_j^k(x, z^k, L(z^k, \Delta^k, \xi), U(z^k, \Delta^k, \xi)) &= h_j^k(x, z^k, \Delta^k, \xi), \\ j &\in M, \end{aligned} \quad (3.27)$$

mit

$$\begin{aligned} f^k(x, z^k, \Delta^k, \xi, \tau) &:= \alpha_{0,0}^k(z^k, \Delta^k, \xi) + \sum_{i \in I_0^+(z^k)} \frac{\alpha_{0,i}^k(x_i, z^k, \Delta_i^k, \xi, \tau)}{(z_i^k + \Delta_i^k + \xi) - x_i} \\ &+ \sum_{i \in I_0^-(z^k)} \frac{\alpha_{0,i}^k(x_i, z^k, \Delta_i^k, \xi, \tau)}{x_i - (z_i^k - \Delta_i^k - \xi)}, \end{aligned} \quad (3.28)$$

$$\begin{aligned} h_j^k(x, z^k, \Delta^k, \xi) &:= \alpha_{j,0}^k(z^k, \Delta^k, \xi) + \sum_{i \in I_j^+(z^k)} \frac{\alpha_{j,i}^k(z_i^k, \Delta_i^k, \xi)}{(z_i^k + \Delta_i^k + \xi) - x_i} \\ &+ \sum_{i \in I_j^-(z^k)} \frac{\alpha_{j,i}^k(z_i^k, \Delta_i^k, \xi)}{x_i - (z_i^k - \Delta_i^k - \xi)}, \end{aligned} \quad (3.29)$$

und

$$\alpha_{0,0}^k(z^k, \Delta^k, \xi) := f(z^k) - \sum_{i=1}^n \left| \frac{\partial f(z^k)}{\partial x_i} \right| (\Delta_i^k + \xi), \quad (3.30)$$

$$\alpha_{0,i}^k(x_i, z^k, \Delta_i^k, \xi, \tau) := \left| \frac{\partial f(z^k)}{\partial x_i} \right| (\Delta_i^k + \xi)^2 + \tau(x_i - z_i^k)^2, \quad (3.31)$$

und

$$\alpha_{j,0}^k(z^k, \Delta^k, \xi) := h_j(z^k) - \sum_{i=1}^n \left| \frac{\partial h_j(z^k)}{\partial x_i} \right| (\Delta_i^k + \xi), \quad (3.32)$$

$$\alpha_{j,i}^k(z^k, \Delta_i^k, \xi) := \left| \frac{\partial h_j(z^k)}{\partial x_i} \right| (\Delta_i^k + \xi)^2. \quad (3.33)$$

Beweis:

Es ist

$$\begin{aligned} & z^k - L(z^k, \Delta^k, \xi) \\ &= z^k - (z^k - \Delta^k - \xi) \\ &= \Delta^k + \xi \\ &= (z^k + \Delta^k + \xi) - z^k \\ &= U(z^k, \Delta^k, \xi) - z^k. \end{aligned} \quad (3.34)$$

Weiter gilt $I_0^-(z^k) \cup I_0^+(z^k) = \{1, \dots, n\}$ (siehe (2.10)) und

$$\begin{aligned} & - \sum_{i \in I_0^+(z^k)} \frac{\partial f(z^k)}{\partial x_i} + \sum_{i \in I_0^-(z^k)} \frac{\partial f(z^k)}{\partial x_i} \\ &= - \sum_{i \in I_0^+(z^k)} \frac{\partial f(z^k)}{\partial x_i} - \sum_{i \in I_0^-(z^k)} \left| \frac{\partial f(z^k)}{\partial x_i} \right| \\ &= - \sum_{i=1, \dots, n} \left| \frac{\partial f(z^k)}{\partial x_i} \right| \end{aligned} \quad (3.35)$$

(analog für Nebenbedingungsfunktionen). Insgesamt folgt die Äquivalenz der Funktionen der Teilprobleme.

q.e.d.

Bei Zillober [74] wird im Teilproblem $SP(z^k, L^k, U^k)$ für ein OP mit Schranken analog (1.9) eine Teilmenge $\check{X}^k \subseteq \mathbb{R}^n$ bestimmt, bei der ein Parameter $\omega \in (0; 1)$ benötigt wird. \check{X}^k ist so definiert, daß die Lösung $y^k \in \mathbb{R}^n$ des Teilproblems einen gewissen, allerdings im Verlauf der Iteration variablen Abstand von den Asymptoten hat. Außerdem werden die unteren und oberen Schranken so für das Teilproblem angepaßt, daß sie zwischen dem Entwicklungspunkt und den Asymptoten liegen. Bei TRSCP wird eine Menge X^k so bestimmt (siehe 3.37), daß sie ähnliche Eigenschaften besitzt wie \check{X}^k bei Zillober [74] bzw. \tilde{X}^k aus Kapitel 2 (siehe 2.20). Bei TRSCP können, falls vorhanden, in die Menge X^k auch Schranken für die primalen Variablen aufgenommen werden (siehe Kapitel 4). Allerdings wird im folgenden X^k ohne solche Schranken definiert. Bei TRSCP wird X^k so gewählt, daß die Lösung y^k sicher ein Element der Trust-Region ist.

Definition 3.2.3

Sei eine Folge von Entwicklungspunkten $\{z^k\}_{k=0,1,2,\dots}$ und eine Folge von Trust-Region-Radien $\{\Delta^k\}_{k=0,1,2,\dots}$ mit $\Delta_i^k > 0$, $i = 1, \dots, n$, gegeben. Dann ist

$$X(z^k, \Delta^k) := \{x \in \mathbb{R}^n : z^k - \Delta^k \leq x \leq z^k + \Delta^k, i = 1, \dots, n\}. \quad (3.36)$$

Ein Übertrag der Ergebnisse aus Zillober [74] auf TRSCP ist mit

$$X^k := X(z^k, \Delta^k) \quad (3.37)$$

anstelle von \check{X}^k möglich, da beide Mengen den gleichen Zweck erfüllen, die primale Variable von den Asymptoten wegzubeschränken.

In der Darstellung der Funktionen in den Teilproblemen von TRSCP wird die Symmetrie der Asymptoten zum Entwicklungspunkt ausgenutzt und dadurch vereinfacht. Außerdem werden die Nebenbedingungen für die Sicherstellung der Trust-Region explizit in das Teilproblem aufgenommen, d.h mit

$$\begin{aligned} g_i^k(x, z^k, \Delta^k) &:= -x_i + z_i^k - \Delta_i^k \leq 0, \\ g_{n+i}^k(x, z^k, \Delta^k) &:= x_i - z_i^k - \Delta_i^k \leq 0, \end{aligned} \quad (3.38)$$

für $i = 1, \dots, n$ wird die Bedingung $x \in X^k := X(z^k, \Delta^k)$ ersetzt. Damit sind die Teilprobleme von TRSCP gegeben durch:

Definition 3.2.4 (Konvexes Teilproblem)

Sei OP , eine Folge von Entwicklungspunkten $\{z^k\}_{k=0,1,2,\dots}$, eine Folge von Trust-Region-Radien $\{\Delta^k\}_{k=0,1,2,\dots}$ und die Konstanten $\xi > 0$ und $\tau > 0$ gegeben. Dann ist das Problem $\mathbf{TRSP}(z^k, \Delta^k)$ gegeben durch

$$\begin{aligned} &\text{Minimiere } f^k(x, z^k, \Delta^k, \xi, \tau), \\ &\text{unter } h_j^k(x, z^k, \Delta^k, \xi) \leq 0, \quad j \in M, \\ &g_i^k(x, z^k, \Delta^k) \leq 0, \quad i = 1, \dots, 2n, \\ &x \in \mathbb{R}^n \end{aligned} \quad (3.39)$$

und heißt konvexes Teilproblem von TRSCP in z^k des nichtlinearen Optimierungsproblems OP .

Für die Teilprobleme $TRSP(z^k, \Delta^k)$ gelten wegen Lemma 3.2.2 die gleichen Aussagen, die für die Teilprobleme $SP(z^k, L^k, U^k)$ getroffen wurden. Das heißt

- $f^k(x, z^k, \Delta^k, \xi, \tau)$ und $h_j^k(x, z^k, \Delta^k, \xi)$, $j \in M$, sind Approximationen erster Ordnung von $f(x)$ bzw. $h_j(x)$,
- die Funktion $f^k : \mathbb{R}^n \rightarrow \mathbb{R}$ ist streng konvex,

- die Funktionen $h_j^k : \mathbb{R}^n \rightarrow \mathbb{R}$, $j \in M$, sind konvex,
- die Funktionen f^k , h_j^k , $j \in M$, und $g_i^k(x, z^k, \Delta^k)$, $i = 1, \dots, 2n$, sind separabel.

Für eine Iteration $k \in \mathbb{N}_0$ ist der zulässige Bereich eines konvexen Teilproblems von OP im Entwicklungspunkt $z^k \in \mathbb{R}^n$ mit dem Trust-Region-Radius $\Delta^k \in \mathbb{R}^n$ und dem Parameter ξ

$$\Omega(z^k, \Delta^k, \xi) := \{x \in X(z^k, \Delta^k) : h_j^k(x, z^k, \Delta^k, \xi) \leq 0, j \in M\}. \quad (3.40)$$

Dieser zulässige Bereich ist konvex und kompakt. Falls $\Omega(z^k, \Delta^k, \xi)$ nicht leer ist, hat das Teilproblem $TRSP(z^k, \Delta^k)$ aufgrund der streng konvexen Zielfunktion $f^k : \mathbb{R}^n \rightarrow \mathbb{R}$ eine eindeutige Lösung $y^k \in \mathbb{R}^n$.

Für die Lösung eines Teilproblems $TRSP(z^k, \Delta^k)$ gilt aufgrund der Definition von $X^k := X(z^k, \Delta^k)$ bzw. der Nebenbedingungen $g_i : \mathbb{R}^n \rightarrow \mathbb{R}$, $i = 1, \dots, 2n$,

$$y^k \in [z^k - \Delta^k; z^k + \Delta^k], \quad (3.41)$$

d.h. die Trust-Region-Eigenschaft von TRSCP ist erfüllt.

Bei Trust-Region-Methoden wird im Allgemeinen in jeder Iteration eine Prognose erstellt, inwiefern der neu bestimmte Kandidat für einen Iterationspunkt sich in einem gewissen Sinn gegenüber dem aktuellen Iterationspunkt verbessert. Dieses Ergebnis wird mit der tatsächlich erreichten Verbesserung verglichen. Für die Prognose wird der Abstieg linear abgeschätzt, indem die Ableitung der Merit-Funktion mit der Suchrichtung multipliziert wird. Die Suchrichtung in der Iteration $k \in \mathbb{N}_0$ ist

$$s^k := \begin{pmatrix} y^k - z^k \\ v^k - u^k \end{pmatrix}, \quad (3.42)$$

mit dem Entwicklungspunkt $z^k \in \mathbb{R}^n$, den Lagrange-Multiplikatoren u^k und der Lösung des Teilproblems (y^k, v^k, w^k) mit $y^k \in \mathbb{R}^n$, $v^k \in \mathbb{R}^m$ und $w^k \in \mathbb{R}^{2n}$. Das heißt

$$\nabla \Phi_r \begin{pmatrix} z^k \\ u^k \end{pmatrix}^T s^k \quad (3.43)$$

wird berechnet. Bevor diese Prognose erstellt werden kann, muß der Penalty-Parameter $r \in \mathbb{R}^m$ angepaßt werden, so daß die Suchrichtung eine Abstiegsrichtung für die erweiterte Lagrange-Funktion (Definition 2.4.4) ist. Die tatsächlich erreichte Verbesserung wird über

$$\Phi_r \begin{pmatrix} y^k \\ v^k \end{pmatrix} - \Phi_r \begin{pmatrix} z^k \\ u^k \end{pmatrix} \quad (3.44)$$

bestimmt. Aufgrund des Verhältnisses von Prognose und erreichter Verbesserung wird im Allgemeinen der Trust-Region-Radius für die nächste Iteration angepaßt. Bei TRSCP ist diese Anpassung nicht notwendig um die globale Konvergenzeigenschaft nachzuweisen. In diesem Kapitel wird eine Basisversion von TRSCP vorgestellt, die eine globale Konvergenzeigenschaft besitzt, aber auf nicht notwendige Änderungen des Trust-Region-Radius verzichtet.

Es folgt eine Beschreibung des Algorithmus Trust-Region Sequential Convex Programming (TRSCP). Er besteht aus vier Schritten, mit einem zusätzlichen Initialisierungsschritt zu Beginn. Im ersten Schritt wird das konvexe Teilproblem für die aktuelle Näherung der Lösung aufgestellt und gelöst. Ist die Lösung des Teilproblems identisch mit der aktuellen Näherung, ist ein stationärer Punkt von OP gefunden worden (Lemma 2.4.3) und die Iteration wird beendet. Im zweiten Schritt werden die Penalty-Parameter der erweiterten Lagrange-Funktion mit Hilfe einer Teiliteration so angepaßt, daß die Suchrichtung zu einer hinreichenden Abstiegsrichtung für die erweiterte Lagrange-Funktion wird. Im dritten Schritt wird die erwartete Verbesserung im Vergleich zum Ausgangsproblem bestimmt und das Ergebnis des Teilproblems hinsichtlich des erzielten Abstiegs überprüft. Im letzten Schritt wird die Lösung des Teilproblems als neue Näherung für einen stationären Punkt von OP akzeptiert und mit der nächsten Iteration begonnen.

Bei iterativen Trust-Region Verfahren wird zwischen inneren und äußeren Iterationen unterschieden. Als innere Iteration wird der Fall bezeichnet bei dem nur die Trust-Region verkleinert, aber keine neue Näherung eines stationären Punkts akzeptiert wird. Als äußere Iteration wird der Fall bezeichnet in dem die Lösung des Teilproblems als neue Näherung akzeptiert und der Trust-Region-Radius entsprechend dem berechneten Quotienten angepaßt wird. Bei der hier vorgestellten Basisversion von TRSCP besteht bei äußeren Iterationen nicht die Möglichkeit, den Trust-Region-Radius zu ändern, außer ihn wieder auf seinen Wert wie zu Beginn der Iteration zu setzen.

Der Teiliterationsindex für innere Iterationen wird bei TRSCP mit q bezeichnet. Der Teiliterationsindex für die Anpassung des Penalty-Parameters wird p sein. Streng genommen müßten die Penalty-Parameter nicht nur mit dem Iterationsindex k und dem Teiliterationsindex p indiziert werden, sondern auch noch mit q . Um die Lesbarkeit des Algorithmus zu gewährleisten wird allerdings auf die Indizierung des Penalty-Parameters mit q verzichtet, was im weiteren keine Unklarheiten erzeugt.

Algorithmus 3.2 (Trust-Region Sequential Convex Programming)

Sei OP gegeben.

Schritt 0: Initialisierung:

Wähle einen Entwicklungspunkt $z^0 \in \mathbb{R}^n$ und einen Lagrange-Multiplikator $u^0 \in \mathbb{R}^m$ mit $u_j^0 \geq 0$ für $j \in M$. Wähle einen Penalty-Parameter $r_0^{-1} \in \mathbb{R}^m$ mit $r_{0,j}^{-1} \geq 1$ für $j \in M$ und Konstanten $\tau > 0$, $\kappa > 1$, $\xi > 0$, $\Delta_{\max} > 0$. Berechne $f(z^0)$, $\nabla_x f(z^0)$ und $h_j(z^0)$, $\nabla_x h_j(z^0)$ für $j \in M$. Ferner sei

$$\begin{aligned} 0 &< \Delta_{0,i}^0 \leq \Delta_{\max}, \quad i = 1, \dots, n, \\ 0 &< \nu < 1, \\ 0 &< \gamma \ll 1. \end{aligned}$$

Initialisiere $k := 0$ und $q := 0$.

Schritt 1: Aufstellen und Lösen des Teilproblems:

Bestimme das Optimum (y_q^k, v_q^k, w_q^k) des konvexen Teilproblems $TRSP(z^k, \Delta_q^k)$.

Ist $z^k = y_q^k$, dann ist $\begin{pmatrix} x^* \\ u^* \end{pmatrix} := \begin{pmatrix} z^k \\ u^k \end{pmatrix}$ stationärer Punkt von OP, STOP.

Ansonsten setze $s_q^k := \begin{pmatrix} y_q^k - z^k \\ v_q^k - u^k \end{pmatrix}$ und $\delta_q^k := \|y_q^k - z^k\|_2$. Initialisiere $p := 0$.

Schritt 2: Anpassen des Penalty-Parameters:

Falls

$$\nabla \Phi_{r_p^{k-1}} \begin{pmatrix} z^k \\ u^k \end{pmatrix}^T s_q^k > -\frac{\tau(\delta_q^k)^2}{4(\Delta_{\max} + \xi)}, \quad (3.45)$$

setze $r_{p+1}^{k-1} := \kappa r_p^{k-1}$, $p := p + 1$ und wiederhole *SCHRITT 2*.

Anderenfalls setze $r_0^k := r_p^{k-1}$ und berechne $f(y^k)$ und $h_j(y^k)$, $j \in M$.

Schritt 3: Überprüfung des Abstiegs:

Falls

$$\Phi_{r_0^k} \begin{pmatrix} y_q^k \\ v_q^k \end{pmatrix} - \Phi_{r_0^k} \begin{pmatrix} z^k \\ u^k \end{pmatrix} > \gamma \nabla \Phi_{r_0^k} \begin{pmatrix} z^k \\ u^k \end{pmatrix}^T s_q^k, \quad (3.46)$$

verkleinere den Trust-Region-Radius durch

$$\Delta_{q+1}^k := \nu \Delta_q^k, \quad (3.47)$$

berechne $g_i(z^k)$ für $i = 1, \dots, 2n$, setze $q := q + 1$ und gehe zu *SCHRITT 1*.

Schritt 4: Wechsel des Entwicklungspunkts:

Setze $z^{k+1} := y_q^k$, $u^{k+1} := v_q^k$ und $\Delta_0^{k+1} := \Delta_0^0$. Berechne $\nabla_x f(z^{k+1})$ und $\nabla_x h_j(z^{k+1})$ für $j \in M$, setze $k := k + 1$, $q := 0$ und gehe zu *SCHRITT 1*.

Das exakte Abbruchkriterium im ersten Schritt kann für die Implementierung folgendermaßen abgeschwächt werden. Der Algorithmus terminiert, wenn eine Lösung (y_q^k, v_q^k, w_q^k) gefunden wurde, die mit einer vorher festzulegenden Genauigkeit die Stationaritätskriterien erfüllt. Die Umsetzung dieses Algorithmus wird in Kapitel 4 erläutert.

Die Vorgehensweise von TRSCP ist vergleichbar mit dem im Abschnitt 3.1 vorgestellten Modell-Algorithmus 3.1 von Yuan [65]. Beide Algorithmen werden im Schritt 0 initialisiert. Es werden eine Ausgangsschätzung (z^0, u^0) für einen stationären Punkt von OP und weitere für das jeweilige Verfahren notwendige Parameter und Hilfsmittel festgelegt.

In Schritt 1 von Algorithmus 3.1 und TRSCP wird das Teilproblem aufgestellt und gelöst. Der Modell-Algorithmus aus Yuan [65] terminiert, wenn der Nullvektor als Lösung des Teilproblems $TP(z^k, \Delta^k, \zeta^k)$ identifiziert wurde. Der Algorithmus TRSCP terminiert, falls y_q^k als Teil der Lösung des Teilproblems $TRSP(z^k, \Delta_q^k)$ identisch mit dem Entwicklungspunkt z^k ist. Diese beiden Abbruchkriterien sind äquivalent, da in beiden Fällen die Näherung z^k , für den die Teilprobleme aufgestellt wurden, als Entwicklungspunkt im folgenden Iterationsschritt vorgeschlagen

werden würde. Identifiziert TRSCP den Entwicklungspunkt z^k als Teil der Lösung des Teilproblems $TRSP(z^k, \Delta_q^k)$, dann folgt, daß in diesem Fall die Trust-Region nicht aktiv sein kann, d.h. es ist $w_{0,i}^k = 0$, $i = 1, \dots, 2n$, garantiert.

Schritt 2 im Algorithmus TRSCP stellt sicher, daß der Penalty-Parameter der erweiterten Lagrange-Funktion so groß ist, daß die gefundene Suchrichtung bezüglich der Merit-Funktion einen Abstieg erzielt. Die Bestimmung des hinreichend großen Penalty-Parameters erfolgt über eine endliche Teiliteration (siehe Abschnitt 3.3). Im dritten Schritt von TRSCP wird überprüft, ob mit der Lösung des Teilproblems eine ausreichende Reduzierung des erweiterten Lagrange-Funktionswerts erzielt wird. Ist dies nicht der Fall, wird eine innere Iteration durchgeführt und der Trust-Region-Radius verkleinert.

In Schritt 2 von Algorithmus 3.1 wird der Quotient Q^k aus erwarteter Verbesserung und erreichter Verbesserung berechnet. Ist dieser Quotient negativ, d.h. die Lösung des Teilproblems liefert einen schlechteren Wert als die alte Schätzung, wird ebenfalls eine innere Iteration durchgeführt. Die beiden Algorithmen unterscheiden sich dahingehend, daß bei TRSCP aufgrund der vorherigen Abfragen kein negativer Quotient Q^k auftreten könnte. Allerdings wird in der in diesem Kapitel vorgestellten Version von TRSCP keine Anpassung des Trust-Region-Radius vorgenommen und deshalb auch kein Quotient Q^k berechnet.

In Schritt 4 von TRSCP wird (y_q^k, v_q^k) als neue Näherung (z^{k+1}, u^{k+1}) für einen stationären Punkt von OP festgelegt und der Trust-Region-Radius falls notwendig auf den Wert des Start-Radius angehoben. Im Algorithmus von Yuan [65] wird, nachdem die Lösung des Teilproblems positiv bewertet wurde, mit dieser Lösung eine neue Näherung z^{k+1} als Entwicklungspunkt für den nächsten Schritt festgelegt und die Trust-Region mit Hilfe des berechneten Quotienten angepaßt.

Wurde die Lösung des Teilproblems als nächste Näherung übernommen und der Trust-Region-Radius angepaßt, dann ist eine Iteration bei TRSCP erfolgreich beendet und es wird mit Schritt 1 fortgefahren. Im Algorithmus 3.1 ist noch Schritt 4 abzuarbeiten, in dem der dort verwendete Penalty-Parameter angepaßt wird. Anschließend ist auch hier eine Iteration beendet und die nächste Iteration wird mit Schritt 1 begonnen.

Außer beim Nachweis, daß jede innere Iteration von TRSCP endlich ist, ist bei den folgenden Beweisen der Teiliterationsindex q uninteressant. Für eine Vereinfachung der Notation wird in diesen Fällen auf eine Indizierung mit q verzichtet. Das heißt, der Trust-Region-Radius wird im Folgenden zumeist mit Δ^k , die Lösung des Teilproblems mit (y^k, v^k, w^k) , die Suchrichtung mit s^k und der Abstand zwischen Entwicklungspunkt und Teilproblemlösung mit δ^k bezeichnet.

3.3 Globale Konvergenz von TRSCP

In diesem Abschnitt wird die globale Konvergenzeigenschaft von TRSCP gezeigt. Es wird nachgewiesen, daß eine unendliche, von TRSCP erzeugte Folge mindestens

einen Häufungspunkt besitzt und, daß mindestens ein Häufungspunkt dieser Folge stationär ist. Das heißt mit einem Häufungspunkt wird die Karush-Kuhn-Tucker-Bedingung (Definition 2.4.1) für das nichtlineare Optimierungsproblem OP erfüllt.

Die globale Konvergenz des Algorithmus TRSCP wird auf ähnliche Weise bewiesen, wie in Zillober [74] die globale Konvergenz für den SCP-Algorithmus 2.4 bewiesen wird. Es sind folgende Eigenschaften von TRSCP zu zeigen:

1. Die Teiliteration für die Anpassung des Penalty-Parameters der erweiterten Lagrange-Funktion ist endlich.
2. Die Teiliteration zur Überprüfung des Abstiegs (innere Iteration) ist endlich.
3. Die von TRSCP erzeugte Folge $\{(z^k, u^k)\}_{k=0,1,2,\dots}$ von Iterationspunkten ist beschränkt.
4. Jedes Folgenglied einer von TRSCP erzeugten Folge $\{s^k\}_{k=0,1,2,\dots}$ von Suchrichtungen bewirkt einen für die globale Konvergenz von TRSCP hinreichenden Abstieg.

Als erstes wird nachgewiesen, daß die Teiliterationen zur Anpassung des Penalty-Parameters und zur Durchführung von inneren Iterationen endlich sind. Danach kann in den Beweisen in diesem Abschnitt davon ausgegangen werden, daß TRSCP eine unendliche Folge $\{(z^k, u^k)\}_{k=0,1,2,\dots}$ von Iterationspunkten erzeugt. Da ein Abbruch der Iteration im Algorithmus 3.2 nur erfolgt, wenn ein stationärer Punkt von OP gefunden wurde, ist in diesem Fall nichts zu zeigen. Die im Kapitel 2.4 eingeführte Kennzeichnung eines stationären Punkts von OP findet auch in diesem Abschnitt Anwendung.

Bezeichnung 3.3.1

Gegeben sei OP und ein Trust-Region-Radius $\Delta^* \in \mathbb{R}^n$ mit $\Delta_i^* > 0$, $i = 1, \dots, n$. $(x^*, u^*) \in \mathbb{R}^{n+m}$ bezeichnet einen stationären Punkt von OP. Ein Teilproblem von TRSCP mit dem Entwicklungspunkt x^* , sowie dem Trust-Region-Radius Δ^* heißt $TRSP(x^*, \Delta^*)$.

Im Schritt 2 von Algorithmus TRSCP wird das konvexe Teilproblem gelöst, d.h es wird ein stationärer Punkt dieses Teilproblems gesucht. Folglich gilt für den stationären Punkt des Teilproblems (y^k, v^k, w^k) die Karush-Kuhn-Tucker-Bedingung.

Bemerkung 3.3.2

Sei OP gegeben, $\xi > 0$, $\tau > 0$ konstant und $k \in \mathbb{N}_0$. Sei $z^k \in \mathbb{R}^n$ ein Entwicklungspunkt, Δ^k ein Trust-Region-Radius mit $\Delta_i^k > 0$, $i = 1, \dots, n$, und $TRSP(z^k, \Delta^k)$ ein Teilproblem. Sei die LICQ erfüllt, dann gilt für die Lösung (y^k, v^k, w^k) des Teil-

problems die Karush-Kuhn-Tucker-Bedingung mit

$$\begin{aligned} \nabla_x f^k(y^k, z^k, \Delta^k, \xi, \tau) &+ \sum_{j=1}^m v_j^k \nabla_x h_j^k(y^k, z^k, \Delta^k, \xi) \\ &+ \sum_{i=1}^{2n} w_i^k \nabla_x g_i^k(y^k, z^k, \Delta^k) = 0, \end{aligned} \quad (3.48)$$

$$h_j^k(y^k, z^k, \Delta^k, \xi) \leq 0, \quad j \in M, \quad (3.49)$$

$$g_i^k(y^k, z^k, \Delta^k) \leq 0, \quad i = 1, \dots, 2n, \quad (3.50)$$

$$v_j^k h_j^k(y^k, z^k, \Delta^k, \xi) = 0, \quad j \in M, \quad (3.51)$$

$$w_i^k g_i^k(y^k, z^k, \Delta^k) = 0, \quad i = 1, \dots, 2n, \quad (3.52)$$

$$v_j^k \geq 0, \quad j \in M, \quad (3.53)$$

$$w_i^k \geq 0, \quad i = 1, \dots, 2n. \quad (3.54)$$

Im Gegensatz zum SCP-Verfahren von Zillober [74] erfüllt jeder Iterationspunkt bei TRSCP die KKT-Bedingung des Teilproblems, dessen Lösung er ist. Bei Zillober [74] ist dies bedingt durch eine eventuell durchgeführte Line-Search nicht notwendigerweise der Fall. Für den Lagrange-Multiplikator w_i^k gilt, daß mindestens die Hälfte der $2n$ Koeffizienten gleich Null ist. Das folgt aus der Tatsache, daß entweder die untere, die obere oder keine der beiden Begrenzung für einen Koeffizienten der primalen Variablen aktiv sein kann.

Eine Voraussetzung des letzten Satzes, sowie der folgenden Sätze dieses Abschnitts 3.3, ist die Lösbarkeit der Teilprobleme. Wie zu Beginn von Abschnitt 3.2 angesprochen wird davon ausgegangen, daß mit jedem Teilproblem $TRSP(z^k, \Delta^k)$ ein eindeutiger Kandidat für einen neuen Entwicklungspunkt gefunden wird, d.h. im folgenden wird ohne Beschränkung der Allgemeinheit davon ausgegangen, daß die Teilprobleme eindeutig lösbar sind.

Für die weiteren Schritte im globalen Konvergenzbeweis von TRSCP ist die Darstellung des Teilproblems von TRSCP zum Teil nicht geeignet. Für die Beweise der nächsten Aussagen wird eine Taylor-Entwicklung der Funktionen des Teilproblems $TRSP(z^k, \Delta^k)$ betrachtet.

Lemma 3.3.3

Sei OP gegeben, $\xi > 0$, $\tau > 0$ konstant und $k \in \mathbb{N}_0$. Sei $z^k \in \mathbb{R}^n$ ein Entwicklungspunkt und Δ^k ein Trust-Region-Radius mit $\Delta_i^k > 0$, $i = 1, \dots, n$. Dann gilt für die

Funktionen eines Teilproblems $TRSP(z^k, \Delta^k)$

$$f^k(x, z^k, \Delta^k, \xi, \tau) = f(z^k) + \nabla_x f(z^k)^T(x - z^k) + R_{f^k}(x, z^k, \Delta^k, \xi, \tau), \quad (3.55)$$

$$h_j^k(x, z^k, \Delta^k, \xi) = h_j(z^k) + \nabla_x h_j(z^k)^T(x - z^k) + R_{h_j^k}(x, z^k, \Delta^k, \xi), \quad (3.56)$$

$$j \in M,$$

mit den nichtlinearen Anteilen

$$\begin{aligned} R_{f^k}(x, z^k, \Delta^k, \xi, \tau) &= \sum_{i \in I_0^+(z^k)} \left(\frac{\partial f(z^k)}{\partial x_i} + \tau \right) \frac{(x_i - z_i^k)^2}{U(z^k, \Delta^k, \xi)_i - x_i} \\ &\quad - \sum_{i \in I_0^-(z^k)} \left(\frac{\partial f(z^k)}{\partial x_i} - \tau \right) \frac{(x_i - z_i^k)^2}{x_i - L(z^k, \Delta^k, \xi)_i}, \end{aligned} \quad (3.57)$$

$$\begin{aligned} R_{h_j^k}(x, z^k, \Delta^k, \xi) &= \sum_{i \in I_j^+(z^k)} \frac{\partial h_j(z^k)}{\partial x_i} \frac{(x_i - z_i^k)^2}{U(z^k, \Delta^k, \xi)_i - x_i} \\ &\quad - \sum_{i \in I_j^-(z^k)} \frac{\partial h_j(z^k)}{\partial x_i} \frac{(x_i - z_i^k)^2}{x_i - L(z^k, \Delta^k, \xi)_i}, \end{aligned} \quad (3.58)$$

$$j \in M.$$

Beweis:

$$\begin{aligned} f^k(x, z^k, \Delta^k, \xi, \tau) &\stackrel{\text{Lemma 3.2.2}}{=} f(z^k) - \sum_{i=1}^n \left| \frac{\partial f(z^k)}{\partial x_i} \right| (\Delta_i^k + \xi) \\ &\quad + \sum_{i \in I_0^+(z^k)} \frac{\left| \frac{\partial f(z^k)}{\partial x_i} \right| (\Delta_i^k + \xi)^2 + \tau(x_i - z_i^k)^2}{(z_i^k + \Delta_i^k + \xi) - x_i} \\ &\quad + \sum_{i \in I_0^-(z^k)} \frac{\left| \frac{\partial f(z^k)}{\partial x_i} \right| (\Delta_i^k + \xi)^2 + \tau(x_i - z_i^k)^2}{x_i - (z_i^k - \Delta_i^k - \xi)} \end{aligned}$$

$$\begin{aligned}
&= f(z^k) + \sum_{i \in I_0^+(z^k)} \left[\frac{\left| \frac{\partial f(z^k)}{\partial x_i} \right| (\Delta_i^k + \xi)^2 + \tau(x_i - z_i^k)^2}{(z_i^k + \Delta_i^k + \xi) - x_i} \right. \\
&\quad \left. - \frac{\left| \frac{\partial f(z^k)}{\partial x_i} \right| (\Delta_i^k + \xi) ((z_i^k + \Delta_i^k + \xi) - x_i)}{(z_i^k + \Delta_i^k + \xi) - x_i} \right] \\
&+ \sum_{i \in I_0^-(z^k)} \left[\frac{\left| \frac{\partial f(z^k)}{\partial x_i} \right| (\Delta_i^k + \xi)^2 + \tau(x_i - z_i^k)^2}{x_i - (z_i^k - \Delta_i^k - \xi)} \right. \\
&\quad \left. - \frac{\left| \frac{\partial f(z^k)}{\partial x_i} \right| (\Delta_i^k + \xi) (x_i - (z_i^k - \Delta_i^k - \xi))}{x_i - (z_i^k - \Delta_i^k - \xi)} \right] \\
&\stackrel{(3.20)}{=} f(z^k) + \sum_{i \in I_0^+(z^k)} \left[\frac{\frac{\partial f(z^k)}{\partial x_i} [(\Delta_i^k + \xi)^2 - (\Delta_i^k + \xi) ((z_i^k + \Delta_i^k + \xi) - x_i)]}{U(z^k, \Delta^k, \xi)_i - x_i} \right. \\
&\quad \left. + \frac{\tau(x_i - z_i^k)^2}{U(z^k, \Delta^k, \xi)_i - x_i} \right] \\
&- \sum_{i \in I_0^-(z^k)} \left[\frac{\frac{\partial f(z^k)}{\partial x_i} [(\Delta_i^k + \xi)^2 - (\Delta_i^k + \xi) (x_i - (z_i^k - \Delta_i^k - \xi))]}{x_i - L(z^k, \Delta^k, \xi)_i} \right. \\
&\quad \left. - \frac{\tau(x_i - z_i^k)^2}{x_i - L(z^k, \Delta^k, \xi)_i} \right] \\
&= f(z^k) + \sum_{i \in I_0^+(z^k)} \left[\frac{\frac{\partial f(z^k)}{\partial x_i} [(\Delta_i^k + \xi)^2 - (\Delta_i^k + \xi) ((z_i^k - x_i) + (\Delta_i^k + \xi))]}{U(z^k, \Delta^k, \xi)_i - x_i} \right. \\
&\quad \left. + \frac{\tau(x_i - z_i^k)^2}{U(z^k, \Delta^k, \xi)_i - x_i} \right] \\
&- \sum_{i \in I_0^-(z^k)} \left[\frac{\frac{\partial f(z^k)}{\partial x_i} [(\Delta_i^k + \xi)^2 - (\Delta_i^k + \xi) ((x_i - z_i^k) + (\Delta_i^k + \xi))]}{x_i - L(z^k, \Delta^k, \xi)_i} \right. \\
&\quad \left. - \frac{\tau(x_i - z_i^k)^2}{x_i - L(z^k, \Delta^k, \xi)_i} \right]
\end{aligned}$$

$$\begin{aligned}
&= f(z^k) + \sum_{i \in I_0^+(z^k)} \frac{\frac{\partial f(z^k)}{\partial x_i} [-(\Delta_i^k + \xi)(z_i^k - x_i)] + \tau(x_i - z_i^k)^2}{U(z^k, \Delta^k, \xi)_i - x_i} \\
&\quad - \sum_{i \in I_0^-(z^k)} \frac{\frac{\partial f(z^k)}{\partial x_i} [-(\Delta_i^k + \xi)(x_i - z_i^k)] - \tau(x_i - z_i^k)^2}{x_i - L(z^k, \Delta^k, \xi)_i} \\
&= f(z^k) + \sum_{i \in I_0^+(z^k)} \left[\frac{\frac{\partial f(z^k)}{\partial x_i}(x_i - z_i^k) [((z_i^k + \Delta_i^k + \xi) - x_i) + (x_i - z_i^k)]}{U(z^k, \Delta^k, \xi)_i - x_i} \right. \\
&\quad \left. + \frac{\tau(x_i - z_i^k)^2}{U(z^k, \Delta^k, \xi)_i - x_i} \right] \\
&\quad - \sum_{i \in I_0^-(z^k)} \left[\frac{\frac{\partial f(z^k)}{\partial x_i}(x_i - z_i^k) [-(x_i - (z_i^k - \Delta_i^k - \xi)) + (x_i - z_i^k)]}{x_i - L(z^k, \Delta^k, \xi)_i} \right. \\
&\quad \left. - \frac{\tau(x_i - z_i^k)^2}{x_i - L(z^k, \Delta^k, \xi)_i} \right] \\
&= f(z^k) + \sum_{i \in I_0^+(z^k)} \left[\frac{\frac{\partial f(z^k)}{\partial x_i}(x_i - z_i^k)(U(z^k, \Delta^k, \xi)_i - x_i) + \frac{\partial f(z^k)}{\partial x_i}(x_i - z_i^k)^2}{U(z^k, \Delta^k, \xi)_i - x_i} \right. \\
&\quad \left. + \frac{\tau(x_i - z_i^k)^2}{U(z^k, \Delta^k, \xi)_i - x_i} \right] \\
&\quad - \sum_{i \in I_0^-(z^k)} \left[\frac{-\frac{\partial f(z^k)}{\partial x_i}(x_i - z_i^k)(x_i - L(z^k, \Delta^k, \xi)_i) + \frac{\partial f(z^k)}{\partial x_i}(x_i - z_i^k)^2}{x_i - L(z^k, \Delta^k, \xi)_i} \right. \\
&\quad \left. - \frac{\tau(x_i - z_i^k)^2}{x_i - L(z^k, \Delta^k, \xi)_i} \right] \\
&= f(z^k) + \sum_{i \in I_0^+(z^k)} \left[\frac{\partial f(z^k)}{\partial x_i}(x_i - z_i^k) + \left(\frac{\partial f(z^k)}{\partial x_i} + \tau \right) \frac{(x_i - z_i^k)^2}{U(z^k, \Delta^k, \xi)_i - x_i} \right] \\
&\quad - \sum_{i \in I_0^-(z^k)} \left[-\frac{\partial f(z^k)}{\partial x_i}(x_i - z_i^k) + \left(\frac{\partial f(z^k)}{\partial x_i} - \tau \right) \frac{(x_i - z_i^k)^2}{x_i - L(z^k, \Delta^k, \xi)_i} \right]
\end{aligned}$$

$$\begin{aligned}
&= f(z^k) + \nabla_x f(z^k)^T (x - z^k) + \sum_{i \in I_0^+(z^k)} \left(\frac{\partial f(z^k)}{\partial x_i} + \tau \right) \frac{(x_i - z_i^k)^2}{U(z^k, \Delta^k, \xi)_i - x_i} \\
&\quad - \sum_{i \in I_0^-(z^k)} \left(\frac{\partial f(z^k)}{\partial x_i} - \tau \right) \frac{(x_i - z_i^k)^2}{x_i - L(z^k, \Delta^k, \xi)_i}.
\end{aligned}$$

Die Äquivalenz der anderen Gleichungen folgt analog.

q.e.d.

Mit der vorgestellten erweiterten MMA-Approximation der Zielfunktion wird als wesentliches Ziel erreicht, daß die Zielfunktion eines Teilproblems $TRSP(z^k, \Delta^k)$ streng konvex ist. Weiter wird erreicht, daß die Nebenbedingungsfunktionen dieser Teilprobleme konvex sind. Hieraus folgt insgesamt, daß die nichtlinearen Anteile der Taylor-Approximationen nicht negativ sind.

Folgerung 3.3.4

Sei ein Teilproblem $TRSP(z^k, \Delta^k)$ gegeben mit $z^k \in \mathbb{R}^n$, $\Delta^k \in \mathbb{R}^n$ mit $\Delta_i^k > 0$, $i = 1, \dots, n$, und $k \in \mathbb{N}_0$. Für die nichtlinearen Anteile der Funktionen dieses Teilproblems gilt für $x \in \mathbb{R}^n$ aufgrund der (strengen) Konvexität der MMA-Approximation

$$R_{f^k}(x, z^k, \Delta^k, \xi, \tau) > 0, \quad \forall x \neq z^k, \quad (3.59)$$

$$R_{h_j^k}(x, z^k, \Delta^k, \xi) \geq 0, \quad \forall x \wedge \forall j \in M. \quad (3.60)$$

Für die nichtlinearen Restglieder der Taylor-Entwicklung der Nebenbedingungen wird $R_{h^k}(x, z^k, \Delta^k, \xi) := \left(R_{h_1^k}(x, z^k, \Delta^k, \xi), \dots, R_{h_m^k}(x, z^k, \Delta^k, \xi) \right)^T$ festgelegt. Es gilt

$$h^k(x, z^k, \Delta^k, \xi) = h(z^k) + J_h(z^k)^T (x - z^k) + R_{h^k}(x, z^k, \Delta^k, \xi),$$

bzw.

$$J_h(z^k)^T (x - z^k) = h^k(x, z^k, \Delta^k, \xi) - h(z^k) - R_{h^k}(x, z^k, \Delta^k, \xi), \quad (3.61)$$

mit der Jacobi-Matrix $J_h(z^k)$ (Bezeichnung 2.4.6).

Für die folgenden Beweise werden die Ableitungen der Taylor-Entwicklung der Funktionen eines Teilproblems $TRSP(z^k, \Delta^k)$ benötigt.

Bemerkung 3.3.5

Sei OP gegeben, $\xi > 0$, $\tau > 0$ konstant und $k \in \mathbb{N}_0$. Sei $z^k \in \mathbb{R}^n$ ein Entwicklungspunkt, Δ^k ein Trust-Region-Radius mit $\Delta_i^k > 0$, $i = 1, \dots, n$, und $TRSP(z^k, \Delta^k)$ ein Teilproblem. Dann gilt für die Ableitung der Funktionen des konvexen Teilproblems $TRSP(z^k, \Delta^k)$

$$\nabla_x f^k(x, z^k, \Delta^k, \xi, \tau) = \nabla_x f(z^k) + \nabla_x R_{f^k}(x, z^k, \Delta^k, \xi, \tau), \quad (3.62)$$

$$\nabla_x h_j^k(x, z^k, \Delta^k, \xi) = \nabla_x h_j(z^k) + \nabla_x R_{h_j^k}(x, z^k, \Delta^k, \xi), \quad \forall j \in M, \quad (3.63)$$

mit

$$\frac{\partial R_{f^k}(x, z^k, \Delta^k, \xi, \tau)}{\partial x_i} = \begin{cases} \left(\frac{\partial f(z^k)}{\partial x_i} + \tau \right) (x_i - z_i^k) \frac{2U(z^k, \Delta^k, \xi)_i - z_i^k - x_i}{(U(z^k, \Delta^k, \xi)_i - x_i)^2}, \\ \text{falls } i \in I_0^+(z^k), \\ \left(\frac{\partial f(z^k)}{\partial x_i} - \tau \right) (x_i - z_i^k) \frac{2L(z^k, \Delta^k, \xi)_i - z_i^k - x_i}{(x_i - L(z^k, \Delta^k, \xi)_i)^2}, \\ \text{falls } i \in I_0^-(z^k), \end{cases} \quad (3.64)$$

und

$$\frac{\partial R_{h_j^k}(x, z^k, \Delta^k, \xi)}{\partial x_i} = \begin{cases} \frac{\partial h_j(z^k)}{\partial x_i} (x_i - z_i^k) \frac{2U(z^k, \Delta^k, \xi)_i - z_i^k - x_i}{(U(z^k, \Delta^k, \xi)_i - x_i)^2}, \\ \text{falls } i \in I_j^+(z^k), \\ \frac{\partial h_j(z^k)}{\partial x_i} (x_i - z_i^k) \frac{2L(z^k, \Delta^k, \xi)_i - z_i^k - x_i}{(x_i - L(z^k, \Delta^k, \xi)_i)^2}, \\ \text{falls } i \in I_j^-(z^k), \end{cases} \quad (3.65)$$

$\forall j \in M.$

Die nichttrivialen, nichtlinearen Anteile der Approximationen des Teilproblems können abgeschätzt werden. Hierfür wird die Karush-Kuhn-Tucker-Bedingung des Teilproblems für (y^k, v^k, w^k) ausgewertet, indem die Taylor-Entwicklungen (3.62) und (3.63) der Ableitungen der Approximationen in die Bedingung (3.48) eingesetzt werden, d.h.

$$\begin{aligned} \nabla_x f(z^k) + \nabla_x R_{f^k}(y^k, z^k, \Delta^k, \xi, \tau) &+ \sum_{j=1}^m v_j^k \left(\nabla_x h_j(z^k) + \nabla_x R_{h_j^k}(y^k, z^k, \Delta^k, \xi) \right) \\ &+ \sum_{i=1}^{2n} w_i^k \nabla_x g_i^k(y^k, z^k, \Delta^k) = 0. \end{aligned} \quad (3.66)$$

Diese Beziehung (3.66) mit $J_h(z^k)$ aus Bezeichnung 2.4.6 umgeformt und mit $[y^k - z^k]$

erweitert ergibt folgende Bedingung:

$$\begin{aligned}
\nabla_x f(z^k)^T [y^k - z^k] &= -\nabla_x R_{f^k}(y^k, z^k, \Delta^k, \xi, \tau)^T [y^k - z^k] \\
&\quad - (v^k)^T J_h(z^k)^T [y^k - z^k] \\
&\quad - \sum_{j=1}^m v_j^k \nabla_x R_{h_j^k}(y^k, z^k, \Delta^k, \xi)^T [y^k - z^k] \\
&\quad - \sum_{i=1}^{2n} w_i^k \nabla_x g_i^k(y^k, z^k, \Delta^k)^T [y^k - z^k].
\end{aligned} \tag{3.67}$$

Für den nichtlinearen Anteil der approximierten Zielfunktion gilt:

Lemma 3.3.6

Sei OP gegeben, $\Delta_{\max} > 0$, $\xi > 0$, $\tau > 0$ konstant und $k \in \mathbb{N}_0$. Sei $z^k \in \mathbb{R}^n$ ein Entwicklungspunkt, Δ^k ein Trust-Region-Radius mit $0 < \Delta_i^k \leq \Delta_{\max}$, $i = 1, \dots, n$, und $TRSP(z^k, \Delta^k)$ ein Teilproblem. Die LICQ sei erfüllt und die Lösung (y^k, v^k, w^k) des Teilproblems $TRSP(z^k, \Delta^k)$ bestimmt. Dann gilt

$$\nabla_x R_{f^k}(y^k, z^k, \Delta^k, \xi, \tau)^T [y^k - z^k] \geq \frac{\tau(\delta^k)^2}{4(\Delta_{\max} + \xi)}, \tag{3.68}$$

mit $\delta^k := \|y^k - z^k\|_2$.

Beweis:

Für jedes feste $i \in I_0^+(z^k)$ gilt folgende Abschätzung für den nichtlinearen Anteil der Ableitung der approximierten Zielfunktion:

$$\begin{aligned}
&\nabla_x R_{f^k}(y^k, z^k, \Delta^k, \xi, \tau)_i [y_i^k - z_i^k] \\
&\stackrel{(3.64)}{=} \left(\frac{\partial f}{\partial x_i}(z^k) + \tau \right) [y_i^k - z_i^k]^2 \frac{\Delta_i^k + \xi + U(z^k, \Delta^k, \xi)_i - y_i^k}{(U(z^k, \Delta^k, \xi)_i - y_i^k)^2} \\
&\geq \tau [y_i^k - z_i^k]^2 \frac{\Delta_i^k + \xi + U(z^k, \Delta^k, \xi)_i - (z_i^k + \Delta_i^k)}{(U(z^k, \Delta^k, \xi)_i - L(z^k, \Delta^k, \xi)_i)^2} \\
&= \tau [y_i^k - z_i^k]^2 \frac{2(\Delta_i^k + \xi) - \Delta_i^k}{(2(\Delta_i^k + \xi))^2} \\
&\geq \tau [y_i^k - z_i^k]^2 \frac{\Delta_i^k + \xi}{(2(\Delta_i^k + \xi))^2} \\
&\geq \frac{\tau [y_i^k - z_i^k]^2}{4(\Delta_{\max} + \xi)}.
\end{aligned}$$

Analog gilt für jedes feste $i \in I_0^-(z^k)$ das entsprechende Ergebnis, d.h. zusammen gilt:

$$\nabla_x R_{f^k}(y^k, z^k, \Delta^k, \xi, \tau)^T [y^k - z^k] \geq \frac{\tau \sum_{i=1}^n [y_i^k - z_i^k]^2}{4(\Delta_{\max} + \xi)} = \frac{\tau(\delta^k)^2}{4(\Delta_{\max} + \xi)}.$$

q.e.d.

Für die nichtlinearen Anteile der approximierten Nebenbedingungen läßt sich folgende Beziehung abschätzen:

Lemma 3.3.7

Sei OP gegeben, $\xi > 0$ konstant und $k \in \mathbb{N}_0$. Sei $z^k \in \mathbb{R}^n$ ein Entwicklungspunkt, Δ^k ein Trust-Region-Radius mit $\Delta_i^k > 0$, $i = 1, \dots, n$, und $TRSP(z^k, \Delta^k)$ ein Teilproblem. Die LICQ sei erfüllt und die Lösung (y^k, v^k, w^k) des Teilproblems $TRSP(z^k, \Delta^k)$ bestimmt. Dann gilt

$$-(v^k)^T \nabla_x R_{h^k}(y^k, z^k, \Delta^k, \xi)^T [y^k - z^k] + (v^k)^T R_{h^k}(y^k, z^k, \Delta^k, \xi) \leq 0. \quad (3.69)$$

Beweis:

Es ist

$$\begin{aligned} & -(v^k)^T \nabla_x R_{h^k}(y^k, z^k, \Delta^k, \xi)^T [y^k - z^k] + (v^k)^T R_{h^k}(y^k, z^k, \Delta^k, \xi) \\ & \stackrel{(3.58), (3.65)}{=} \sum_{I_j^+(z^k)} -v_j^k \left(\frac{\partial h_j(z^k)}{\partial x_i} [y_i^k - z_i^k] \frac{2U(z^k, \Delta^k, \xi)_i - z_i^k - y_i^k}{(U(z^k, \Delta^k, \xi)_i - y_i^k)^2} [y_i^k - z_i^k] \right. \\ & \quad \left. - \frac{\partial h_j(z^k)}{\partial x_i} \frac{[y_i^k - z_i^k]^2}{U(z^k, \Delta^k, \xi)_i - y_i^k} \right) \\ & \quad + \sum_{I_i^-(z^k)} -v_j^k \left(\frac{\partial h_j(z^k)}{\partial x_i} [y_i^k - z_i^k] \frac{2L(z^k, \Delta^k, \xi)_i - z_i^k - y_i^k}{(y_i^k - L(z^k, \Delta^k, \xi)_i)^2} [y_i^k - z_i^k] \right. \\ & \quad \left. + \frac{\partial h_j(z^k)}{\partial x_i} \frac{[y_i^k - z_i^k]^2}{y_i^k - L(z^k, \Delta^k, \xi)_i} \right) \end{aligned}$$

$$\begin{aligned}
&= \sum_{I_j^+(z^k)} -v_j^k \frac{\partial h_j(z^k)}{\partial x_i} [y_i^k - z_i^k]^2 \frac{2U(z^k, \Delta^k, \xi)_i - z_i^k - y_i^k - (U(z^k, \Delta^k, \xi)_i - y_i^k)}{(U(z^k, \Delta^k, \xi)_i - y_i^k)^2} \\
&\quad + \sum_{I_j^-(z^k)} -v_j^k \frac{\partial h_j(z^k)}{\partial x_i} [y_i^k - z_i^k]^2 \frac{2L(z^k, \Delta^k, \xi)_i - z_i^k - y_i^k + (y_i^k - L(z^k, \Delta^k, \xi)_i)}{(y_i^k - L(z^k, \Delta^k, \xi)_i)^2} \\
(3.20) \quad &\stackrel{=}{=} \sum_{I_j^+(z^k)} -v_j^k \frac{\partial h_j(z^k)}{\partial x_i} [y_i^k - z_i^k]^2 \frac{z_i^k + \Delta_i^k + \xi - z_i^k}{(U(z^k, \Delta^k, \xi)_i - y_i^k)^2} \\
&\quad + \sum_{I_j^-(z^k)} -v_j^k \frac{\partial h_j(z^k)}{\partial x_i} [y_i^k - z_i^k]^2 \frac{z_i^k - \Delta_i^k - \xi - z_i^k}{(y_i^k - L(z^k, \Delta^k, \xi)_i)^2} \\
&= \sum_{I_j^+(z^k)} \underbrace{-v_j^k}_{\leq 0} \underbrace{\frac{\partial h_j(z^k)}{\partial x_i}}_{\geq 0} \underbrace{[y_i^k - z_i^k]^2}_{\geq 0} \underbrace{\frac{\overbrace{\Delta_i^k + \xi}^{>0}}{(U(z^k, \Delta^k, \xi)_i - y_i^k)^2}}_{\geq 0} \\
&\quad + \sum_{I_j^-(z^k)} \underbrace{v_j^k}_{\geq 0} \underbrace{\frac{\partial h_j(z^k)}{\partial x_i}}_{< 0} \underbrace{[y_i^k - z_i^k]^2}_{\geq 0} \underbrace{\frac{\overbrace{\Delta_i^k + \xi}^{>0}}{(y_i^k - L(z^k, \Delta^k, \xi)_i)^2}}_{\geq 0} \\
&\leq 0.
\end{aligned}$$

q.e.d.

Nachdem Abschätzungen für die nichtlinearen Anteile der approximierten Funktionen vorgestellt wurden, folgen Abschätzungen, die unterscheiden in welcher Indexmenge die Indizes der Nebenbedingungen liegen.

Lemma 3.3.8

Sei OP gegeben, $\Delta_{\max} > 0$, $\xi > 0$ konstant und $k \in \mathbb{N}_0$. Weiter gegeben sei ein Entwicklungspunkt $z^k \in \mathbb{R}^n$, ein Lagrange-Multiplikator $u^k \in \mathbb{R}^m$ mit $u_j^k \geq 0$, $j \in M$, ein Trust-Region-Radius Δ^k mit $0 < \Delta_i^k \leq \Delta_{\max}$, $i = 1, \dots, n$ und ein Penalty-Parameter $r \in \mathbb{R}^m$ mit $r_j > 1$, $j \in M$. Die LICQ sei erfüllt und die Lösung (y^k, v^k, w^k) des Teilproblems $TRSP(z^k, \Delta^k)$ bestimmt. $A(r, y^k, v^k)$, $K(r, y^k, v^k)$, $\bar{h}(x)$, \bar{u}^k und $P \in \mathbb{R}^{m \times m}$ seien entsprechend Bezeichnung 2.4.6 definiert. Dann ist

$$\sum_{j \in K(r, y^k, v^k)} v_j^k h_j(z^k) \leq 0 \quad (3.70)$$

und

$$(\bar{u}^k + P\bar{h}(z^k))^T h^k(y^k, z^k, \Delta^k, \xi) \leq 0, \quad (3.71)$$

$$-(\bar{u}^k + P\bar{h}(z^k))^T R_{h^k}(y^k, z^k, \Delta^k, \xi) \leq 0. \quad (3.72)$$

Beweis:

(y^k, v^k, w^k) ist stationärer Punkt des Teilproblems $TRSP(z^k, \Delta^k)$. Aus der Karush-Kuhn-Tucker Bedingung (3.53) folgt $v_j^k \geq 0$, $j \in M$. Weiter gilt $h_j(z^k) < 0$ für $j \in K(r_0^k, y^k, v^k)$ und zusammen folgt der erste Teil der Behauptung. Die weitere Behauptung folgt für alle $j \in K(r, y^k, v^k)$ aus der Definition von $\bar{h}(x)$ bzw. \bar{u}^k und für alle $j \in A(r, y^k, v^k)$ mit $h_j(z^k) \geq -\frac{u^k}{r}$. Das heißt, für $j \in A(r, y^k, v^k)$ ist

$$u_j + r_j^k h_j(z^k) \geq u_j - r_j \frac{u_j^k}{r_j} = 0. \quad (3.73)$$

Zusammen mit $h^k(y^k, z^k, \Delta^k, \xi) \leq 0$, $R_{h^k}(y^k, z^k, \Delta^k, \xi) \geq 0$ folgt die Behauptung.

q.e.d.

Es wird in den folgenden Beweisen gefordert, daß die LICQ für die Lösung des Teilproblems erfüllt ist. Diese Forderung alleine garantiert, daß die Lagrange-Multiplikatoren eindeutig sind. Also sind für ein fixes $k \in \mathbb{N}_0$ die dualen Variablen auch beschränkt. Es ist möglich, daß die Lagrange-Multiplikatoren zwar endlich sind, aber im Verlauf der Iteration unbeschränkt ansteigen. Wird das Einhalten der LICQ nicht nur für Glieder der Iterationsfolge gefordert, sondern auch für ihre Häufungspunkte, dann sind die Lagrange-Multiplikatoren unabhängig vom Iterationsparameter beschränkt.

Satz 3.3

Sei OP gegeben und $\xi > 0$, $\tau > 0$ konstant. Weiter sei eine Folge von Entwicklungspunkten $\{z^k\}_{k=0,1,2,\dots} \subset \mathbb{R}^n$ mit dafür geeigneten Lagrange-Multiplikatoren $\{u^k\}_{k=0,1,2,\dots} \subset \mathbb{R}^m$ mit $u_j^k \geq 0$, $j \in M$, und Trust-Region-Radien $\{\Delta^k\}_{k=0,1,2,\dots} \subset \mathbb{R}^n$ mit $\Delta_i^k > 0$, $i = 1, \dots, n$, gegeben. Die LICQ sei für jeden Entwicklungspunkt, als auch für jeden Häufungspunkt der Folge von Entwicklungspunkten erfüllt. Die Folge $\{(y^k, v^k, w^k)\}_{k=0,1,2,\dots}$ von Lösungen der Teilprobleme $TRSP(z^k, \Delta^k)$ sei bestimmt. Dann existieren $V \subset \mathbb{R}^m$ kompakt und $W \subset \mathbb{R}^{2n}$ kompakt mit $v^k \in V$ und $w^k \in W$ für alle $k \in \mathbb{N}_0$.

Beweis:

Sei

$$\bar{v}^k := \max_{j \in M} v_j^k < \infty, \quad (3.74)$$

$$\bar{w}^k := \max_{i=1,\dots,2n} w_i^k < \infty. \quad (3.75)$$

Aufgrund der erfüllten LICQ für alle Entwicklungspunkte, sowie für alle Häufungspunkte dieser Folge, liefert die Störungstheorie in Fiacco [14], daß die Änderung der dualen Variable kontinuierlich ist. Das heißt, ist der Unterschied in der primalen Variablen zweier Teilproblemlösungen klein, dann ist das auch beim Unterschied der

zwei dualen Lösungen der Fall. Also folgt, daß eine Schranke $u_{\max} < \infty$ existiert mit $\bar{v}^k, \bar{w}^k \leq u_{\max}$. Das heißt die Mengen

$$\begin{aligned} V &:= \{v \in \mathbb{R}^m : 0 \leq v_j \leq u_{\max}, j \in M\}, \\ W &:= \{w \in \mathbb{R}^{2n} : 0 \leq w_i \leq u_{\max}, i = 1, \dots, 2n\} \end{aligned} \quad (3.76)$$

sind kompakt und es gilt $v^k \in V$, $w^k \in W$ und damit die Behauptung.

q.e.d.

Folgerung 3.3.9

In Bemerkung 2.4.5 wird darauf hingewiesen, daß für jedes $z^k \in X$ kompakt mit dem zugehörigen $u^k \in V$ kompakt ein $\bar{r}^k \in \mathbb{R}^m$ existiert, so daß für alle $r \in \mathbb{R}^m$ mit $r_j \geq \bar{r}_j^k$, $j \in M$, der Wert der erweiterten Lagrange-Funktion $\Phi_r \begin{pmatrix} z^k \\ u^k \end{pmatrix}$ nach unten beschränkt ist, d.h. für die in Satz 3.3 betrachteten Folgen ist die erweiterte Lagrange-Funktion nach unten beschränkt.

Als nächstes wird gezeigt, daß es einen endlichen Penalty-Parameter gibt, mit dem die Suchrichtung bzgl. der erweiterten Lagrange-Funktion eine für die Konvergenz von TRSCP hinreichende Abstiegsseigenschaft besitzt. Hieraus folgt bedingt durch die gewählte Form des Penalty-Parameter-Updates in TRSCP, daß jede Anpassung des Penalty-Parameters im Verfahren endlich ist.

Satz 3.4

Sei OP und $k \in \mathbb{N}_0$ gegeben, sowie $\Delta_{\max} > 0$, $\xi > 0$, $\tau > 0$ konstant. Weiter gegeben sei ein Entwicklungspunkt $z^k \in \mathbb{R}^n$, ein Lagrange-Multiplikator $u^k \in \mathbb{R}^m$ mit $u_j^k \geq 0$, $j \in M$, und ein Trust-Region-Radius $\Delta^k \in \mathbb{R}^n$ mit $0 < \Delta_i^k \leq \Delta_{\max}$, $i = 1, \dots, n$. Die LICQ sei erfüllt und die Lösung (y^k, v^k, w^k) des Teilproblems $TRSP(z^k, \Delta^k)$ sei bestimmt. Sei $s^k := \begin{pmatrix} y^k - z^k \\ v^k - u^k \end{pmatrix}$, $\delta^k := \|y^k - z^k\|_2$ und \bar{u}^k , $\bar{h}(x)$, $\hat{h}(r, x, u)$, P und $J_h(x)$ wie in Bezeichnung 2.4.6 festgelegt. Dann existiert $\bar{r}^k \in \mathbb{R}^m$, so daß s^k für alle $r \in \mathbb{R}^m$ mit $r_j \geq \bar{r}_j^k$, $j \in M$, eine Abstiegsrichtung für Φ_r ist, mit

$$\nabla \Phi_r(z^k, u^k)^T s^k \leq -\frac{\tau(\delta^k)^2}{8(\Delta_{\max} + \xi)}. \quad (3.77)$$

Beweis:

Als erstes wird $\Gamma_r^k := \nabla \Phi_r(z^k, u^k)^T s^k$ für Penalty-Parameter $r \in \mathbb{R}^m$ mit $r_j > 1$, $j \in M$, betrachtet. Γ_r^k wird mit Hilfe von (2.28) umgeformt.

$$\begin{aligned} \Gamma_r^k &:= \nabla \Phi_r(z^k, u^k)^T s^k \\ &= \nabla_x f(z^k)^T [y^k - z^k] + (\bar{u}^k + P\bar{h}(z^k))^T J_h(z^k)^T [y^k - z^k] \\ &\quad + \hat{h}(r, z^k, u^k)^T [v^k - u^k]. \end{aligned}$$

Weiter kommen erst die Beziehungen (3.67) und sofort anschließend (3.61) zum Einsatz.

$$\begin{aligned}
\Gamma_r^k &= -\nabla_x R_{f^k}(y^k, z^k, \Delta^k, \xi, \tau)^T [y^k - z^k] \\
&\quad - \underbrace{(v^k)^T (h^k(y^k, z^k, \Delta^k, \xi) - h(z^k) - R_{h^k}(y^k, z^k, \Delta^k, \xi))}_{(3.61)} \\
&\quad - \sum_{j=1}^m v_j^k \nabla_x R_{h_j^k}(y^k, z^k, \Delta^k, \xi)^T [y^k - z^k] \\
&\quad - \sum_{i=1}^{2n} w_i^k \nabla_x g_i^k(y^k, z^k, \Delta^k)^T [y^k - z^k] \\
&\quad + (\bar{u}^k + P\bar{h}(z^k))^T \underbrace{(h^k(y^k, z^k, \Delta^k, \xi) - h(z^k) - R_{h^k}(y^k, z^k, \Delta^k, \xi))}_{(3.61)} \\
&\quad + \hat{h}(r, z^k, u^k)^T [v^k - u^k].
\end{aligned}$$

Nun wird Γ_r^k mit Hilfe von (3.51) vereinfacht und mit (3.71) und (3.72) abgeschätzt.

$$\begin{aligned}
\Gamma_r^k &\leq -\nabla_x R_{f^k}(y^k, z^k, \Delta^k, \xi, \tau)^T [y^k - z^k] + (v^k)^T h(z^k) + (v^k)^T R_{h^k}(y^k, z^k, \Delta^k, \xi) \\
&\quad - \sum_{j=1}^m v_j^k \nabla_x R_{h_j^k}(y^k, z^k, \Delta^k, \xi)^T [y^k - z^k] \\
&\quad - \sum_{i=1}^{2n} w_i^k \nabla_x g_i^k(y^k, z^k, \Delta^k)^T [y^k - z^k] \\
&\quad - (\bar{u}^k + P\bar{h}(z^k))^T h(z^k) + \hat{h}(r, z^k, u^k)^T [v^k - u^k].
\end{aligned}$$

Mit den Aussagen der Lemmata 3.3.6 und 3.3.7 wird Γ_r^k weiter abgeschätzt. Außerdem sind Koeffizienten des Lagrange-Multiplikators für inaktive Nebenbedingungen zum Sicherstellen der Trust-Region gleich Null (folgt aus (3.52)) und $\nabla_x g^k(y^k, z^k, \Delta^k)$ besitzt nur Diagonalelemente, d.h. ein Element der Diagonale ist Eins oder Minus-Eins.

$$\begin{aligned}
\Gamma_r^k &\stackrel{(3.68)}{\leq} -\frac{\tau(\delta^k)^2}{4(\Delta_{\max} + \xi)} + \overbrace{\sum_{j=1}^m v_j^k h_j(z^k)}^1 \\
&\quad + \sum_{i \in \{1, \dots, n\}: y_i^k = z_i^k - \Delta_i^k} w_i^k [y_i^k - z_i^k] - \sum_{i \in \{1, \dots, n\}: y_i^k = z_i^k + \Delta_i^k} w_{n+i}^k [y_i^k - z_i^k] \\
&\quad - (\bar{u}^k + P\bar{h}(z^k))^T h(z^k) + \underbrace{\hat{h}(r, z^k, u^k)^T [v^k - u^k]}_2.
\end{aligned}$$

Dieser Ausdruck für Γ_r^k wird über die Indexmengen $A(r, y^k, v^k)$ und $K(r, y^k, v^k)$ umsortiert.

$$\begin{aligned}
\Gamma_r^k \leq & -\frac{\tau(\delta^k)^2}{4(\Delta_{\max} + \xi)} - \sum_{j \in A(r, y^k, v^k)} u_j^k h_j(z^k) - \sum_{j \in A(r, y^k, v^k)} r_j h_j^2(z^k) \\
& + \underbrace{\sum_{j \in A(r, y^k, v^k)} h_j(z^k) [v_j^k - u_j^k]}_2 + \underbrace{\sum_{j \in A(r, y^k, v^k)} v_j^k h_j(z^k)}_1 \\
& - \sum_{i \in \{1, \dots, n\}: y_i^k = z_i^k - \Delta_i^k} w_i^k \Delta_i^k - \sum_{i \in \{1, \dots, n\}: y_i^k = z_i^k + \Delta_i^k} w_{n+i}^k \Delta_i^k \\
& - \underbrace{\sum_{j \in K(r, y^k, v^k)} \frac{u_j^k}{r_j} [v_j^k - u_j^k]}_2 + \underbrace{\sum_{j \in K(r, y^k, v^k)} v_j^k h_j(z^k)}_1.
\end{aligned}$$

Mit Aussage (3.70) und $w_i^k \Delta_i^k \geq 0$, für $i = 1, \dots, n$, fallen weitere Teile des Ausdrucks weg und zusammengefaßt ergibt sich:

$$\begin{aligned}
\nabla \Phi_r(z^k, u^k)^T s^k \leq & -\frac{\tau(\delta^k)^2}{4(\Delta_{\max} + \xi)} - \sum_{j \in A(r, y^k, v^k)} r_j h_j^2(z^k) \\
& + \sum_{j \in A(r, y^k, v^k)} 2h_j(z^k) [v_j^k - u_j^k] \\
& - \sum_{j \in K(r, y^k, v^k)} \frac{u_j^k}{r_j} [v_j^k - u_j^k].
\end{aligned} \tag{3.78}$$

Mit $2h_j(z^k) [v_j^k - u_j^k] \leq 2|h_j(z^k)| u_{\max}$ und $\frac{u_j^k}{r_j} [v_j^k - u_j^k] \geq -\frac{u_{\max}^2}{r_j}$ ergibt sich folgende Abschätzung:

$$\begin{aligned}
\nabla \Phi_r(z^k, u^k)^T s^k \leq & -\frac{\tau(\delta^k)^2}{4(\Delta_{\max} + \xi)} \\
& + \sum_{j \in A(r, y^k, v^k)} (2|h_j(z^k)| u_{\max} - r_j h_j^2(z^k)) \\
& + \sum_{j \in K(r, y^k, v^k)} \frac{u_{\max}^2}{r_j},
\end{aligned} \tag{3.79}$$

wobei Satz 3.3 die Existenz eines u_{\max} mit $0 \leq u_j^k, v_j^k \leq u_{\max} < \infty$ für alle $j \in M$ liefert. Damit gilt für $h_j(z^k) = 0$ oder $r_j \geq \frac{2u_{\max}}{|h_j(z^k)|}$, $j \in A(r, y^k, v^k)$:

$$\nabla \Phi_r(z^k, u^k)^T s^k \leq -\frac{\tau(\delta^k)^2}{4(\Delta_{\max} + \xi)} + \sum_{j \in K(r, y^k, v^k)} \frac{u_{\max}^2}{r_j}, \tag{3.80}$$

und für hinreichend große Penalty-Parameter $r \in \mathbb{R}^m$ mit

$$\sum_{j \in K(r, y^k, v^k)} \frac{1}{r_j} \leq \frac{1}{u_{\max}^2} \frac{\tau(\delta^k)^2}{8(\Delta_{\max} + \xi)} \quad (3.81)$$

gilt

$$\sum_{j \in K(r, y^k, v^k)} \frac{u_{\max}^2}{r_j} \leq \frac{\tau(\delta^k)^2}{8(\Delta_{\max} + \xi)},$$

und damit die Behauptung.

q.e.d.

Im Algorithmus TRSCP wird der Penalty-Parameter mit Hilfe einer Teiliteration so lange erhöht, bis die erweiterte Lagrange-Funktion eine für die Konvergenz des Algorithmus hinreichende Abstiegseigenschaft besitzt. Durch die Form der Erhöhung mittels eines konstanten Faktors größer Eins und der Existenz eines endlichen hinreichend großen Penalty-Parameters folgt die Endlichkeit dieser Teiliteration.

Nachdem die Endlichkeit der Teiliteration zur Anpassung der Penalty-Parameter gezeigt wurde, wird als nächstes nachgewiesen, daß die Teiliteration der inneren Iterationen endlich ist. Um dies zu zeigen muß für innere Iterationen ein bestimmtes Verhalten des Lagrange-Multiplikators vorausgesetzt werden. Dafür sei für $k \in \mathbb{N}_0$ der folgende Quotient $\beta(z^k, y^k, u^k, v^k) \in \mathbb{R}$ für Vektoren $u^k, v^k \in \mathbb{R}^m$ und $z^k, y^k \in \mathbb{R}^n$ mit $z^k \neq y^k$ über

$$\beta(z^k, y^k, u^k, v^k) := \frac{\|v^k - u^k\|_2^2}{\|y^k - z^k\|_2^2} \quad (3.82)$$

definiert.

Lemma 3.3.10

Sei OP und $k, q \in \mathbb{N}_0$ gegeben, sowie $\Delta_{\max} > 0$, $\xi > 0$, $\tau > 0$ konstant. Weiter gegeben sei ein Entwicklungspunkt $z^k \in \mathbb{R}^n$, ein Lagrange-Multiplikator $u^k \in \mathbb{R}^m$ mit $u_j^k \geq 0$, $j \in M$, und ein Trust-Region-Radius $\Delta_q^k \in \mathbb{R}^n$ mit $0 < \Delta_{q,i}^k \leq \Delta_{\max}$, $i = 1, \dots, n$. Die LICQ sei erfüllt und die Lösung (y_q^k, v_q^k, w_q^k) des Teilproblems $TRSP(z^k, \Delta_q^k)$ sei bestimmt. Sei $s_q^k := \begin{pmatrix} y_q^k - z^k \\ v_q^k - u^k \end{pmatrix}$ und $\delta_q^k := \|y_q^k - z^k\|_2$. Existiert $\bar{\beta} > 0$ mit $\beta(z^k, y_q^k, u^k, v_q^k) \leq \bar{\beta}$ und gilt $j \in A(r, y_q^k, v_q^k) \Leftrightarrow h_j(y_q^k) = 0$, dann ist

$$\nabla \Phi_{\hat{r}}(z^k, u^k)^T s_q^k \leq -\frac{\tau(\delta_q^k)^2}{8(\Delta_{\max} + \xi)}, \quad (3.83)$$

für $\delta_q^k \leq \delta_0^k$ und Penalty-Parametern $\hat{r} \in \mathbb{R}^m$ mit

$$\hat{r}_j \geq \frac{8\bar{\beta}(\Delta_{\max} + \xi)}{\tau}, \quad (3.84)$$

für $j \in M$.

Beweis:

Die Voraussetzungen für Satz 3.4 sind erfüllt, d.h. mit (3.78) existiert $\hat{r} \in \mathbb{R}^m$ mit $\hat{r}_j > 1$, $j \in M$, und

$$\begin{aligned}
\nabla \Phi_{\hat{r}}(z^k, u^k)^T s_q^k &\leq -\frac{\tau(\delta_q^k)^2}{4(\Delta_{\max} + \xi)} - \sum_{j \in A(\hat{r}, y_q^k, v_q^k)} \hat{r}_j h_j^2(z^k) \\
&\quad + \sum_{j \in A(\hat{r}, y_q^k, v_q^k)} 2h_j(z^k)[v_{q,j}^k - u_j^k] \\
&\quad - \sum_{j \in K(\hat{r}, y_q^k, v_q^k)} \frac{u_j^k}{\hat{r}_j} [v_{q,j}^k - u_j^k] \\
&= -\frac{\tau(\delta_q^k)^2}{4(\Delta_{\max} + \xi)} - \sum_{j \in A(\hat{r}, y_q^k, v_q^k)} [\hat{r}_j h_j^2(z^k) - 2h_j(z^k)[v_{q,j}^k - u_j^k]] \\
&\quad + \sum_{j \in K(\hat{r}, y_q^k, v_q^k)} \frac{v_{q,j}^k - u_j^k}{\hat{r}_j} [v_{q,j}^k - u_j^k],
\end{aligned}$$

mit $v_{q,j}^k = 0$ für $j \in K(\hat{r}, y_q^k, v_q^k)$ nach Voraussetzung. Weiter ist

$$\begin{aligned}
\nabla \Phi_{\hat{r}}(z^k, u^k)^T s_q^k &\leq -\frac{\tau(\delta_q^k)^2}{4(\Delta_{\max} + \xi)} - \sum_{j \in A(\hat{r}, y_q^k, v_q^k)} [\hat{r}_j h_j^2(z^k) - 2h_j(z^k)[v_{q,j}^k - u_j^k]] \\
&\quad + \sum_{j \in K(\hat{r}, y_q^k, v_q^k)} \frac{1}{\hat{r}_j} [v_{q,j}^k - u_j^k]^2 \\
&= -\frac{\tau(\delta_q^k)^2}{4(\Delta_{\max} + \xi)} + \sum_{j \in M} \frac{1}{\hat{r}_j} [v_{q,j}^k - u_j^k]^2 \\
&\quad - \sum_{j \in A(\hat{r}, y_q^k, v_q^k)} \left[\hat{r}_j h_j^2(z^k) - 2h_j(z^k)[v_{q,j}^k - u_j^k] + \frac{1}{\hat{r}_j} [v_{q,j}^k - u_j^k]^2 \right] \\
&= -\frac{\tau(\delta_q^k)^2}{4(\Delta_{\max} + \xi)} + \sum_{j \in M} \frac{1}{\hat{r}_j} [v_{q,j}^k - u_j^k]^2 \\
&\quad - \sum_{j \in A(\hat{r}, y_q^k, v_q^k)} \left(\sqrt{\hat{r}_j} h_j(z^k) - \frac{1}{\sqrt{\hat{r}_j}} [v_{q,j}^k - u_j^k] \right)^2 \\
&\leq -\frac{\tau(\delta_q^k)^2}{4(\Delta_{\max} + \xi)} + \sum_{j \in M} \frac{1}{\hat{r}_j} [v_{q,j}^k - u_j^k]^2
\end{aligned}$$

Sei

$$\hat{r}_j \geq \frac{8\bar{\beta}(\Delta_{\max} + \xi)}{\tau},$$

für $j \in M$, dann ist

$$\begin{aligned} \sum_{j \in M} \frac{1}{\hat{r}_j} [v_{q,j}^k - u_j^k]^2 &\leq \sum_{j \in M} \frac{\tau}{8\bar{\beta}(\Delta_{\max} + \xi)} [v_{q,j}^k - u_j^k]^2 \\ &= \frac{\tau}{8\bar{\beta}(\Delta_{\max} + \xi)} \|v_q^k - u^k\|_2^2 \\ &= \frac{\tau}{8\bar{\beta}(\Delta_{\max} + \xi)} \beta(z^k, y_q^k, u^k, v_q^k) \|y_q^k - z^k\|_2^2 \\ &\leq \frac{\tau}{8\bar{\beta}(\Delta_{\max} + \xi)} \bar{\beta}(\delta_q^k)^2 \\ &= \frac{\tau(\delta_q^k)^2}{8(\Delta_{\max} + \xi)}. \end{aligned}$$

Das heißt, für Penalty-Parameter \hat{r} nach (3.84) gilt

$$\begin{aligned} \nabla \Phi_{\hat{r}}(z^k, u^k)^T s_q^k &\leq -\frac{\tau(\delta_q^k)^2}{4(\Delta_{\max} + \xi)} + \sum_{j \in M} \frac{1}{\hat{r}_j} [v_{q,j}^k - u_j^k]^2 \\ &\leq -\frac{\tau(\delta_q^k)^2}{4(\Delta_{\max} + \xi)} + \frac{\tau(\delta_q^k)^2}{8(\Delta_{\max} + \xi)} \\ &= -\frac{\tau(\delta_q^k)^2}{8(\Delta_{\max} + \xi)} \end{aligned}$$

und damit die Behauptung.

q.e.d.

In Lemma 3.3.10 werden zwei Voraussetzungen gefordert, die in praktischen Anwendungen normalerweise schwer oder gar nicht nachzuweisen sind. Erstens wird über $\bar{\beta} \in \mathbb{R}$ gefordert, daß sich in inneren Iterationen die Änderung der dualen Variablen entsprechend der Änderung der primalen Variablen verhält. Diese Forderung kann abgeschwächt werden, was im Anschluß an Satz 3.6 angesprochen wird. Zweitens wird gefordert, daß die Indexmenge $A(r, y_q^k, v_q^k)$ nur Indizes aktiver Restriktionen enthält. Diese Forderung hängt mit der ersten zusammen, d.h. sie wird mit der Abschwächung der ersten Forderung ebenfalls weniger einschränkend.

Im Verlauf einer angenommenen unendlichen inneren Iteration muß bedingt durch einen stetig schrumpfenden Trust-Region-Radius der Abstand zwischen Entwicklungspunkt und Teilproblemlösung ebenfalls kleiner werden. Damit ist mit Satz 3.4 und Lemma 3.3.10 gezeigt, daß der Penalty-Parameter in einer unendlichen inneren

Iterationen beschränkt ist. Um die Endlichkeit der inneren Iterationen zu zeigen, kann folglich ohne Beschränkung der Allgemeinheit ein hinreichend großer konstanter Penalty-Parameter $\hat{r} \in \mathbb{R}^n$ angenommen werden, um den Beweis klarer darzustellen. Sollte der Penalty-Parameter zu Beginn der angenommenen unendlichen inneren Iteration noch nicht hinreichend groß sein, so würde er solange ansteigen, bis er größer oder gleich \hat{r} ist und dann konstant bleiben, was den Beweis nur unwesentlich ändern würde, d.h. die Annahme eines konstanten Penalty-Parameters \hat{r} zum Nachweis der Endlichkeit der inneren Iteration ist zulässig.

Satz 3.5

Sei OP und $k \in \mathbb{N}_0$ gegeben, sowie $\Delta_{\max} > 0$, $\xi > 0$, $\tau > 0$, $0 < \nu < 1$, $\gamma \ll 1$ konstant. Weiter gegeben sei ein Entwicklungspunkt $z^k \in \mathbb{R}^n$, ein Lagrange-Multiplikator $u^k \in \mathbb{R}^m$ mit $u_j^k \geq 0$, $j \in M$, und ein Trust-Region-Radius $\Delta_0^k \in \mathbb{R}^n$ mit $0 < \Delta_{0,i}^k \leq \Delta_{\max}$, $i = 1, \dots, n$, und sei $\Delta_{q+1}^k := \nu \Delta_q^k$. Die LICQ sei erfüllt und die Lösungen (y_q^k, v_q^k, w_q^k) der Teilprobleme $TRSP(z^k, \Delta_q^k)$ seien eindeutig bestimmt für $q \in \mathbb{N}_0$. Sei $s_q^k := \begin{pmatrix} y_q^k - z^k \\ v_q^k - u^k \end{pmatrix}$ und $\delta_q^k := \|y_q^k - z^k\|_2$. Sei $\hat{r} \in \mathbb{R}^m$ konstant und hinreichend groß mit

$$\nabla \Phi_r(z^k, u^k)^T s_q^k \leq -\frac{\tau(\delta_q^k)^2}{8(\Delta_{\max} + \xi)} \quad (3.85)$$

für alle $r \in \mathbb{R}^m$ mit $r_j \geq \hat{r}_j$, $j \in M$. Existiert $\bar{\beta} > 0$ mit $\beta(z^k, y_q^k, u^k, v_q^k) \leq \bar{\beta}$, dann existiert $\bar{q}^k \in \mathbb{N}_0$, so daß

$$\Phi_r \left(\begin{pmatrix} y_q^k \\ v_q^k \end{pmatrix} \right) - \Phi_r \left(\begin{pmatrix} z^k \\ u^k \end{pmatrix} \right) \leq \gamma \nabla \Phi_r \left(\begin{pmatrix} z^k \\ u^k \end{pmatrix} \right)^T s_q^k < 0, \quad (3.86)$$

für alle $q \geq \bar{q}^k$ und $r_j \geq \hat{r}_j$, $j \in M$.

Beweis:

Sei eine unendliche innere Iteration mit dem Iterationsindex q angenommen, d.h. bedingt durch den monoton schrumpfenden Trust-Region-Radius $\Delta_q^k < \Delta_{q+1}^k$ wird die Suchrichtung s_q^k im Verlauf dieser Iteration beliebig klein. Für die dualen Variablen gilt Beschränktheit (Satz 3.3). Ebenfalls ist der Penalty-Parameter beschränkt (Lemma 3.3.10), d.h. es kann im weiteren ohne Einschränkung mit dem angenommenen \hat{r} gearbeitet werden. Sei $r \in \mathbb{R}^m$ mit $r_j \geq \hat{r}_j$, $j \in M$. Aufgrund der Abstiegseigenschaft der Suchrichtungen s_q^k und der stetigen Differenzierbarkeit von $\nabla \Phi_r$ (folgt aus Voraussetzung 2.1) folgt, daß es für jede beliebige positive Konstante, z.B.

$\frac{\tau}{16(1 + \bar{\beta})(\Delta_{\max} + \xi)}$, ein $\hat{q}^k \geq 0$ gibt, für das

$$\begin{aligned}
& \left| \nabla \Phi_r \left(\begin{pmatrix} y_q^k \\ v_q^k \end{pmatrix} \right)^T s_q^k - \nabla \Phi_r \left(\begin{pmatrix} z^k \\ u^k \end{pmatrix} \right)^T s_q^k \right| \\
&= \left| \nabla \Phi_r \left(\left(\begin{pmatrix} z^k \\ u^k \end{pmatrix} + s_q^k \right)^T s_q^k - \nabla \Phi_r \left(\begin{pmatrix} z^k \\ u^k \end{pmatrix} \right)^T s_q^k \right| \\
&\leq \left\| \nabla \Phi_r \left(\left(\begin{pmatrix} z^k \\ u^k \end{pmatrix} + s_q^k \right) - \nabla \Phi_r \left(\begin{pmatrix} z^k \\ u^k \end{pmatrix} \right) \right\|_2 \|s_q^k\|_2 \\
&\leq \frac{\tau}{16(1 + \bar{\beta})(\Delta_{\max} + \xi)} \|s_q^k\|_2^2,
\end{aligned} \tag{3.87}$$

für alle $q \geq \hat{q}^k$ erfüllt ist. Mit (3.82) ist

$$\|v_q^k - u^k\|_2^2 = \beta(z^k, y_q^k, u^k, v_q^k) \|y_q^k - z^k\|_2^2. \tag{3.88}$$

Es folgt

$$\begin{aligned}
\|s_q^k\|_2^2 &= \|y_q^k - z^k\|_2^2 + \|v_q^k - u^k\|_2^2 \\
&= \|y_q^k - z^k\|_2^2 + \beta(z^k, y_q^k, u^k, v_q^k) \|y_q^k - z^k\|_2^2 \\
&= (1 + \beta(z^k, y_q^k, u^k, v_q^k)) \|y_q^k - z^k\|_2^2 \\
&\leq (1 + \bar{\beta}) \|y_q^k - z^k\|_2^2 \\
&= (1 + \bar{\beta}) (\delta_q^k)^2.
\end{aligned} \tag{3.89}$$

Das heißt für $q \geq \hat{q}^k$ ist

$$\left| \nabla \Phi_r \left(\left(\begin{pmatrix} z^k \\ u^k \end{pmatrix} + s_q^k \right)^T s_q^k - \nabla \Phi_r \left(\begin{pmatrix} z^k \\ u^k \end{pmatrix} \right)^T s_q^k \right| \leq \frac{\tau(\delta_q^k)^2}{16(\Delta_{\max} + \xi)}. \tag{3.90}$$

Weiter folgt aus der Stetigkeit und der Abstiegseigenschaft, daß es für ein $\zeta_q^k \in (0; 1)$ ein $\bar{q}^k \geq \hat{q}^k$ gibt, mit

$$\left| \nabla \Phi_r \left(\left(\begin{pmatrix} z^k \\ u^k \end{pmatrix} + \zeta_q^k s_q^k \right)^T s_q^k - \nabla \Phi_r \left(\begin{pmatrix} z^k \\ u^k \end{pmatrix} \right)^T s_q^k \right| \leq \frac{\tau(\delta_q^k)^2}{16(\Delta_{\max} + \xi)}, \tag{3.91}$$

für alle $q \geq \bar{q}^k$. Also ist

$$\nabla \Phi_r \left(\left(\begin{pmatrix} z^k \\ u^k \end{pmatrix} + \zeta_q^k s_q^k \right)^T s_q^k \leq \nabla \Phi_r \left(\begin{pmatrix} z^k \\ u^k \end{pmatrix} \right)^T s_q^k + \frac{\tau(\delta_q^k)^2}{16(\Delta_{\max} + \xi)}, \tag{3.92}$$

für $q \geq \bar{q}^k$. Aus dem Mittelwertsatz der Differentialrechnung folgt weiter, daß mindestens ein $\zeta_q^k \in (0; 1)$ existiert, für das

$$\begin{aligned} \Phi_r \begin{pmatrix} y_q^k \\ v_q^k \end{pmatrix} - \Phi_r \begin{pmatrix} z^k \\ u^k \end{pmatrix} &= \Phi_r \left(\begin{pmatrix} z^k \\ u^k \end{pmatrix} + s_q^k \right) - \Phi_r \begin{pmatrix} z^k \\ u^k \end{pmatrix} \\ &= \nabla \Phi_r \left(\begin{pmatrix} z^k \\ u^k \end{pmatrix} + \zeta_q^k s_q^k \right)^T s_q^k \end{aligned} \quad (3.93)$$

gilt. Aus (3.92) und (3.93) folgt es existiert $\zeta_q^k > 0$ mit

$$\begin{aligned} &\Phi_r \begin{pmatrix} y_q^k \\ v_q^k \end{pmatrix} - \Phi_r \begin{pmatrix} z^k \\ u^k \end{pmatrix} - \gamma \nabla \Phi_r \begin{pmatrix} z^k \\ u^k \end{pmatrix}^T s_q^k \\ &= \nabla \Phi_r \left(\begin{pmatrix} z^k \\ u^k \end{pmatrix} + \zeta_q^k s_q^k \right)^T s_q^k - \gamma \nabla \Phi_r \begin{pmatrix} z^k \\ u^k \end{pmatrix}^T s_q^k \\ &\leq \nabla \Phi_r \begin{pmatrix} z^k \\ u^k \end{pmatrix}^T s_q^k + \frac{\tau(\delta_q^k)^2}{16(\Delta_{\max} + \xi)} - \gamma \nabla \Phi_r \begin{pmatrix} z^k \\ u^k \end{pmatrix}^T s_q^k \\ &= (1 - \gamma) \nabla \Phi_r \begin{pmatrix} z^k \\ u^k \end{pmatrix}^T s_q^k + \frac{\tau(\delta_q^k)^2}{16(\Delta_{\max} + \xi)} \\ &\stackrel{\text{Satz 3.4}}{\leq} -(1 - \gamma) \frac{\tau(\delta_q^k)^2}{8(\Delta_{\max} + \xi)} + \frac{\tau(\delta_q^k)^2}{16(\Delta_{\max} + \xi)} \\ &= (\gamma - 1) \frac{\tau(\delta_q^k)^2}{8(\Delta_{\max} + \xi)} + \frac{\tau(\delta_q^k)^2}{16(\Delta_{\max} + \xi)} \\ &= (\gamma - 1) \frac{\tau(\delta_q^k)^2}{16(\Delta_{\max} + \xi)} \\ &< 0, \end{aligned}$$

für alle $q \geq \bar{q}^k$.

q.e.d.

Ist Bedingung (3.46) erfüllt ist eine innere Iteration bei TRSCP beendet (wegen Schritt 3 im Algorithmus 3.2). Also folgt mit Satz 3.5 die Endlichkeit der Teiliterationen von TRSCP und es folgt die Wohldefiniertheit des Algorithmus. Es können nun beliebige von TRSCP erzeugte Iterationsfolgen betrachtet bzw. angenommen

werden. Da der vorgestellte Algorithmus nur stoppt, wenn ein stationärer Punkt von OP gefunden wurde, wird im weiteren Verlauf stets von einer unendlichen von TRSCP erzeugten Folge von Entwicklungspunkten ausgegangen. Es wird gezeigt, daß diese Folge mindestens einen stationären Häufungspunkt besitzt. Hierfür wird als nächstes gezeigt, daß der Penalty-Parameter der erweiterten Lagrange-Funktion unabhängig vom Iterationsparameter beschränkt ist, wenn die Lösung des Teilproblems von einem stationären Punkt weg beschränkt ist.

Satz 3.6

Sei OP gegeben und $\Delta_{\max} > 0$, $\xi > 0$, $\tau > 0$ konstant. Seien eine von TRSCP erzeugte Folge $\{(z^k, u^k)\}_{k=0,1,2,\dots} \subset \mathbb{R}^{n+m}$ von Entwicklungspunkten und Lagrange-Multiplikatoren, sowie eine zugehörige Folge $\{\Delta^k\}_{k=0,1,2,\dots} \subset \mathbb{R}^n$ von Trust-Region-Radien, mit $0 < \Delta_i^k \leq \Delta_{\max}$, $i = 1, \dots, n$, gegeben. Weiter sei die LICQ für jedes Element und jeden Häufungspunkt der Folge von Entwicklungspunkten erfüllt und die Lösungen (y^k, v^k, w^k) der Teilprobleme $TRSP(z^k, \Delta^k)$ seien bestimmt. Sei $s^k := \begin{pmatrix} y^k - z^k \\ v^k - u^k \end{pmatrix}$ und $\delta^k := \|y^k - z^k\|_2$. Sei $\delta > 0$ gegeben, dann existiert $\bar{r}^\delta \in \mathbb{R}^m$, so daß für jedes k mit $\delta^k \geq \delta$

$$\nabla \Phi_r(z^k, u^k)^T s^k \leq -\frac{\tau(\delta^k)^2}{8(\Delta_{\max} + \xi)} \leq -\frac{\tau\delta^2}{8(\Delta_{\max} + \xi)} < 0 \quad (3.94)$$

erfüllt ist für alle $r \in \mathbb{R}^m$ mit $r_j \geq \bar{r}_j^\delta$, $j \in M$.

Beweis:

Die Voraussetzungen für Satz 3.4 sind erfüllt, d.h. mit (3.78) existiert $r \in \mathbb{R}^m$ mit $r_j > 1$, $j \in M$ und

$$\begin{aligned} \nabla \Phi_r(z^k, u^k)^T s^k &\leq -\frac{\tau(\delta^k)^2}{4(\Delta_{\max} + \xi)} - \sum_{j \in A(r, y^k, v^k)} r_j h_j^2(z^k) \\ &\quad + \sum_{j \in A(r, y^k, v^k)} 2h_j(z^k)[v_j^k - u_j^k] \\ &\quad - \sum_{j \in K(r, y^k, v^k)} \frac{u_j^k}{r_j} [v_j^k - u_j^k]. \end{aligned}$$

Weiter sind die Voraussetzungen von Satz 3.3 erfüllt, d.h. die Lagrange-Multiplikatoren liegen in einer kompakten Menge. Also existiert ein $u_{\max} > 0$ mit $u_j^k, v_j^k \leq u_{\max}$, $j \in M$. Sei für $j \in K(r, y^k, v^k)$ der Penalty-Parameter so groß, daß

$$r_j \geq m \frac{16u_{\max}^2 (\Delta_{\max} + \xi)}{\tau\delta^2} \quad (3.95)$$

erfüllt ist. Dann ist

$$\sum_{j \in K(r, y^k, v^k)} \frac{1}{r_j} \leq \frac{1}{u_{\max}^2} \frac{\tau(\delta^k)^2}{16(\Delta_{\max} + \xi)}.$$

Hieraus folgt, daß diese Bedingung nach einer endlichen Anzahl iterativer Erhöhungen des Penalty-Parameters in TRSCP erfüllt wird. Dies bedeutet weiter, daß die Penalty-Parameter des Terms

$$- \sum_{j \in K(r, y^k, v^k)} \frac{u_j^k}{r_j} (v_j^k - u_j^k)$$

bei TRSCP beschränkt sind, für alle $\delta^k \geq \delta > 0$.

Für die restlichen Penalty-Parameter im Term

$$\sum_{j \in A(r, y^k, v^k)} (2 |h_j(z^k)| u_{\max} - r_j h_j^2(z^k))$$

werden zwei Fälle unterschieden. Für alle $j \in A(r, y^k, v^k)$ mit

$$|h_j(z^k)| \leq \frac{1}{m} \frac{\tau \delta^2}{32 u_{\max} (\Delta_{\max} + \xi)}$$

gilt:

$$\begin{aligned} 2 |h_j(z^k)| u_{\max} - r_j h_j^2(z^k) &\leq 2 |h_j(z^k)| u_{\max} \\ &\leq \frac{1}{m} \frac{\tau \delta^2}{16 (\Delta_{\max} + \xi)}. \end{aligned}$$

Für die restlichen $j \in A(r, y^k, v^k)$, d.h.

$$|h_j(z^k)| > \frac{1}{m} \frac{\tau \delta^2}{32 u_{\max} (\Delta_{\max} + \xi)}$$

ist erfüllt, wird $r_j \geq m \frac{64 u_{\max}^2 (\Delta_{\max} + \xi)}{\tau \delta^2}$ gesetzt, und es gilt:

$$\begin{aligned} &2 |h_j(z^k)| u_{\max} - r_j h_j^2(z^k) \\ &\leq 2 |h_j(z^k)| u_{\max} - m \frac{64 u_{\max}^2 (\Delta_{\max} + \xi)}{\tau \delta^2} h_j^2(z^k) \\ &= |h_j(z^k)| \left(2 u_{\max} - m \frac{64 u_{\max}^2 (\Delta_{\max} + \xi)}{\tau \delta^2} |h_j(z^k)| \right) \\ &\leq |h_j(z^k)| \left(2 u_{\max} - m \frac{64 u_{\max}^2 (\Delta_{\max} + \xi)}{\tau \delta^2} \frac{1}{m} \frac{\tau \delta^2}{32 u_{\max} (\Delta_{\max} + \xi)} \right) \\ &= |h_j(z^k)| (2 u_{\max} - 2 u_{\max}) \\ &= 0. \end{aligned}$$

q.e.d.

Im Beweis zu Satz 3.6 wird gezeigt, daß ein für die Konvergenz des Algorithmus TRSCP hinreichender Abstieg unabhängig vom Iterationsparameter erreicht wird, wenn die Lösung des Teilproblems einen gewissen Abstand vom Entwicklungspunkt nicht unterschreitet. Mit der gleichen Beweisargumentation können die Voraussetzungen von Lemma 3.3.10 abgeschwächt werden. Die strenge Forderung,

$$\text{es existiert in inneren Iterationen } \bar{\beta} > 0 \text{ mit } \beta(z^k, y_q^k, u^k, v_q^k) \leq \bar{\beta}, \quad (3.96)$$

kann mit Hilfe eines festen $\delta > 0$ eingeschränkt werden auf Bereiche, für die $\delta^k < \delta$ gilt. Entsprechend muß

$$j \in A(r, y_q^k, v_q^k) \Leftrightarrow h_j(y_q^k) = 0 \quad (3.97)$$

auch nur hier gelten. Mit einem hinreichend klein gewählten $\delta > 0$ sollte die Voraussetzung weniger problematisch sein.

Unter den getroffenen Voraussetzungen wurde gezeigt, daß die von TRSCP erzeugte Folge von Entwicklungspunkten in einer kompakten Menge liegt und die erzeugten Penalty-Parameter der von TRSCP verwendeten erweiterten Lagrange-Funktion beschränkt sind. Diese Beschränktheit ist zudem unabhängig vom Iterationsparameter, wenn der Entwicklungspunkt von einem stationären Punkt weg beschränkt ist. Weiter ist der Wert der erweiterten Lagrange-Funktion nach unten beschränkt und mit einer unendlichen Teilfolge von Entwicklungspunkten kann jeweils ein für die Konvergenz hinreichender Abstieg in der erweiterten Lagrange-Funktion erzielt werden. Außerdem kann gezeigt werden, daß diese Folge von Entwicklungspunkten eine konvergente Teilfolge besitzt.

Satz 3.7

Sei OP gegeben und $\Delta_{\max} > 0$, $\xi > 0$, $\tau > 0$, $0 < \gamma \ll 1$ konstant. Seien die Folgen $\{(z^k, u^k)\}_{k=0,1,2,\dots} \subset \mathbb{R}^{n+m}$ mit $u^k \in V$ kompakt, $\{\Delta^k\}_{k=0,1,2,\dots} \subset \mathbb{R}^n$ mit $0 < \Delta_i^k \leq \Delta_{\max}$, $i = 1, \dots, n$, und $\{r_0^k\}_{k=0,1,2,\dots} \subset \mathbb{R}^m$ mit $r_{j,0}^k \geq 1$, $j \in M$, von TRSCP erzeugt. Die LICQ sei in jedem Glied und in jedem Häufungspunkt der Folge von Entwicklungspunkten erfüllt. Weiter seien die Lösungen (y^k, v^k, w^k) der Teilprobleme $TRSP(z^k, \Delta^k)$ bestimmt und $s^k := \begin{pmatrix} y^k - z^k \\ v^k - u^k \end{pmatrix}$. Dann existiert eine Teilfolge $\{(z^{k_i}, u^{k_i})\}_{i=0,1,2,\dots} \subset \mathbb{R}^{n+m}$, bei der es für jedes $\delta > 0$ mindestens ein $\hat{i} \in \mathbb{N}_0$ gibt, für das

$$\|z^{k_{i+1}} - z^{k_i}\|_{\infty} \leq \delta \quad (3.98)$$

für alle $i \geq \hat{i}$ erfüllt ist.

Beweis:

Annahme:

Es existiert für $\delta > 0$ kein $\hat{i} \in \mathbb{N}_0$ mit $\|z^{k_{i+1}} - z^{k_i}\|_{\infty} \leq \delta$ für $i \geq \hat{i}$.

Wegen Satz 3.6 und der Penalty-Parameter-Anpassung in TRSCP existiert $\bar{k} \geq 0$ und $\bar{r} \geq 1$ mit $r_0^k = \bar{r}$ für alle $k \geq \bar{k}$. Sei o.B.d.A. $r_0^k = \bar{r}$, für $k \in \mathbb{N}_0$. Dann folgt aus Satz 3.6, daß

$$\nabla \Phi_{\bar{r}}(z^k, u^k)^T s^k \leq -\frac{\tau \delta^2}{8(\Delta_{\max} + \xi)} \quad (3.99)$$

gilt für alle $k \in \mathbb{N}_0$. Es gilt

$$k_{i+1} \geq k_i + 1 > k_i$$

und wegen Schritt 3 in Algorithmus 3.2

$$\Phi_{\bar{r}}(z^{k+1}, u^{k+1}) < \Phi_{\bar{r}}(z^k, u^k).$$

Also ist

$$\begin{aligned} \Phi_{\bar{r}} \begin{pmatrix} z^{k_{i+1}} \\ v^{k_{i+1}} \end{pmatrix} - \Phi_{\bar{r}} \begin{pmatrix} z^{k_i} \\ u^{k_i} \end{pmatrix} &\leq \Phi_{\bar{r}} \begin{pmatrix} z^{k_{i+1}} \\ v^{k_{i+1}} \end{pmatrix} - \Phi_{\bar{r}} \begin{pmatrix} z^{k_i} \\ u^{k_i} \end{pmatrix} \\ &\stackrel{\text{Alg. 3.2 Schritt 3}}{\leq} \gamma \nabla \Phi_{\bar{r}} \begin{pmatrix} z^{k_i} \\ u^{k_i} \end{pmatrix}^T s^{k_i} \\ &\leq -\gamma \frac{\tau \delta^2}{8(\Delta_{\max} + \xi)}. \end{aligned}$$

Das allerdings führt zu $\lim_{i \rightarrow \infty} \Phi_{\bar{r}} \begin{pmatrix} z^{k_i} \\ u^{k_i} \end{pmatrix} = -\infty$ und zu einem Widerspruch zu Folgerung 3.3.9. Damit ist die Annahme widerlegt und die Behauptung bewiesen.

q.e.d.

Wie aus Schittkowski [46] ersichtlich, ist die Aussage von Satz 3.7 notwendig um die globale Konvergenzaussage nachzuweisen. Zum Schluß dieses Abschnitts folgt der Nachweis der globalen Konvergenzeigenschaft des Algorithmus TRSCP. Die Aussage ist entsprechend Zillober [74] bzw. analog Lehmann [30] und folgt der Beweisführung in Schittkowski [46].

Satz 3.8

Sei OP gegeben und $\Delta_{\max} > 0$, $\xi > 0$, $\tau > 0$, $0 < \gamma \ll 1$ konstant. Seien die Folge von Entwicklungspunkten $\{(z^k, u^k)\}_{k=0,1,2,\dots} \subset \mathbb{R}^{n+m}$, die Folge von Trust-Region-Radien $\{\Delta^k\}_{k=0,1,2,\dots} \subset \mathbb{R}^n$ und die Folge von Penalty-Parametern $\{r_0^k\}_{k=0,1,2,\dots} \subset \mathbb{R}^m$ von TRSCP erzeugt. Die LICQ sei in jedem Entwicklungspunkt, sowie in jedem Häufungspunkt der Folge von Entwicklungspunkten erfüllt, und die Lösungen (y^k, v^k, w^k) der Teilprobleme $TRSP(z^k, \Delta^k)$ seien bestimmt. Dann besitzt die Folge $\{(z^k, u^k)\}_{k=0,1,2,\dots}$ mindestens einen Häufungspunkt, der der KKT-Bedingung von OP genügt.

Beweis:

Es gilt aufgrund Voraussetzung 2.2, daß ein $b < \infty$ existiert mit $z_i^k \leq b$, $i = 1, \dots, n$. Mit $X := \{x \in \mathbb{R}^n : |x_i| \leq b, i = 1, \dots, n\}$ gilt $z^k \in X$ kompakt für alle $k \in \mathbb{N}_0$. Wegen $\Delta^k \in \mathbb{R}^n$ von TRSCP erzeugt gilt $\Delta_i^k > 0$, $i = 1, \dots, n$, und somit werden die Voraussetzungen von Satz 3.3 erfüllt. Das heißt, es existiert $V \subset \mathbb{R}^m$ kompakt mit $u^k \in V$, ebenfalls für alle $k \in \mathbb{N}_0$. Zusammen folgt, daß TRSCP eine Iterationsfolge von Entwicklungspunkten auf einem Kompaktum erzeugt. Mit Satz 3.7 und der

Kompaktheit der Menge $X \times V$ folgt, daß mindestens ein Häufungspunkt (x^*, u^*) und eine Teilfolge $\{(z^{k_t}, u^{k_t})\}_{t=0,1,2,\dots}$ existiert, so daß

$$\lim_{t \rightarrow \infty} \|z^{k_{t+1}} - z^{k_t}\|_{\infty} = 0, \quad (3.100)$$

$$\lim_{t \rightarrow \infty} z^{k_t} = x^*, \quad (3.101)$$

$$\lim_{t \rightarrow \infty} u^{k_t} = u^*, \quad (3.102)$$

erfüllt ist. Um zu zeigen, daß (x^*, u^*) die KKT-Bedingung von OP erfüllt, d.h.

$$\nabla_x f(x^*) + \sum_{j \in M} u_j^* \nabla_x h_j(x^*) = 0, \quad (3.103)$$

$$h_j(x^*) \leq 0, \quad j \in M, \quad (3.104)$$

$$u_j^* h_j(x^*) = 0, \quad j \in M, \quad (3.105)$$

$$u_j^* \geq 0, \quad j \in M, \quad (3.106)$$

werden die KKT-Bedingungen der Teilprobleme (Bemerkung 3.3.2) untersucht, d.h.

$$\begin{aligned} \nabla_x f^{k_t}(y^{k_t}, z^{k_t}, \Delta^{k_t}, \xi, \tau) &+ \sum_{j=1}^m v_j^{k_t} \nabla_x h_j^{k_t}(y^{k_t}, z^{k_t}, \Delta^{k_t}, \xi) \\ &+ \sum_{i=1}^{2n} w_i^{k_t} \nabla_x g_i^{k_t}(y^{k_t}, z^{k_t}, \Delta^{k_t}) = 0, \end{aligned} \quad (3.107)$$

$$h_j^{k_t}(y^{k_t}, z^{k_t}, \Delta^{k_t}, \xi) \leq 0, \quad j \in M, \quad (3.108)$$

$$g_i^{k_t}(y^{k_t}, z^{k_t}, \Delta^{k_t}) \leq 0, \quad i = 1, \dots, 2n, \quad (3.109)$$

$$v_j^{k_t} h_j^{k_t}(y^{k_t}, z^{k_t}, \Delta^{k_t}, \xi) = 0, \quad j \in M, \quad (3.110)$$

$$w_i^{k_t} g_i^{k_t}(y^{k_t}, z^{k_t}, \Delta^{k_t}) = 0, \quad i = 1, \dots, 2n, \quad (3.111)$$

$$v_j^{k_t} \geq 0, \quad j \in M, \quad (3.112)$$

$$w_i^{k_t} \geq 0, \quad i = 1, \dots, 2n, \quad (3.113)$$

ist jeweils erfüllt für alle $t \in \mathbb{N}_0$. Durch die Annäherung von Entwicklungspunkt und Häufungspunkt ((3.100) bis (3.102)) existiert, bedingt durch die Erhöhung des Trust-Region-Radius im vierten Schritt von Algorithmus TRSCP, ein $\bar{t} \geq 0$ mit

$\delta^{kt} < \Delta_{\min}^k$ und $w_i^{kt} = 0$ für alle $i = 1, \dots, 2n$ und $t \geq \bar{t}$. Es gilt für $j \in M$:

$$\begin{aligned} & |h_j^{kt}(y^{kt}, z^{kt}, \Delta^{kt}, \xi) - h_j(z^{kt})| \\ & \stackrel{(3.29)}{=} \left| h_j(z^{kt}) - \sum_{i=1}^n \left| \frac{\partial h_j(z^{kt})}{\partial x_i} \right| (\Delta_i^{kt} + \xi) + \sum_{i \in I_j^+(z^{kt})} \frac{\left| \frac{\partial h_j(z^{kt})}{\partial x_i} \right| (\Delta_i^{kt} + \xi)^2}{(z_i^{kt} + \Delta_i^{kt} + \xi) - y_i^{kt}} \right. \\ & \quad \left. + \sum_{i \in I_j^-(z^{kt})} \frac{\left| \frac{\partial h_j(z^{kt})}{\partial x_i} \right| (\Delta_i^{kt} + \xi)^2}{y_i^{kt} - (z_i^{kt} - \Delta_i^{kt} - \xi)} - h_j(z^{kt}) \right|. \end{aligned}$$

Für $|y^{kt} - z^{kt}| \rightarrow 0$ (siehe (3.100)) folgt

$$(z_i^{kt} + \Delta_i^{kt} + \xi) - y_i^{kt} \rightarrow \Delta_i^{kt} + \xi$$

und

$$y_i^{kt} - (z_i^{kt} - \Delta_i^{kt} - \xi) \rightarrow \Delta_i^{kt} + \xi.$$

Insgesamt folgt $|h_j^{kt}(y^{kt}, z^{kt}, \Delta^{kt}, \xi) - h_j(z^{kt})| \rightarrow 0$. Das heißt mit (3.108) und (3.101) folgt (3.104) für alle $j \in M$. Aus (3.112) und (3.102) folgt (3.106). Zusammen folgt mit (3.110) Bedingung (3.105). Es ist für $i = 1, \dots, n$:

$$\begin{aligned} & \left| \frac{\partial f^{kt}(y^{kt}, z^{kt}, \Delta^{kt}, \xi, \tau)}{\partial x_i} - \frac{\partial f(z^{kt})}{\partial x_i} \right| \\ & \stackrel{(3.28)}{=} \left| \frac{\partial f(z^{kt})}{\partial x_i} + \right. \\ & \quad \left. \left\{ \begin{array}{l} \left(\frac{\partial f(z^{kt})}{\partial x_i} + \tau \right) (y_i^{kt} - z_i^{kt}) \frac{2(z_i^{kt} + \Delta_i^{kt} + \xi) - z_i^{kt} - y_i^{kt}}{((z_i^{kt} + \Delta_i^{kt} + \xi) - y_i^{kt})^2} \right. \\ \quad \left. \text{(falls } i \in I_0^+(z^{kt}) \text{)} \right. \\ \left. \left(\frac{\partial f(z^{kt})}{\partial x_i} - \tau \right) (y_i^{kt} - z_i^{kt}) \frac{2(z_i^{kt} - \Delta_i^{kt} - \xi) - z_i^{kt} - y_i^{kt}}{(y_i^{kt} - (z_i^{kt} - \Delta_i^{kt} - \xi))^2} \right. \\ \quad \left. \text{(falls } i \in I_0^-(z^{kt}) \text{)} \right. \end{array} \right\} \\ & \quad \left. - \frac{\partial f(z^{kt})}{\partial x_i} \right| \end{aligned}$$

Mit Voraussetzung 2.2, Lemma 2.3.6, $\tau > 0$ konstant, $\xi > 0$ konstant und Bedingung (3.100), d.h. $|y^{kt} - z^{kt}| \rightarrow 0$ folgt $|\nabla_x f^{kt}(y^{kt}, z^{kt}, \Delta^{kt}, \tau, \xi) - \nabla_x f(z^{kt})| \rightarrow 0$. Analog folgt für die Nebenbedingungen $|\nabla_x h_j^{kt}(y^{kt}, z^{kt}, \Delta^{kt}, \xi) - \nabla_x h_j(z^{kt})| \rightarrow 0$ für alle $j \in M$. Insgesamt gilt die Bedingung (3.103) und die globale Konvergenzaussage für TRSCP ist bewiesen.

q.e.d.

Kapitel 4

Numerische Resultate

Dieses Kapitel beschreibt die Umsetzung des Algorithmus TRSCP in das Programm TRSCP1.0 und enthält Resultate des Algorithmus bei verschiedenen Testproblemen. Außerdem wird TRSCP mit dem in Kapitel 2 vorgestellten SCP-Verfahren von Zillober [74] bezüglich des benötigten Aufwands zur Lösung von Testproblemen verglichen, d.h. es wird betrachtet, wie viele Funktions- und Gradientenauswertungen jeder Algorithmus für die Lösung eines Testproblems benötigt. In Anlehnung an die Bezeichnung TRSCP (Trust-Region Sequential Convex Programming) wird im folgenden das SCP-Verfahren aus Kapitel 2 mit LSSCP (Line-Search Sequential Convex Programming) bezeichnet.

Das Programm TRSCP1.0 ist eine Weiterentwicklung des von Zillober [76, 77] entwickelten Programms SCPIP3.0 durch die Implementierung eines Trust-Region SCP-Verfahrens basierend auf dem Algorithmus aus Abschnitt 3.2. Hieraus folgt, daß ein direkter Vergleich zwischen LSSCP und TRSCP möglich ist.

Für die Lösung des Teilproblems kommt stets die von Zillober [71, 75] programmierte Innere-Punkte-Methode aus dem Programm SCPIP3.0 [76, 77] zum Einsatz. Die Möglichkeit, das Teilproblem über das duale Problem unter Ausnutzung der Separabilität zu lösen, wird nicht betrachtet. Die Art und Weise, wie das Teilproblem gelöst wird, ist für den Vergleich der Algorithmen 2.4 (LSSCP) und 3.2 (TRSCP) nicht relevant. Zillober [74] beschreibt ausführlich die implementierte Innere-Punkte-Methode und die Lösung des Teilproblems. Weiter wird auf Zillober [69] verwiesen, in dem eine Lösung des Teilproblems unter Ausnutzung der Separabilität nachgelesen werden kann.

Für LSSCP kommt die Asymptotenerzeugung nach Algorithmus 2.3 zum Einsatz. Eine andere Methode zur Asymptotenerzeugung bei LSSCP, wie zum Beispiel von Schenk [45] vorgeschlagen, wird in dieser Arbeit nicht betrachtet. Auswirkungen des Teilproblemlösers und der Asymptotenwahl für LSSCP sind unter anderem in Zillober [74] ausgeführt.

Der nächste Abschnitt beschreibt die Umsetzung des Algorithmus 3.2 im Programm TRSCP1.0. Im Anschluß daran folgt der Vergleich der Algorithmen TRSCP und

LSSCP im Programm TRSCP1.0. Als Testprobleme für dieses Kapitel dienen zum einen Teil die Sammlungen von Hock und Schittkowski [27] und Schittkowski [49] und zum anderen Teil Strukturoptimierungsbeispiele. Die Strukturoptimierungsprobleme wurden in Sigmund [54] als Übungen für den Matlab-Code aus Sigmund [53] herangezogen. Diese Beispiele der Topologieoptimierung basieren auf einfachen Strukturen, wie beispielsweise dem MBB-Beam¹. Es werden verschiedene Diskretisierungen und Massen der zu optimierenden Strukturen betrachtet.

4.1 Implementierungsdetails

In diesem Abschnitt wird beschrieben, mit welchen im Vergleich zu Algorithmus 3.2 zusätzlichen Änderungen der Algorithmus TRSCP im Programm TRSCP1.0 implementiert wurde. TRSCP1.0 basiert auf dem Programm SCIP3.0 [76, 77]. Bei der Integration des Algorithmus TRSCP in SCIP3.0 wurde darauf geachtet, daß TRSCP viele der im Programm SCIP3.0 vorhandenen Eigenschaften, die teilweise die Leistungsfähigkeit von LSSCP steigern, nutzen kann. Dies ermöglicht später den Vergleich zwischen den Algorithmen 3.2 (TRSCP) und 2.4 (LSSCP) so genau wie möglich durchzuführen.

In Abschnitt 3.2 ist von TRSCP eine Basisversion vorgestellt worden. Hier wurde, um die globale Konvergenzeigenschaft anschaulicher nachweisen zu können, auf leistungssteigernde Teile verzichtet, die für die globale Konvergenzeigenschaft nicht notwendig sind. Im Gegensatz dazu wird in diesem Abschnitt der Algorithmus TRSCP so vorgestellt, wie er im Programm TRSCP1.0 implementiert ist. Auf Veränderungen, die Einfluß auf den Beweis der globalen Konvergenzeigenschaft haben könnten, wird jeweils im Anschluß an die vorgestellte Änderung eingegangen.

Eine der wichtigsten Modifikationen im Algorithmus TRSCP in diesem Abschnitt ist, für die aktuelle Lösung des Teilproblems eine Prognose zu erstellen, die die Entwicklung des Zielfunktionswerts vorhersagt. Diese wird anschließend mit der tatsächlichen Änderung verglichen. Dies geschieht entsprechend anderer Trust-Region-Verfahren über die Berechnung des Quotienten $Q^k \in \mathbb{R}$ aus erwarteter Verbesserung und erreichter Verbesserung, siehe z.B. Algorithmus 3.1. Die Änderung des Trust-Region-Radius wird analog anderer Trust-Region-Verfahren abhängig vom Ergebnis dieses Vergleichs gemacht.

Die Prognose wird bei TRSCP mittels einer linearen Extrapolation durchgeführt und die tatsächliche Änderung wird über den Vergleich der Funktionswerte der erweiterten Lagrange-Funktion bestimmt, d.h. es wird

$$Q^k := \frac{\Phi_{r_0^k} \begin{pmatrix} z^{k+1} \\ u^{k+1} \end{pmatrix} - \Phi_{r_0^k} \begin{pmatrix} z^k \\ u^k \end{pmatrix}}{\nabla \Phi_{r_0^k} \begin{pmatrix} z^k \\ u^k \end{pmatrix}^T s^k} \quad (4.1)$$

¹Ein MBB-Beam ist ein Bauteil mit sechsfacher Länge wie Höhe, das an seinen beiden Enden unten fixiert ist und in der Mitte unten belastet wird.

berechnet.

In Abschnitt 3.2 wird das Optimierungsproblem OP ohne explizite Schranken für die primalen Variablen und ohne Gleichungsnebenbedingungen vorgestellt. In den betrachteten Testproblemen sind solche Schranken, zumindest für einzelne Koeffizienten, und Gleichungsnebenbedingungen durchaus üblich. Für diese Fälle wird OP entsprechend erweitert und folgende Indexmengen werden definiert. Seien die Indexteilmengen

$$\begin{aligned} I_u &\subseteq \{1, \dots, n\}, \\ I_o &\subseteq \{1, \dots, n\} \end{aligned} \quad (4.2)$$

gegeben. Dann ist das erweiterte Optimierungsproblem EOP gegeben über

$$\begin{aligned} \text{(EOP)} \quad & \text{Minimiere} \quad f(x), \\ & \text{unter} \quad h_j(x) = 0, \quad j = 1, \dots, m_g, \\ & \quad h_j(x) \leq 0, \quad j = m_g + 1, \dots, m, \\ & \quad g_i(x) \leq 0, \quad i \in I_u, \\ & \quad g_{n+i}(x) \leq 0, \quad i \in I_o, \\ & \quad x \in \mathbb{R}^n, \end{aligned} \quad (4.3)$$

mit

$$\begin{aligned} g_i(x) &:= -x_i + \underline{x}_i, \quad i \in I_u, \\ g_{n+i}(x) &:= x_i - \bar{x}_i, \quad i \in I_o. \end{aligned} \quad (4.4)$$

Ein EOP ist ein NLP mit zusätzlichen Schranken-Nebenbedingungen. Wie Schranken im Nachweis der globalen Konvergenzeigenschaft berücksichtigt werden müssen, kann in Lehmann [30] nachgelesen werden. Änderungen am globalen Konvergenzbeweis, die durch die Aufnahme von Gleichungsrestriktionen entstehen, sind in Zillober [74] ausgeführt.

Durch die Erweiterung des Optimierungsproblems muß im Teilproblem der Trust-Region-Radius mit den Schranken abgestimmt werden. Dies geschieht über die Anpassung der betroffenen Nebenbedingungen $g_i : \mathbb{R}^n \rightarrow \mathbb{R}$, $i \in I_u$, und $g_{n+i} : \mathbb{R}^n \rightarrow \mathbb{R}$, $i \in I_o$, d.h. diese Funktionen werden im Teilproblem mit

$$\begin{aligned} g_i^k(x, z^k, \Delta^k, \underline{x}_i) &:= -x_i + \max\{\underline{x}_i, z_i^k - \Delta_i^k\} \leq 0, \quad \text{für } i \in I_u, \\ g_{n+i}^k(x, z^k, \Delta^k, \bar{x}_i) &:= x_i - \min\{\bar{x}_i, z_i^k - \Delta_i^k\} \leq 0, \quad \text{für } i \in I_o, \end{aligned} \quad (4.5)$$

ersetzt. Für alle anderen Koeffizienten bleiben die Nebenbedingungsfunktionen im Teilproblem zum Einhalt der Trust-Region unverändert. Seien

$$\begin{aligned} n_u &:= |I_u|, \\ n_o &:= |I_o|, \end{aligned} \quad (4.6)$$

die Anzahlen der Indizes in den Indexteilmengen I_u und I_o . Durch die Einführung von Schranken ändert sich die Dimension der dualen Variablen, d.h. der Lagrange-Multiplikator ist $u \in \mathbb{R}^{m+n_u+n_o}$. Sei

$$\begin{aligned} \underline{m} &:= m + n_u + n_o, \\ \underline{n} &:= 2n - n_u - n_o. \end{aligned} \quad (4.7)$$

Durch diese Änderung wird mit dem Vektor $\underline{h}(x) \in \mathbb{R}^m$, $x \in \mathbb{R}^n$, der aus den Nebenbedingungswerten $h_j(x)$, $j \in M$, $g_i(x)$, $i \in I_u$, sowie $g_{n+i}(x)$, $i \in I_o$, zusammengesetzt ist und den Indexmengen

$$\begin{aligned} \underline{A}(r, x, u) &:= \{j = 1, \dots, m_g\} \cup \\ &\quad \left\{ j = m_g + 1, \dots, \underline{m} : \underline{h}_j(x) \geq -\frac{u_j}{r_j} \right\}, \\ \underline{K}(r, x, u) &:= \{1, \dots, \underline{m}\} \setminus \underline{A}(r, x, u), \end{aligned} \quad (4.8)$$

mit $r \in \mathbb{R}^m$, $x \in \mathbb{R}^n$ und $u \in \mathbb{R}^m$, die Notation vereinfacht.

Zu Beginn im Schritt 0 des Algorithmus TRSCP muß eine Schätzung $z^0 \in \mathbb{R}^n$, $u^0 \in \mathbb{R}^m$ für eine Lösung von EOP gewählt werden. Dieses z^0 sollte zumindest die Schranken für die primale Variable einhalten. Um mit einem willkürlichen $x^{\text{Eingabe}} \in \mathbb{R}^n$ beginnen zu können, wird der erste Entwicklungspunkt durch

$$z_i^0 := \begin{cases} \min\{\max\{x_i^{\text{Eingabe}}; \underline{x}_i\}; \bar{x}_i\}, & \text{falls } i \in I_u \cap I_o, \\ \max\{x_i^{\text{Eingabe}}; \underline{x}_i\}, & \text{falls } i \in I_u \setminus I_o, \\ \min\{x_i^{\text{Eingabe}}; \bar{x}_i\}, & \text{falls } i \in I_o \setminus I_u, \\ x_i^{\text{Eingabe}}, & \text{sonst,} \end{cases} \quad (4.9)$$

gesetzt. Die Koeffizienten der dualen Variablen $u^0 \in \mathbb{R}^m$ werden im Schritt 0 gleichmäßig mit Null initialisiert.

Für den Algorithmus TRSCP muß die Konstante $\Delta_{\max} \in \mathbb{R}$ festgelegt werden. Für die Implementierung hat es sich als vorteilhaft herausgestellt, für jeden Koeffizienten des Trust-Region-Radius eine individuelle Schranke zu setzen. Sei EOP, $z^0 \in \mathbb{R}^n$ nach (4.9) und eine Konstante $T_1 > 0$ gegeben. Dann ist $\Delta^{max} \in \mathbb{R}^n$ definiert durch

$$\Delta_i^{max} := \begin{cases} \min\{\max\{T_1; |z_i^0|\}; \bar{x}_i - \underline{x}_i\}, & \text{falls } i \in I_u \cap I_o, \\ \max\{T_1; |z_i^0|\}, & \text{sonst,} \end{cases} \quad (4.10)$$

und der maximale Trust-Region-Radius $\Delta_{\max} \in \mathbb{R}$ ist definiert durch

$$\Delta_{\max} := \|\Delta^{max}\|_{\infty}. \quad (4.11)$$

Der Trust-Region-Startradius $\Delta^0 \in \mathbb{R}^n$ wird im Programm TRSCP1.0 über

$$\Delta_i^0 := \begin{cases} \min\{\Delta_i^{max}; \max\{T_2; |z_i^0|\}; \lambda(\bar{x}_i - \underline{x}_i)\}, & \text{falls } i \in I_u \cap I_o, \\ \min\{\Delta_i^{max}; \max\{T_2; |z_i^0|\}\}, & \text{sonst,} \end{cases} \quad (4.12)$$

für $i = 1, \dots, n$, $\lambda \in (0; 1)$ und $T_2 > 0$ gesetzt.

Durch $T_1 > 0$ wird verhindert, daß ein Δ_i^{max} zu klein wird, beispielsweise für z_i^0 identisch Null. Mit der hier vorgeschlagenen Vorgehensweise wird versucht ein für

die Problemstellung geeignetes Δ_{\max} bzw. Δ^{\max} automatisch zu generieren. Mit der Konstante $T_2 > 0$ wird die Erzeugung des Startradius flexibler und unabhängiger von Δ^{\max} . Die Konstante Δ_{\max} kommt z.B. bei der Aktualisierung der Penalty-Parameter der erweiterten Lagrange-Funktion zum Einsatz. Der konstante Vektor Δ^{\max} wird hingegen z.B. bei der Anpassung des Trust-Region-Radius eingesetzt, um jeden Koeffizienten des Trust-Region-Radius individuell nach oben zu beschränken.

Der Algorithmus TRSCP führt, um den Abstieg in der Merit-Funktion zu garantieren, falls notwendig innere Iterationen durch, d.h. Iterationen, bei denen der Entwicklungspunkt nicht gewechselt und nur der Trust-Region-Radius verkleinert wird. Bei den betrachteten Testproblemen ist es vorteilhaft, die Anzahl der inneren Iterationen zwischen dem Wechsel des Entwicklungspunkts zu beschränken. Außerdem ist es im Fall des Abbruchs einer Serie von inneren Iterationen vorteilhaft, den Trust-Region-Radius wieder auf ein gewisses Niveau anzuheben, ähnlich wie dies im vierten Schritt des Basisalgorithmus gemacht wird. Nachdem die Serie von inneren Iterationen nach $q \in \mathbb{N}$ aufeinanderfolgenden inneren Iterationen abgebrochen wird, wird der Entwicklungspunkt mit der aktuellen Lösung des Teilproblems ersetzt und der Trust-Region-Radius auf den Wert wie zu Beginn der inneren Iteration gesetzt, d.h.

$$\Delta_0^{k+1} := \Delta_0^k. \quad (4.13)$$

Die Beschänkung der Anzahl von inneren Iterationen zwischen zwei äußeren Iterationen ähnelt der Vorgehensweise in SCIP3.0 bezüglich der Line-Search. Hier ist die Anzahl von Teiliterationen mit 10 beschränkt. Ist eine Line-Search im Programm SCIP3.0 nicht erfolgreich, d.h. es kann mit Hilfe der Line-Search kein neuer Iterationspunkt gefunden werden, der einen hinreichenden Abstieg in der erweiterten Lagrange-Funktion erzielt, wird in SCIP3.0 die aktuelle Lösung des Teilproblems als nächster Iterationspunkt herangezogen. Für den Nachweis der globalen Konvergenzeigenschaft sowohl von TRSCP als auch von LSSCP muß allerdings weiterhin die Anzahl der inneren Iterationen bzw. der Line-Search-Schritte als unbeschränkt angenommen werden.

Beim Algorithmus 3.2 (TRSCP) bzw. 2.4 (LSSCP) endet die Iteration, falls die primale Variable der Lösung eines Teilproblems identisch mit dem Entwicklungspunkt des Teilproblems ist. Dieser exakte Abbruch ist im Programm TRSCP1.0 abgeschwächt. Hierfür muß die Lagrange-Funktion $L : \mathbb{R}^{n+m} \rightarrow \mathbb{R}$ von EOP ausgewertet werden, d.h.

$$L(x, u) := f(x) + \sum_{j=1}^m u_j h_j(x) \quad (4.14)$$

wird berechnet.

Definition 4.1.1

Sei EOP geben und $k \in \mathbb{N}_0$, $\varepsilon > 0$ konstant. Weiter sei ein Entwicklungspunkt $z^k \in \mathbb{R}^n$ und ein Trust-Region-Radius $\Delta^k \in \mathbb{R}^n$ gegeben. Sei die Lösung (y^k, v^k, w^k) des Teilproblems $TRSP(z^k, \Delta^k)$ mit $y^k \in \mathbb{R}^n$, $v^k \in \mathbb{R}^m$ und $w^k \in \mathbb{R}^m$ bestimmt.

Der im Programm TRSCP1.0 implementierte Algorithmus TRSCP beendet eine Iteration erfolgreich, falls die Bedingungen

$$\|\nabla L(y^k, v^k)\|_\infty \leq \varepsilon, \quad (4.15)$$

$$|\underline{h}_j(y^k)| \leq \varepsilon, \quad j = 1, \dots, m_g, \quad (4.16)$$

$$\underline{h}_j(y^k) \leq \varepsilon, \quad j = m_g + 1, \dots, \underline{m}, \quad (4.17)$$

gleichzeitig erfüllt sind.

Für den Algorithmus 2.4 kann Definition 4.1.1 geeignet angepaßt bzw. umformuliert werden (siehe Zillober [74]). Beim Vergleich der zwei Algorithmen mit dem Programm TRSCP1.0 wird obiges Abbruchkriterium mit einer festzulegenden Genauigkeitsforderung von höchstens $\varepsilon := 10^{-7}$ ausgewertet. Die Genauigkeitsforderung hängt von der Klasse von Problemen ab, die gelöst werden soll.

In Schritt 2 von Algorithmus 3.2 (TRSCP) wird der Penalty-Parameter der erweiterten Lagrange-Funktion mit Hilfe des Parameters κ angepaßt. Hierbei kann der Penalty-Parameter nur gleichmäßig in allen Koeffizienten erhöht werden. Eine erweiterte Strategie aus Zillober [74] erlaubt ein koeffizientenweises Anpassen des Penalty-Parameters.

Definition 4.1.2 (Penalty-Parameter-Update)

Sei EOP geben, $k \in \mathbb{N}_0$ und $1 < \kappa_1 < \kappa_2$ konstant. Weiter gegeben seien ein Teilproblem TRSP(z^k, Δ^k) mit $z^k \in \mathbb{R}^n$ und $\Delta^k \in \mathbb{R}^n$, sowie ein geeigneter Lagrange-Multiplikator $u^k \in \mathbb{R}^m$. Ferner gegeben sei die Lösung (y^k, v^k, w^k) des Teilproblems mit $y^k \in \mathbb{R}^n$, $v^k \in \mathbb{R}^m$ und $w^k \in \mathbb{R}^n$ und die Abstiegsrichtung $s^k := \begin{pmatrix} y^k - z^k \\ v^k - u^k \end{pmatrix}$.

Im Programm TRSCP1.0 werden die Penalty-Parameter $r_p^{k-1} \in \mathbb{R}^m$, $p \in \mathbb{N}_0$, über folgende Vorschrift aktualisiert:

Bestimme

$$r_{p+1,j}^{k-1} := \min \left\{ \kappa_2 r_{p,j}^{k-1}, \max \left\{ \kappa_1 r_{p,j}^{k-1}, \left| \frac{2(v_j^k - u_j^k)}{\underline{h}_j(z^k)} \right| \right\} \right\}, \quad (4.18)$$

für alle $j \in \underline{A}(r_p^{k-1}, z^k, u^k)$, falls $\underline{h}_j(z^k) > 0$ und $\nabla \underline{h}_j(z^k)^T (y^k - z^k) \neq 0$ bzw. falls $\underline{h}_j(z^k) < 0$ und $\nabla \underline{h}_j(z^k)^T (y^k - z^k) > 0$.

Anderenfalls setze

$$r_{p+1,j}^{k-1} := \kappa_1 r_{p,j}^{k-1}. \quad (4.19)$$

Weiter bestimme

$$r_{p+1,j}^{k-1} := \min \left\{ \kappa_2 r_{p,j}^{k-1}, \max \left\{ \kappa_1 r_{p,j}^{k-1}, \left| \frac{u_j^k (v_j^k - u_j^k) \delta \underline{m}(\Delta_i^k + \xi)}{\tau(\delta^k)^2} \right| \right\} \right\}, \quad (4.20)$$

für alle $j \in \underline{K}(r_p^{k-1}, z^k, u^k)$, falls $v_j^k - u_j^k < 0$.

Anderenfalls setze

$$r_{p+1,j}^{k-1} := \kappa_1 r_{p,j}^{k-1}. \quad (4.21)$$

Die Idee bei diesem Update des Penalty-Parameters ist es, diejenigen Koeffizienten stärker anzupassen, deren Einfluß für das Erreichen der Abstiegsbedingung am größten ist. Durch κ_2 wird das Anwachsen des Koeffizienten nach oben eingeschränkt, mit κ_1 wird erreicht, daß sich jeder Koeffizient erhöht. Das Penalty-Parameter-Update wird für TRSCP und LSSCP analog durchgeführt, um die Algorithmen vergleichbarer zu machen.

Als weitere Modifikation des Algorithmus TRSCP wird eine Möglichkeit vorgestellt, mit Teilproblemen umzugehen, die einen leeren zulässigen Bereich besitzen. Diese Modifikation ist notwendig, um stets einen neuen Vorschlag für einen Entwicklungspunkt finden zu können. Wie schon im Abschnitt 2.3 angedeutet, wird in diesem Fall mit Hilfe von Schlupfvariablen ein Ersatz-Teilproblem mit nichtleerem zulässigen Bereich aufgestellt und gelöst. Mit dieser Lösung läßt sich eine Abstiegsrichtung für die erweiterte Lagrange-Funktion von EOP definieren.

Der Fall, der hier betrachtet wird, kann nur eintreten, wenn im Entwicklungspunkt $z \in \mathbb{R}^n$, für den ein Teilproblem aufgestellt werden soll, mindestens eine Nebenbedingung von EOP verletzt ist. Würde dies nicht gelten, läge zumindest z in der zulässigen Menge von EOP, als auch in der eines jeden in diesem Punkt aufgestellten Teilproblems. Diese Verletzung der Nebenbedingungen ist nach Konstruktion nicht für eine Box-constraint möglich.

Sei EOP und $z \in \mathbb{R}^n$ gegeben. Dann ist

$$M_v := \{j \in \{1, \dots, m_g\} : |\underline{h}_j(z)| \neq 0\} \cup \{j \in \{m_g + 1, \dots, m\} : h_j(z) > 0\} \quad (4.22)$$

eine Teilmenge der Indizes $j \in 1, \dots, \underline{m}$. Diese Teilmenge enthält alle Indizes der Nebenbedingungen von EOP, die verletzt sind. Die Anzahl der Indizes der Menge M_v sei gleich $m_v \in \mathbb{N}_0$.

Mit diesen Festlegungen ergibt sich das folgende alternative Teilproblem für TRSCP im Fall eines original Teilproblems mit leerem zulässigen Bereich:

Definition 4.1.3 (Alternatives konvexes Teilproblem von TRSCP)

Sei EOP, $k \in \mathbb{N}_0$, ein Entwicklungspunkt $z^k \in \mathbb{R}^n$, ein Trust-Region-Radius $\Delta^k \in \mathbb{R}^n$, sowie die Konstanten $\xi > 0$, $\tau > 0$ und $\rho \geq 1$ gegeben. Dann ist das Problem

ASP($\mathbf{z}^k, \Delta^k, \rho$) gegeben durch

$$\begin{aligned}
& \text{Minimiere} && f^k(x, z^k, \Delta^k, \xi, \tau) + \frac{1}{2} \sum_{j \in M} \rho_j \theta_j^2, \\
& \text{unter} && h_j^k(x, z^k, \Delta^k, \xi) - \theta_j h_j(z^k) = 0, \quad j = 1, \dots, m_g, \\
& && h_j^k(x, z^k, \Delta^k, \xi) - \theta_j h_j(z^k) \leq 0, \quad j = m_g + 1, \dots, m, \\
& && g_i^k(x, z^k, \Delta^k, \underline{x}_i) \leq 0, \quad i \in I_u, \\
& && g_{n+i}^k(x, z^k, \Delta^k, \bar{x}_i) \leq 0, \quad i \in I_o, \\
& && g_i^k(x, z^k, \Delta^k) \leq 0, \quad i = 1, \dots, n \wedge i \notin I_u \\
& && g_{n+i}^k(x, z^k, \Delta^k) \leq 0, \quad i = 1, \dots, n \wedge i \notin I_o \\
& && 0 \leq \theta_j \leq 1, \quad j \in M, \\
& && x \in \mathbb{R}^n,
\end{aligned} \tag{4.23}$$

und heißt alternatives, konvexes Teilproblem in z^k des nichtlinearen Optimierungsproblems EOP.

Die Definition der Funktionen $f^k(x, z^k, \Delta^k, \xi, \tau)$, $h_j^k(x, z^k, \Delta^k, \xi)$, $g_i^k(x, z^k, \Delta^k, \underline{x}_i)$ und $g_i^k(x, z^k, \Delta^k)$ wird aus dem Lemma 3.2.2 und (3.38) bzw. (4.5) ersichtlich.

Für $\theta := (1, \dots, 1)$ ist der Entwicklungspunkt z^k Element des zulässigen Bereichs des alternativen Teilproblems. Beginnend mit kleinem ρ kann in einer endlichen Iteration ρ gesteigert und die Schlupfvariable θ reduziert werden. Mit Hilfe des Ergebnisses des alternativen Teilproblems läßt sich eine Abstiegsrichtung definieren und ein neuer Entwicklungspunkt bestimmen, mit dem die Iteration fortgesetzt werden kann. Eine weitergehende Ausführung hierzu findet sich in der Arbeit von Zillober [69]. Der Fall eines leeren zulässigen Bereichs wird im weiteren nicht betrachtet und die Lösung dieses Falles ist im folgenden Algorithmus auch nicht integriert.

Um die Asymptoten $(L, U) \in \mathbb{R}^{2n}$ für ein Teilproblem zu bestimmen wird bei MMA für jeden Koeffizienten der primalen Variablen die Richtung betrachtet, in die sich dieser in den letzten zwei Iterationen bewegt hat (siehe Svanberg [59]). Die bei MMA angewandte Strategie (siehe Algorithmus 2.3) führt dazu, daß der Abstand von z zu den Asymptoten (L, U) klein wird, wenn diese Richtung ständig wechselt. Das kann unter anderem dann passieren, wenn in der Nähe eines stationären Punkts die neu bestimmte Näherung in zu großem Abstand von der aktuellen Näherung bestimmt wird. Wie leicht einzusehen ist, ist in diesem Fall die Reduktion des Abstands günstig. Bei LSSCP wird ebenfalls mit dieser Strategie gearbeitet und da die Information zur Bestimmung der MMA-Strategie zur Asymptotenwahl ohne Aufwand zu

erhalten ist, fließt bei TRSCP diese Information ebenfalls in die Bestimmung des Trust-Region-Radius $\Delta \in \mathbb{R}^n$ ein.

Wie aus Algorithmus 2.3 ersichtlich werden bei der MMA-Strategie zur Wahl der Asymptoten in der k -ten Iteration zwei Fälle unterschieden. Entwickelte sich in der Iteration $k - 1$ der Koeffizient $i = 1, \dots, n$ des Entwicklungspunkts in die gleiche Richtung wie in Iteration $k - 2$, d.h.

$$(z_i^k - z_i^{k-1}) (z_i^{k-1} - z_i^{k-2}) > 0, \quad (4.24)$$

dann wird im Koeffizienten i der Abstand zwischen Asymptote und dem Entwicklungspunkt z^k erhöht. Gilt (4.24) nicht, dann wird der Abstand reduziert.

Basierend auf der Auswertung des Quotienten von erwarteter und erreichter Verbesserung werden bei der Trust-Region-Strategie (TR-Strategie) in jeder Iteration drei Fälle unterschieden (siehe (3.6)). Erstens, der Trust-Region-Radius wird reduziert, weil die prognostizierte Verbesserung wesentlich besser ist als die tatsächlich erreichte. Zweitens, der Trust-Region-Radius wird erhöht, weil die Prognose sehr gut erfüllt oder sogar übertroffen wird. Trifft weder erstens noch zweitens zu, dann bleibt im dritten Fall der Radius konstant.

Folglich ergeben sich für eine Kombination von MMA-Strategie und TR-Strategie sechs unterschiedliche Fälle. Tabelle 4.1 listet diese sechs Fälle auf, wobei + für eine Vergrößerung, – für eine Verkleinerung und 0 für ein Beibehalten des Abstands bzw. des Trust-Region-Radius steht. Beide Strategien werden wie folgt kombiniert:

1. Die MMA-Strategie entscheidet, ob der Trust-Region-Radius vergrößert oder verkleinert wird.
2. Die Trust-Region-Strategie entscheidet im Fall einer Vergrößerung, wie stark der Trust-Region-Radius vergrößert wird.

Die vorgeschlagenen Kombination der beiden Strategien ist in Tabelle 4.1 dargestellt, wobei + + + eine sehr starke Vergrößerung, ++ eine starke Vergrößerung und + eine einfache Vergrößerung bezeichnet. Die Verkleinerung ist gleich für alle drei Fälle und mit – gekennzeichnet.

<i>TR – Strategie</i>	<i>MMA – Strategie</i>	<i>Kombination</i>
+	+	+ + +
0	+	++
–	+	+
+	–	–
0	–	–
–	–	–

Tabelle 4.1: Veränderung der Trust-Region bei Kombination der MMA-Strategie und der TR-Strategie

Die für TRSCP implementierte kombinierte Strategie gilt mangels Information erst ab $k \geq 2$. In den ersten beiden Iterationsschritten wird die normale Trust-Region-Strategie zur Bestimmung des Trust-Region-Radius angewendet, d.h. ein Q^k größer eines bestimmten Schwellwerts führt zu einer Vergrößerung, Q^k unter einem festgelegten Schwellwert zu einer Verkleinerung des Trust-Region-Radius.

In den betrachteten Testproblemen aus Hock und Schittkowski [27] und Schittkowski [49] hat es sich in keiner betrachteten Variation als vorteilhaft erwiesen unterschiedliche Faktoren bei der Reduzierung des Trust-Region-Radius in äußeren Iterationen anzuwenden. Für diese Betrachtung wurden immer alle Testprobleme untersucht und die Bewertung aufgrund des Gesamtergebnisses getroffen.

Mit diesen Änderungen bzw. Erweiterungen des Basisalgorithmus 3.2 aus Abschnitt 3.2 ergibt sich der folgende Algorithmus, der im Programm TRSCP1.0 implementiert ist:

Algorithmus 4.1 (TRSCP)

Sei EOP, $x^{\text{Eingabe}} \in \mathbb{R}^n$ und $\varepsilon > 0$ gegeben.

Schritt 0: Initialisierung:

Setze $z^0 \in \mathbb{R}^n$ nach (4.9), $u^0 \in \mathbb{R}^m$ mit $u_j^0 := 0$ für $j = 1, \dots, m$, einen Penalty-Parameter $r_0^{-1} \in \mathbb{R}^m$ mit $r_{0,j}^{-1} := 1$ für $j = 1, \dots, m$ und Konstanten $\tau := 0.001$, $\kappa_1 := 2$, $\kappa_2 := 10$, $q := 5$. Berechne $f(z^0)$, $\nabla f(z^0)$ und $\underline{h}_j(z^0)$, $\nabla \underline{h}_j(z^0)$ für $j = 1, \dots, m$. Ferner setze Δ^{\max} nach (4.10), Δ_{\max} nach (4.11), Δ_0^0 nach (4.12) mit $T_1 := 5$, $T_2 := 1$, $\lambda := 0.25$ und es sei

$$\begin{aligned} \xi &:= 0.2, \\ \mu_1 &:= 0.1, \\ \mu_2 &:= 0.9, \\ \nu_0 &:= 0.\bar{3}, \\ \nu_1 &:= 0.7, \\ \nu_2 &:= 1.15, \\ \nu_3 &:= 2.0, \\ \nu_4 &:= 4.0, \\ \gamma &:= 0.001, \end{aligned}$$

Initialisiere $k := 0$, $q := 0$.

Schritt 1: Aufstellen und Lösen des Unterproblems:

Bestimme das Optimum (y_q^k, v_q^k, w_q^k) des konvexen Teilproblems $TRSP(z^k, \Delta_q^k)$.

Ist Definition 4.1.1 für y_q^k erfüllt, dann ist $\begin{pmatrix} x^* \\ u^* \end{pmatrix} := \begin{pmatrix} y_q^k \\ v_q^k \end{pmatrix}$ stationärer Punkt von EOP, STOP.

Ansonsten ist $s_q^k := \begin{pmatrix} y_q^k - z^k \\ v_q^k - u^k \end{pmatrix}$ und $\delta_q^k := \|y_q^k - z^k\|_2$. Initialisiere $p := 0$.

Schritt 2: Anpassen des Penalty-Parameters:

Falls

$$\nabla \Phi_{r_p^{k-1}} \begin{pmatrix} z^k \\ u^k \end{pmatrix}^T s_q^k > -\frac{\tau(\delta_q^k)^2}{4(\Delta_{\max} + \xi)}, \quad (4.25)$$

erzeuge r_{p+1}^{k-1} (Definition 4.1.2), setze $p := p + 1$ und wiederhole *SCHRITT 2*.
Anderenfalls setze $r_0^k := r_p^{k-1}$ und berechne $f(y_q^k)$ und $\underline{h}_j(y_q^k)$, $j = 1, \dots, \underline{m}$.

Schritt 3: Überprüfung des Abstiegs:Falls $q \leq \underline{q}$ und

$$\Phi_{r_0^k} \begin{pmatrix} y_q^k \\ v_q^k \end{pmatrix} - \Phi_{r_0^k} \begin{pmatrix} z^k \\ u^k \end{pmatrix} > \gamma \nabla \Phi_{r_0^k} \begin{pmatrix} z^k \\ u^k \end{pmatrix}^T s_q^k, \quad (4.26)$$

verkleinere den Trust-Region-Radius durch

$$\Delta_{q+1}^k := \nu_0 \Delta_q^k, \quad (4.27)$$

setze $q := q + 1$ und gehe zu *SCHRITT 1*.**Schritt 4: Überprüfung des Trust-Region-Radius:**Setze $z^{k+1} := y_q^k$, $u^{k+1} := v_q^k$, $s^k := s_q^k$ und bestimme den Quotienten

$$Q^k := \frac{\Phi_{r_0^k} \begin{pmatrix} z^{k+1} \\ u^{k+1} \end{pmatrix} - \Phi_{r_0^k} \begin{pmatrix} z^k \\ u^k \end{pmatrix}}{\nabla \Phi_{r_0^k} \begin{pmatrix} z^k \\ u^k \end{pmatrix}^T s^k}. \quad (4.28)$$

Falls $q > \underline{q}$, setze $\Delta_0^{k+1} := \Delta_0^k$,ansonsten, für $i = 1, \dots, n$, falls $(z_i^k - z_i^{k-1})(z_i^{k-1} - z_i^{k-2}) > 0$ setze

$$\Delta_{0,i}^{k+1} := \begin{cases} \min\{\nu_4 \Delta_{0,i}^k; \Delta_{\max}\}, & \text{für } Q_k > \mu_2, \\ \min\{\nu_3 \Delta_{0,i}^k; \Delta_{\max}\}, & \text{für } \mu_1 \leq Q_k \leq \mu_2, \\ \min\{\nu_2 \Delta_{0,i}^k; \Delta_{\max}\}, & \text{für } Q_k < \mu_1, \end{cases} \quad (4.29)$$

anderenfalls setze

$$\Delta_{0,i}^{k+1} := \nu_1 \Delta_{0,i}^k. \quad (4.30)$$

Setze $q := 0$, berechne $\nabla f(z^{k+1})$ und $\nabla \underline{h}_j(z^{k+1})$ für $j = 1, \dots, \underline{m}$, setze $k := k + 1$ und gehe zu *SCHRITT 1*.

Wesentliche Unterschiede zum Basisalgorithmus sind,

1. die Möglichkeit den Trust-Region-Radius über den Wert des Startradius hinaus zu vergrößern und
2. individuelle Trust-Region-Radien für jeden Koeffizienten der primalen Variablen zu erzeugen.

Die für den im Programm TRSCP1.0 implementierten Algorithmus TRSCP gesetzten Parameter orientieren sich an im Allgemeinen benutzte Werte für entsprechende Parameter im Algorithmus 2.4 (LSSCP) bzw. dem Algorithmus 3.1 von Yuan. Allerdings wurden Parameter geändert, wenn sich dies als vorteilhaft für TRSCP bei verschiedenen Testläufen herausgestellt hat.

4.2 Vergleich von TRSCP mit LSSCP

Dieser Abschnitt enthält Ergebnisse des Programms TRSCP1.0 für die Lösung von Testproblemen. Die Testprobleme sind aus den Sammlungen Hock und Schittkowski [27] und Schittkowski [49]. Mit TRSCP1.0 werden sowohl die Ergebnisse für den Algorithmus TRSCP, als auch die Ergebnisse für den Algorithmus LSSCP erzeugt. TRSCP1.0 ist eine Weiterentwicklung des Programms SCIP3.0 von Zillober [76, 77].

Für den Vergleich der Algorithmen TRSCP und LSSCP werden nur die Testprobleme aus Hock und Schittkowski [27] bzw. Schittkowski [49] aufgeführt, bei denen mindestes einer der Algorithmen eine Lösung liefert. Es werden bei diesem Vergleich drei Kategorien von Ergebnissen unterschieden. Erstens, ein Algorithmus beendet die Iteration mit einem Lösungsvorschlag für das Testproblem und der Anzahl von durchgeführten Funktions- bzw. Gradientenauswertungen. Zweitens, die Iteration endet mit einem Fehler und ohne einen Lösungsvorschlag. Drittens die Iteration endet nach einer gewissen Anzahl von Funktionsauswertungen, ebenfalls ohne einen Lösungsvorschlag.

Im ersten Fall, das heißt bei der erfolgreichen Beendigung der Iteration, wird nicht nur die Anzahl der benötigten Funktions- und Gradientenauswertungen aufgelistet. Es wird weiter unterschieden, ob die Algorithmen eine Lösung präsentieren, die bei den Testproblemen als optimale Lösung angeführt ist oder nicht. In den meisten Fällen stimmt die generierte Lösung von TRSCP und LSSCP überein, wenn beide Algorithmen erfolgreich terminieren.

Bricht die Iteration mit einer Fehlermeldung ab, wird bei diesem Vergleich nicht unterschieden, welche Art von Fehler den Abbruch hervorgerufen hat. Die häufigsten Arten von Fehlermeldung sind Abbruch, weil kein zulässiger Vektor gefunden werden kann, Abbruch, weil keine Lösung des Teilproblems gefunden werden kann

und Abbruch, weil trotz gefundener Abstiegsrichtung mit der Line-Search bei LS-SCP kein Iterationspunkt mit einem kleineren Wert in der Merit-Funktion gefunden werden kann.

Beim Abbruch der Iteration aufgrund einer zu großen Iterationsanzahl muß das Abbruchkriterium in TRSCP1.0 im Vergleich zu SCIP3.0 geändert werden. Um für beide Algorithmen ein ähnliches Abbruchkriterium zu erhalten, wird eine Iteration zu dem Zeitpunkt abgebrochen, wenn am Ende einer Iteration mehr als $F_{\max} \in \mathbb{IN}$ Funktionsauswertungen durchgeführt wurden. Das Abbruchkriterium in SCIP3.0 beendet eine Iteration nach einer festen Anzahl von Iterationen. Bedingt durch die Programmstruktur von TRSCP1.0 haben TRSCP und LSSCP eine unterschiedliche maximale Anzahl von Funktionsauswertungen in einer Iteration, d.h. das ursprüngliche Abbruchkriterium von SCIP3.0 ist zu unterschiedlich für diese beiden Algorithmen. Die Untersuchung wurde mit

$$F_{\max} := 5000 \tag{4.31}$$

durchgeführt.

Die für diese Untersuchung herangezogenen Testprobleme stellen für Optimierungsprogramme besonders schwierige Aufgaben dar. SCP-Methoden sollten zur Lösung derartiger Aufgaben nicht herangezogen werden, SQP-Verfahren sind hierfür besser geeignet. Aber gerade aus diesem Grund wird der Vergleich von TRSCP und LS-SCP mit diesen Testproblemen durchgeführt, um die SCP-Methoden besonders zu fordern. Als Implementierung der Testprobleme wird die Version von Schittkowski [52] Stand Februar 2011 verwendet. Zum Lösen von Problemen der Art wie die der hier betrachteten eignet sich z.B. die Fortran-Routine NLPQL.f (Schittkowski [48]). SCP-Verfahren haben im Vergleich zu SQP-Verfahren ihre Stärken bei Strukturoptimierungsproblemen (siehe Schittkowski et al. [50]).

Für die Parameter bei LSSCP wird die Standardsetzung von SCIP3.0 verwendet. Bei TRSCP wird die Untersuchung mit den im Algorithmus 4.1 angegebenen Parameterwerten durchgeführt. Die Genauigkeitsanforderung wird für diese Testprobleme mit $\varepsilon := 10^{-7}$ festgelegt. Die folgenden Tabellen enthalten die Anzahlen der benötigten Funktionsauswertungen (f), die Anzahlen der benötigten Gradientenauswertungen (df) und die Kennzeichnung (X), falls ein Algorithmus eine andere Lösung als bei Schittkowski [52] angegeben liefert. Die Testprobleme aus Hock und Schittkowski [27] sind TP1 bis TP119, die Testprobleme aus Schittkowski [49] sind TP201 bis TP395.

Für die insgesamt 306 betrachteten Testproblemen liefert TRSCP 186 und LSSCP 183 Optimallösungen. Davon entfallen bei TRSCP 75 auf Aufgaben aus Hock und Schittkowski [27] und 111 auf Aufgaben aus Schittkowski [49]. Bei LSSCP sind das entsprechend 71 und 112. Von den Testproblemen werden 103 weder von TRSCP noch von LSSCP gelöst. Bei 147 Aufgaben identifiziert TRSCP eine bei den Testproblemen angegebene Lösung, 155 sind es bei LSSCP.

Bei diesem durchgeführten Test erweist sich das im Programm TRSCP1.0 implementierte SCP-Verfahren mit Trust-Region-Stabilisierung als genauso leistungsstark, wie

Testproblem	TRSCP			LSSCP		
		f	df		f	df
TP2		167	101	X	358	313
TP3		23	11		22	21
TP4		3	3		3	3
TP5		58	36		144	32
TP6		42	24		98	54
TP7		11	11		51	34
TP8		6	6		6	6
TP9		14	14		16	16
TP10		26	23		17	17
TP12		18	17		18	18
TP16	X	7	7	X	7	7
TP17		17	15		116	61
TP18		49	49		21	20
TP20		7	7		8	8
TP21		6	3		103	34
TP22		9	7		8	8
TP23		12	11		14	13
TP24		8	8		10	10
TP26		48	28		max	
TP27		max			1865	1060
TP28		120	74		150	44
TP29		18	17		17	17
TP30		16	16		77	52
TP31		226	153		105	49
TP33		10	10		8	8
TP34		8	8		8	8
TP35		24	18		22	19
TP36		11	11		8	8
TP37		543	322		22	22
TP39		60	42		223	79
TP40		12	11		11	11
TP41		13	13		15	15
TP42		182	57		214	75
TP43		26	23		20	20
TP44		10	10		12	12
TP45		5	5		4	4
TP46		153	104		max	
TP47		60	39		115	52
TP48		77	51		145	81

Tabelle 4.2: Ergebnisse bei Testproblemen aus Hock und Schittkowski [27] (Teil 1)

Testproblem	TRSCP		LSSCP			
		f	df		f	df
TP50		Fehler			122	64
TP51		53	37		98	67
TP53		26	21		Fehler	
TP54	X	3	3	X	3	3
TP55	X	9	9	X	9	9
TP57	X	45	25		max	
TP58		19	17		20	17
TP59		22	20		27	22
TP60		71	44		46	30
TP62		1092	674		66	62
TP63		95	68		27	26
TP65		23	19		31	29
TP66		25	25		18	18
TP68		1609	1330		max	
TP70	X	1574	949		915	710
TP73	X	1026	209		Fehler	
TP76		33	23		18	18
TP77		80	53		55	37
TP79		38	23		46	32
TP80		15	12		11	11
TP84		15	15		11	11
TP86		33	27		21	18
TP88		1070	693		77	45
TP90	X	282	163	X	168	79
TP91	X	449	271	X	242	140
TP92	X	4527	4338	X	194	91
TP93		899	670		50	49
TP95		6	6		8	8
TP96		5	5		8	8
TP97		11	11		17	17
TP98		12	12		17	17
TP100		109	62		23	21
TP104		max			95	62
TP105	X	2	1	X	5	2
TP108	X	25	24	X	37	27
TP110		21	9		25	8
TP113		186	110		32	28
TP116		22	20		Fehler	
TP117		219	219		79	75

Tabelle 4.3: Ergebnisse bei Testproblemen aus Hock und Schittkowski [27] (Teil 2)

Testproblem	TRSCP			LSSCP		
		f	df		f	df
TP201		40	23		255	57
TP202	X	605	454	X	245	55
TP203		370	220		226	76
TP204		99	70		164	31
TP205		957	572		231	62
TP206		697	573		81	24
TP207		124	88		220	57
TP212		4033	3997		130	32
TP215		26	13		324	317
TP216	X	3169	2844	X	510	507
TP218		8	8		68	54
TP219		66	43		110	59
TP220		81	81		80	80
TP221		14	14		15	15
TP222		7	7		6	6
TP223		8	8		7	7
TP224		192	150		33	33
TP225		12	11		14	13
TP226		14	12		11	11
TP227		6	5		7	7
TP228		29	16		50	29
TP230		max			9	8
TP232		9	9		9	9
TP233		max			4524	3186
TP234		34	34		20	20
TP235		2131	1452		377	249
TP236	X	27	22		11	10
TP237		18	18		18	16
TP238	X	25	21	X	22	20
TP239	X	27	22		12	10
TP240		max			289	76
TP242	X	1148	900		1356	1158
TP243		76	54		247	67
TP244		835	571		206	106
TP245		Fehler			569	381
TP247		71	37		30	6
TP249		27	21		105	65

Tabelle 4.4: Ergebnisse bei Testproblemen aus Schittkowski [49] (Teil 1)

Testproblem	TRSCP		LSSCP			
	f	df	f	df		
TP250	11	11	8	8		
TP251	474	272	22	22		
TP252	max		1667	1605		
TP253	52	29	1566	1240		
TP254	13	13	14	14		
TP257	1156	885	956	639		
TP258	1490	1097	397	165		
TP259	2273	1871	588	412		
TP260		1497	1101	491	254	
TP261		722	556	237	58	
TP264		162	89	27	24	
TP265		75	28	174	85	
TP269		18	16	Fehler		
TP270	X	2	2	X	306	160
TP271		2	2		295	67
TP272		71	25		20	6
TP274		2	2		153	39
TP275		4043	3022		max	
TP277		27	23		42	42
TP278		309	309		60	60
TP279		654	426		79	79
TP280		160	160		80	80
TP284		27	20		29	29
TP285		3	3		Fehler	
TP289		26	14		86	26
TP290		2	2		7	4
TP291		2	2		7	4
TP292		2	2		7	4
TP293		2	2		Fehler	
TP294		794	593		max	
TP295		1516	1124		max	
TP296		51	30		max	
TP297		1732	1295		max	
TP298		2344	1758		max	
TP299		1769	1308		max	
TP303		1685	1207		10	4
TP304		max			10	4
TP305		Fehler			10	4
TP307		max			95	88
TP309		790	289	X	153	36

Tabelle 4.5: Ergebnisse bei Testproblemen aus Schittkowski [49] (Teil 2)

Testproblem	TRSCP			LSSCP		
		f	df		f	df
TP310	X	2	2		Fehler	
TP311		57	39		307	67
TP314		275	190		246	54
TP315		11	11		12	12
TP324		49	49		21	20
TP326		7	6		9	9
TP327	X	45	25		max	
TP328		57	36		138	30
TP330		19	16		24	22
TP331		45	37		118	64
TP332	X	2	2	X	2	2
TP333		max			277	229
TP335		max			109	54
TP337		145	74		99	46
TP338	X	48	38		Fehler	
TP339		62	49		26	24
TP341		20	19		17	17
TP342		18	17		17	17
TP344		42	23		72	47
TP345		59	38		40	27
TP347		6	6		5	5
TP349	X	2	1	X	2	2
TP350		max			2254	1661
TP354		169	114		64	44
TP355		Fehler			72	44
TP357	X	2	1	X	7	2
TP358	X	1364	891		Fehler	
TP359		52	52		26	26
TP360		15	15		11	11
TP362	X	2	2	X	7	2
TP363	X	2	2	X	2	2
TP364	X	2	2	X	2	2
TP366	X	2	2	X	6	2
TP368	X	59	35	X	133	79
TP370	X	2	1	X	2	2
TP371	X	2	1	X	2	2
TP377	X	3	3	X	3	3
TP378	X	126	35		Fehler	
TP379		max			2022	1542

Tabelle 4.6: Ergebnisse bei Testproblemen aus Schittkowski [49] (Teil 3)

Testproblem	TRSCP			LSSCP		
		f	df		f	df
TP380	X	7	7	X	7	7
TP381	X	3723	755		Fehler	
TP382	X	3855	775		Fehler	
TP384		91	45		21	21
TP385		79	46		28	28
TP386		Fehler			34	34
TP387		max			28	28
TP388		max			47	46
TP389		max			42	40
TP391	X	2	2	X	2	2
TP393	X	2	1	X	2	2
TP394		max			238	94
TP395		535	450		205	91

Tabelle 4.7: Ergebnisse bei Testproblemen aus Schittkowski [49] (Teil 4)

das SCP-Verfahren mit Line-Search-Stabilisierung. TRSCP hat leichte Vorteile bei der Anzahl gelöster Testprobleme, LSSCP bei der Anzahl exakt, d.h. entsprechend der angegebenen Lösung, gelöster Probleme.

Das setzen der Parameter hat bei TRSCP einen großen Einfluß auf das Ergebnis. Werden einige Parameter bei TRSCP, wie z.B. μ_i , $i = 1, 2$ oder ν_j , $j = 0, \dots, 4$ wesentlich anders gesetzt, dann hat das entscheidenden Einfluß auf die Leistung von TRSCP beim Lösen der betrachteten Testprobleme.

4.3 Weitere Resultate von TRSCP

Zum Abschluß dieser Arbeit wird das Ergebnis von TRSCP1.0 bei Testproblemen aus dem Bereich der Strukturoptimierung präsentiert. Entsprechend Abschnitt 4.2 werden einige Aufgaben mit TRSCP und LSSCP bearbeitet und die jeweils benötigte Anzahl von Funktionsauswertungen bzw. Gradientenauswertungen verglichen. Die Genauigkeitsanforderung wird für diese Testprobleme mit $\varepsilon := 10^{-5}$ festgelegt. Auf den Einsatz eines Topologie-Filters wird im weiteren verzichtet, um die beiden Verfahren ohne störende Einflüsse vergleichen zu können, d.h. die Topologieoptimierung wird ohne einen Topologie-Filter, der die Gradienten der Funktionen des Testproblems verändert durchgeführt.

Für die Aufgabenbeschreibungen und Lösungsvorschläge wird auf Sigmund [53] verwiesen. Die Aufgaben in diesem Abschnitt werden in der Feinheit der Diskretisierung als auch im angestrebten Materialanteil variiert. Graphisch werden in diesem Abschnitt nur die Endergebnisse der SCP-Verfahren gezeigt und dazu die Anzahl der durchgeführten Funktions- und Gradientenauswertungen notiert.

Es wird zu jeder Lösung angegeben, mit welchem Faktor die Diskretisierung bzw. der angestrebte Materialanteil verändert wurde. 'x1' bezeichnet die original Diskretisierung aus Sigmund [53]. 'x2' bezeichnet beispielsweise eine Verfeinerung der Diskretisierung auf doppelt so viele Elemente in jeder Dimension. Das heißt bei 'x2' kommen bei diesen Beispielen viermal so viele Elemente zum Einsatz. '40%' bedeutet, daß das Bauteil auf vierzig Prozent des möglichen Materials gebracht werden soll.

TRSCP beendet jede Iteration für die Testproblemen in diesem Abschnitt erfolgreich. Bei LSSCP hingegen gibt es zwei mögliche Ausgänge für die Iteration. Erstens, LSSCP beendet die Iteration erfolgreich und liefert neben einer Anzahl durchgeführter Funktions- und Gradientenauswertungen die optimierte Topologie des Bauteils. Zweitens, LSSCP beendet die Iteration mit der Fehlermeldung, daß keine Lösung für ein Teilproblem gefunden werden kann. In diesem Fall wird neben der bis dahin benötigten Anzahl von Funktions- bzw. Gradientenauswertungen der letzte Stand der Struktur ausgegeben. Ein Ergebnis, das bei einem Programmabbruch erzeugt wurde ist in den folgenden Tabellen in runde Klammern gesetzt.

Zum Lösen der Topologieoptimierungsaufgaben kommt eine Implementierung von TRSCP in das Programm SCIP4.0 zum Einsatz. SCIP4.0 ist eine neue Version von SCIP3.0 [76, 77] speziell für die Topologieoptimierung. Durch die neue Version wird die Leistung von LSSCP verbessert. Im Gegensatz zu LSSCP hat TRSCP mit dem Programmstand von SCIP3.0 alle Topologieoptimierungsaufgaben gelöst. Zum Vergleich der Versionen wird bei TRSCP die von Version SCIP3.0 benötigte Anzahl von Funktions- und Gradientenauswertungen in eckigen Klammern mit angegeben. Die abgebildeten von TRSCP erzeugten Strukturen sind mit der Programmstand SCIP4.0 berechnet.

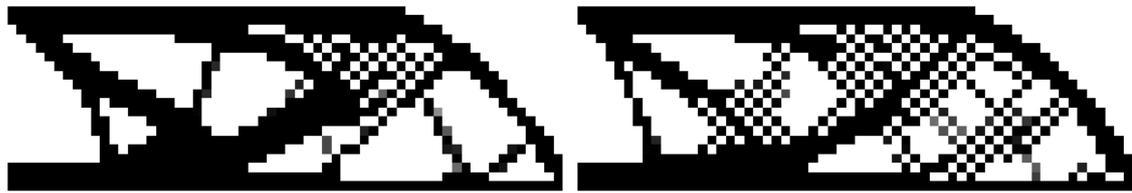
MBB-Halfbeam

Der MBB-Beam ist ein Bauteil mit sechsfacher Länge wie Höhe, das an seinen beiden Enden unten fixiert ist und in der Mitte unten belastet wird. Hier wird das äquivalente Problem² betrachtet, bei dem das Bauteil halbiert wird und die Fixierungen und der Lastfall entsprechend angepaßt werden.

	TRSCP		LSSCP	
	f	df	f	df
x1/50%	84 [62]	84 [57]	98	95
x2/50%	64 [71]	64 [66]	(215)	(146)
x1/40%	84 [119]	84 [119]	87	81
x2/40%	55 [74]	55 [70]	73	71
x3/40%	54 [69]	54 [69]	(110)	(80)
x3/30%	75 [78]	75 [74]	(92)	(67)

Tabelle 4.8: Ergebnisse beim MBB-Halfbeam

²der sogenannte 'MMB-Halfbeam'



(a) TRSCP

(b) LSSCP

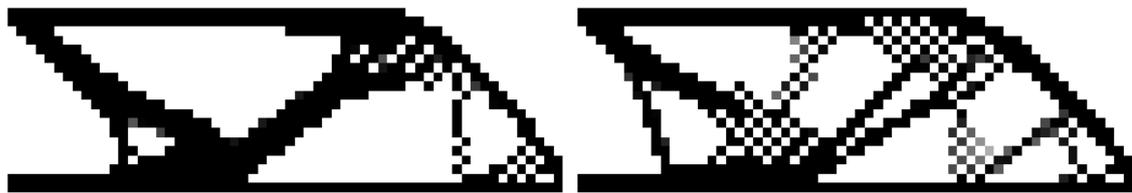
Abbildung 4.1: MBB-Halfbeam (x1/50%)



(a) TRSCP

(b) (LSSCP)

Abbildung 4.2: MBB-Halfbeam (x2/50%)



(a) TRSCP

(b) LSSCP

Abbildung 4.3: MBB-Halfbeam (x1/40%)



(a) TRSCP

(b) LSSCP

Abbildung 4.4: MBB-Halfbeam (x2/40%)



Abbildung 4.5: MBB-Halfbeam (x3/40%)



Abbildung 4.6: MBB-Halfbeam (x3/30%)

TRSCP beendet alle Iterationen erfolgreich. LSSCP benötigt zwar in einem Fall weniger Gradientenauswertungen, bricht aber die Iteration in der Hälfte der betrachteten Fälle vorzeitig ab. In den Fällen mit Iterationsabbruch hat LSSCP zum Teil deutlich öfters die Funktionen und Gradienten ausgewertet, als TRSCP um zum Endergebnis zu kommen. Die erzeugten Strukturen weisen bei TRSCP weniger 'checkerboard-structures' auf als bei LSSCP.

Es werden weitere Lösungen von TRSCP1.0 für das MBB-Beam-Problem aus Sigmund [53] vorgestellt. Es wird eine noch etwas höhere Diskretisierung verwendet und der Materialanteil wird zwischen 90% und 20% variiert.

	TRSCP		LSSCP	
	f	df	f	df
90%	86 [59]	86 [59]	62	59
80%	66 [85]	66 [79]	66	63
70%	89 [98]	89 [98]	72	56
60%	62 [71]	62 [67]	(251)	(143)
50%	60 [111]	60 [102]	101	79
40%	79 [69]	79 [65]	64	57
30%	3599 [78]	3599 [74]	74	65
20%	41 [594]	41 [594]	84	65

Tabelle 4.9: Ergebnisse für MBB-Halfbeam (x3.5) bei verschiedenen Materialanteilen

Wiederum zeigt sich, daß TRSCP stabil Ergebnisse liefert und im Vergleich zu LSSCP weniger 'checkerboard-structures' erzeugt. LSSCP liefert zwar die Ergebnisse

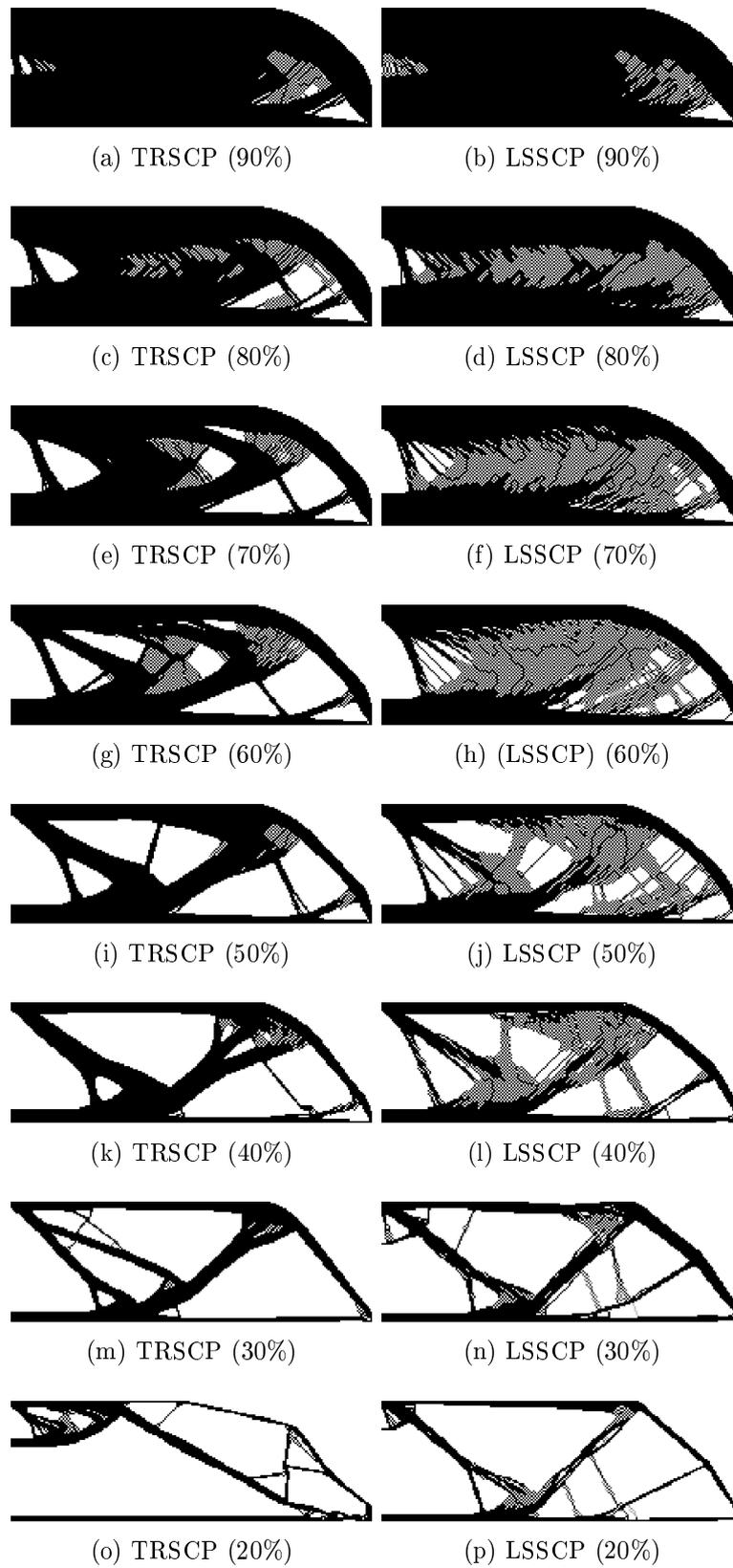


Abbildung 4.7: MBB-Halfbeam (x3.5/90%-20%)

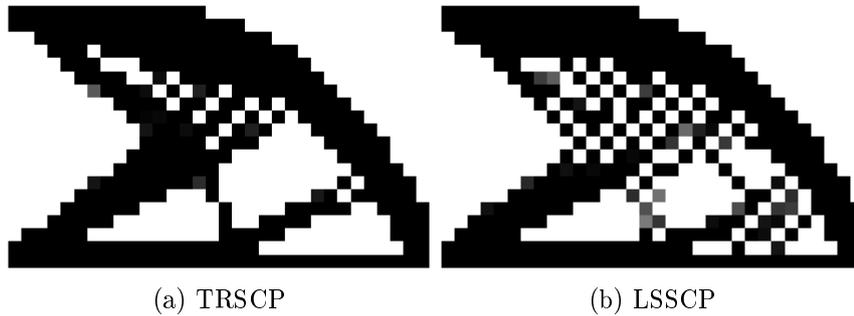


Abbildung 4.8: Cantilever-Beam (x1/50%)

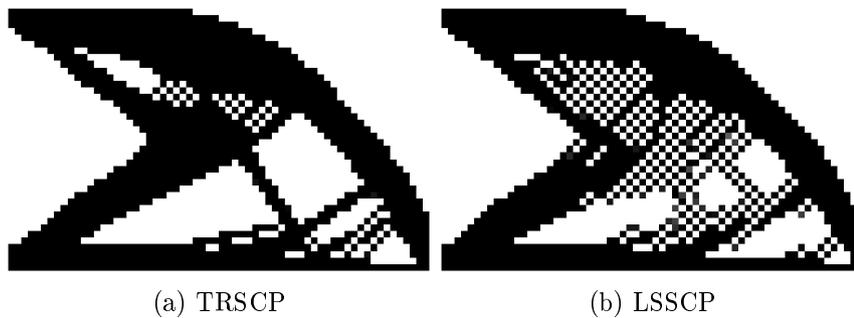


Abbildung 4.9: Cantilever-Beam (x2/50%)

fast immer mit weniger Rechenaufwandt als TRSCP, hat aber dafür zum Teil großflächig Strukturen mit Schachbrettmuster.

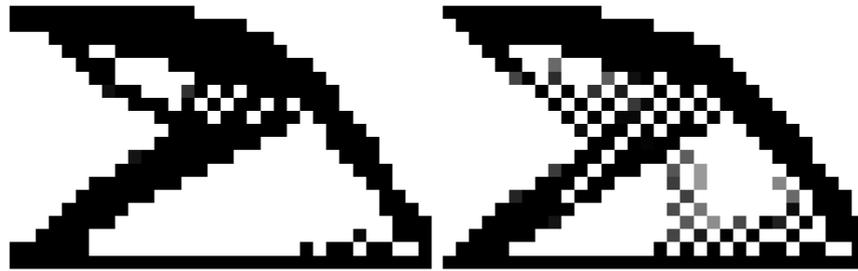
Cantilever-Beam

Der Cantilever-Beam ist ein Bauteil mit der 1.6-fachen Länge wie Höhe. Es ist an der linken Seite fixiert und wird an der rechten Seite unten belastet.

	TRSCP		LSSCP	
	f	df	f	df
x1/50%	56 [56]	56 [56]	76	71
x2/50%	46 [64]	46 [60]	82	65
x1/40%	44 [46]	44 [45]	77	73
x2/40%	32 [52]	32 [52]	112	88
x3/40%	38 [57]	38 [53]	70	62
x3/30%	39 [69]	39 [64]	(183)	(149)

Tabelle 4.10: Ergebnisse beim Cantilever-Beam

Bei diesem Testproblem benötigt TRSCP nicht nur durchweg weniger Funktions- und Gradientenauswertungen als LSSCP, die erzeugten Ergebnisse weisen darüber hinaus weniger Struktur mit Schachbrettmuster auf. LSSCP bricht in einem weiteren Fall die Iteration vorzeitig ab.



(a) TRSCP

(b) LSSCP

Abbildung 4.10: Cantilever-Beam (x1/40%)



(a) TRSCP

(b) LSSCP

Abbildung 4.11: Cantilever-Beam (x2/40%)



(a) TRSCP

(b) LSSCP

Abbildung 4.12: Cantilever-Beam (x3/40%)



(a) TRSCP

(b) (LSSCP)

Abbildung 4.13: Cantilever-Beam (x3/30%)

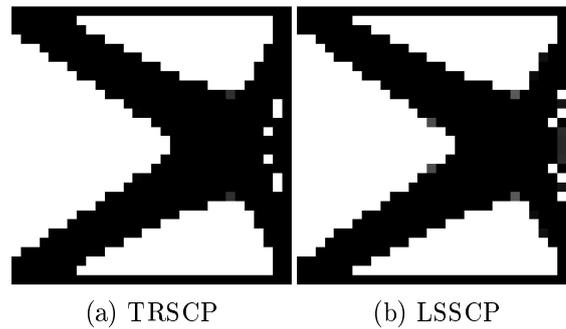


Abbildung 4.14: Cantilever-Beam mit 2 Lasten (x1/50%)

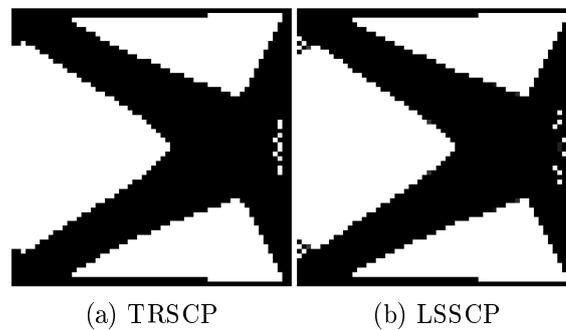


Abbildung 4.15: Cantilever-Beam mit 2 Lasten (x2/50%)

Cantilever-Beam mit zwei Lasten

Der Cantilever-Beam, der durch zwei Lasten beeinflusst wird, ist ein quadratisches Bauteil, dessen linke Seite fixiert ist, und bei dem an der rechten Seite jeweils oben und unten eine Last wirkt. Die Lasten sind gleich stark, wirken aber in die entgegengesetzte Richtung.

	TRSCP		LSSCP	
	f	df	f	df
x1/50%	64 [64]	64 [64]	45	45
x2/50%	46 [46]	46 [46]	45	44
x1/40%	46 [46]	46 [46]	36	36
x2/40%	48 [48]	48 [48]	51	51
x3/40%	135 [132]	135 [132]	42	41
x3/30%	51 [46]	51 [46]	39	39

Tabelle 4.11: Ergebnisse beim Cantilever-Beam mit zwei Lasten

Bei diesem Testproblem ist im Vergleich des Rechenaufwands LSSCP im Vorteil gegenüber TRSCP. In nur einem Fall benötigt TRSCP weniger Funktions- und Gradientenauswertungen als LSSCP, dafür in einem anderen Fall deutlich mehr. In drei Fällen zeigt die von TRSCP erzeugte Struktur weniger 'checkerboard-structures' auf, in den anderen drei Fällen ist die Struktur ähnlich.

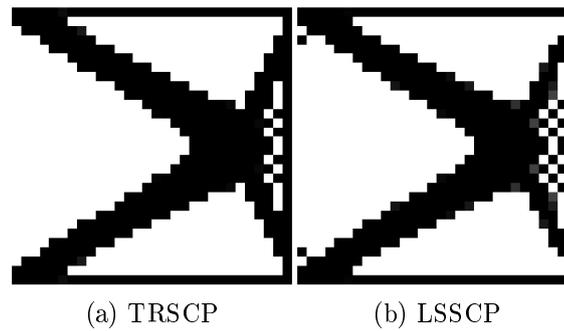


Abbildung 4.16: Cantilever-Beam mit 2 Lasten (x1/40%)

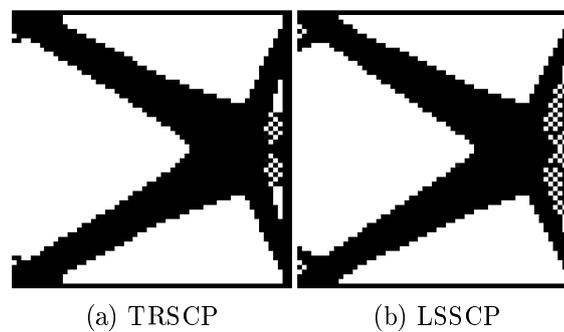


Abbildung 4.17: Cantilever-Beam mit 2 Lasten (x2/40%)

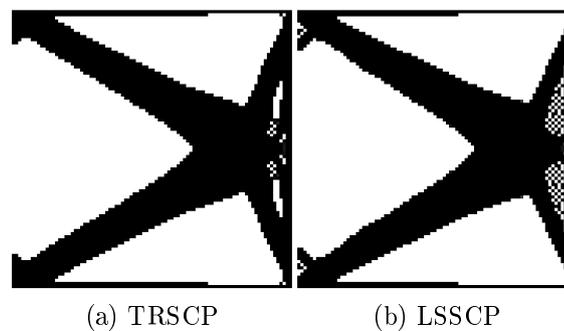


Abbildung 4.18: Cantilever-Beam mit 2 Lasten (x3/40%)

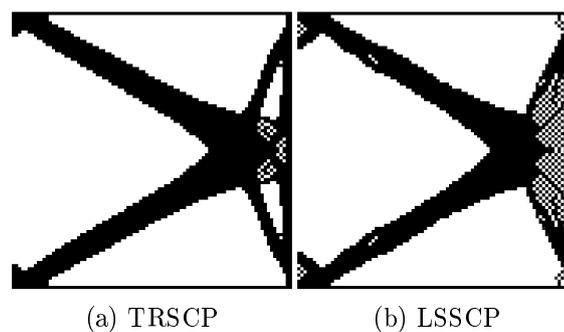


Abbildung 4.19: Cantilever-Beam mit 2 Lasten (x3/30%)

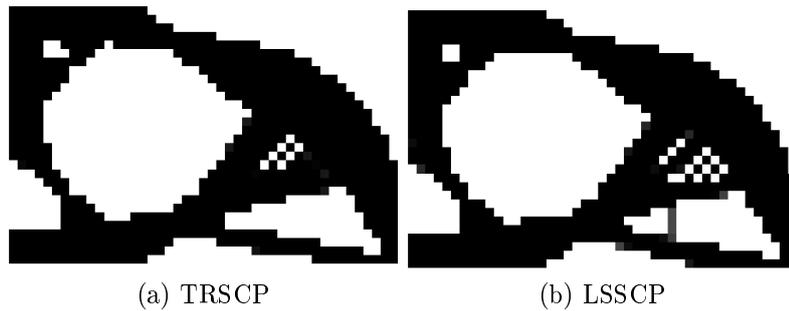


Abbildung 4.20: Cantilever-Beam mit fixem Loch (x1/50%)

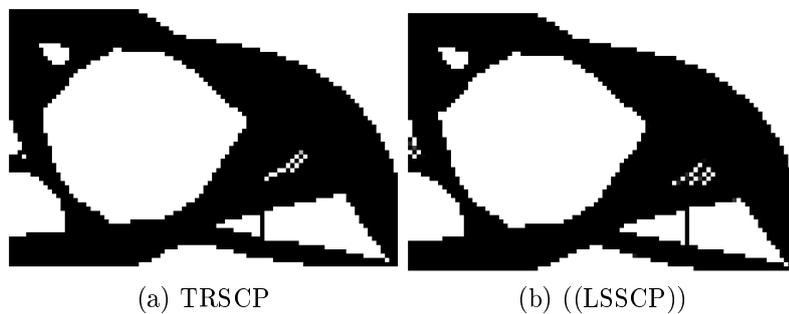


Abbildung 4.21: Cantilever-Beam mit fixem Loch (x2/50%)

Cantilever-Beam mit fixem Loch

Der Cantilever-Beam, der durch ein fixes Loch im 'Rohling' gekennzeichnet ist, ist ein Bauteil, mit anderhalbfacher Länge wie Höhe. Die linke Seite des Bauteils ist fixiert und an der rechten Seite unten wirkt eine Last.

	TRSCP		LSSCP	
	f	df	f	df
x1/50%	53 [63]	53 [59]	55	49
x2/50%	42 [41]	42 [41]	(470)	(150)
x1/40%	41 [41]	41 [41]	75	64
x2/40%	45 [45]	45 [45]	40	40
x3/40%	35 [106]	35 [106]	72	61
x3/30%	55 [64]	55 [60]	75	42

Tabelle 4.12: Ergebnisse beim Cantilever-Beam mit fixem Loch

In einem Fall liefert LSSCP mit weniger Funktions- und Gradientenauswertungen das Ergebnis, in einem anderen Fall benötigt LSSCP zwar mehr Funktions-, aber dafür weniger Gradientenauswertungen. In den anderen Fällen benötigt TRSCP weniger Funktions- und Gradientenauswertungen. In einem Fall bricht LSSCP die Iteration vorzeitig ab. Die erzeugten Strukturen sind sehr ähnlich, obwohl bei TRSCP wieder weniger Struktur mit Schachbrettmuster zu erkennen ist.

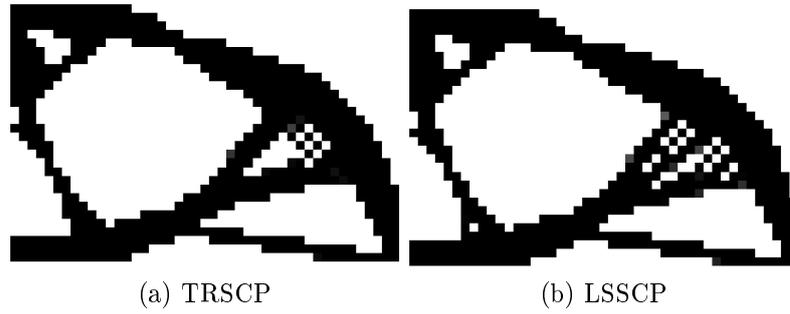


Abbildung 4.22: Cantilever-Beam mit fixem Loch (x1/40%)

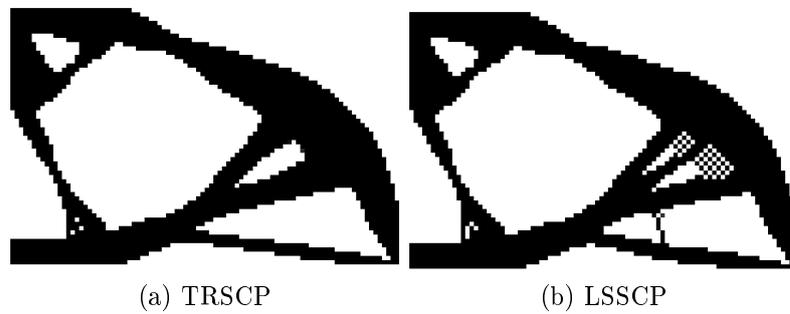


Abbildung 4.23: Cantilever-Beam mit fixem Loch (x2/40%)

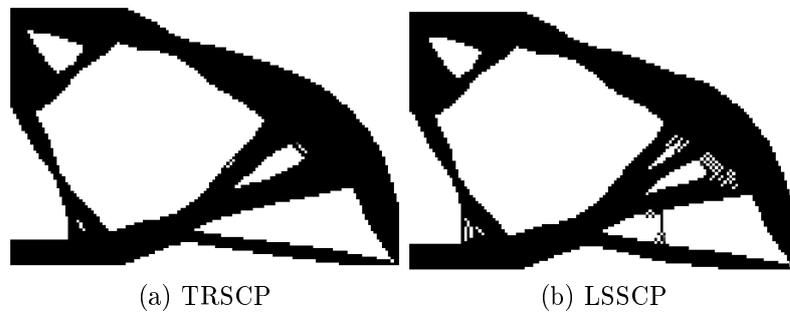


Abbildung 4.24: Cantilever-Beam mit fixem Loch (x3/40%)

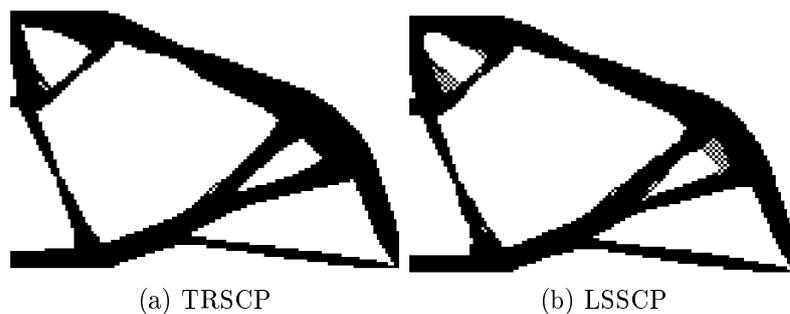


Abbildung 4.25: Cantilever-Beam mit fixem Loch (x3/30%)

Bewertung der Ergebnisse

Der TRSCP-Algorithmus 4.1 in der Implementierung TRSCP1.0 erweist sich bei den betrachteten Topologie-Testproblemen als stabiles und leistungsstarkes Verfahren. In keinem der betrachteten Fälle bricht das Programm vorzeitig ab. Obwohl diese Untersuchung ohne einen Topologie-Filter durchgeführt wurde, halten sich Strukturen mit einem Schachbrettmuster bei TRSCP in Grenzen.

In 11 der betrachteten 32 Fälle benötigt TRSCP mehr Funktionsauswertungen als LSSCP. In zwei Fällen im Vergleich der beiden Algorithmen benötigt der eine weniger Funktions-, dafür aber mehr Gradientenauswertungen. In sechs Fällen bricht LSSCP die Iteration mit einer Fehlermeldung ab. In diesen Fällen ist zumeist der Rechenaufwand von LSSCP immens hoch.

Auffällig ist, daß TRSCP implementiert in das Programm SCPIP4.0 bei den betrachteten Topologieoptimierungsaufgaben keine inneren Iterationen durchgeführt hat. In der Implementierung in die Version SCPIP3.0 war dies der Fall. Die erzeugten Strukturen sind bei beiden Versionen ähnlich, obwohl bei der neueren Version noch weniger Struktur mit einem Schachbrettmuster auftritt. Beide Versionen berechnen eine ähnlich auffällige Struktur für den MBB-Beam (x3.5/20%). Nur im Fall MBB-Beam (x3.5/30%) benötigt TRSCP unverhältnismäßig viel Rechenaufwand. In einigen der anderen Fällen benötigt LSSCP wesentlich mehr Funktions- bzw. Gradientenauswertungen als TRSCP.

Kapitel 5

Fazit und Ausblick

Das Ziel dieser Arbeit war es ein SCP-Verfahren zu entwickeln, das durch eine Trust-Region-Stabilisierung eine globale Konvergenzeigenschaft besitzt. Bei dem neu entwickelten Verfahren wird der Bereich zwischen den Asymptoten als Trust-Region genutzt und keine zusätzliche Trust-Region eingeführt.

Das in dieser Arbeit vorgestellte neue Verfahren Trust-Region Sequential Convex Programming (TRSCP) verbindet die Approximationsidee der MMA-Methoden erfolgreich und auf einfache Weise mit einer Trust-Region-Stabilisierung und ist ein global konvergentes iteratives SCP-Verfahren. Dieser Ansatz kombiniert Nebenbedingungen zum Einhalt der Trust-Region und Schranken Nebenbedingungen, soweit vorhanden. Das heißt, wird ein nichtlineares Optimierungsproblem mit unteren und oberen Schranken für die primale Variable betrachtet, dann entsteht durch diesen Ansatz keine zusätzliche Nebenbedingung, wie das üblicherweise bei einer Trust-Region-Strategie der Fall ist. Im Gegensatz zu anderen kombinierten Trust-Region/Moving-Asymptotes-Ansätzen können mit TRSCP allgemeine kontinuierliche und restringierte nichtlineare Probleme gelöst werden, d.h. Probleme mit Gleichungs- und Ungleichungsnebenbedingungen, sowie Schranken.

Bei TRSCP wird ein fest definierter Teilbereich zwischen den MMA-Asymptoten als Trust-Region angesehen. Durch eine Abstimmung der Schranken des Teilproblems in TRSCP mit dem Trust-Region-Radius wird garantiert, daß jede Lösung eines Teilproblems in der Trust-Region liegt. Durch das Festlegen des Trust-Region-Radius werden bei TRSCP automatisch die MMA-Asymptoten bestimmt. Weiter werden durch die Schranken von NLP und dem Trust-Region-Radius die Schranken des Teilproblems festgelegt. Insgesamt heißt das, durch den Trust-Region-Radius und den Entwicklungspunkt wird bei TRSCP das gesamte Teilproblem von NLP definiert.

Für die Anpassung des Trust-Region-Radius in jedem Iterationsschritt wird bei TRSCP eine Kombination aus klassischer Trust-Region-Strategie und der Strategie von MMA-Methoden zur Bestimmung der Asymptoten verwendet. Dadurch ist es im Gegensatz zu anderen Trust-Region-Verfahren möglich die Trust-Region für jeden Koeffizienten der primalen Variablen individuell anzupassen.

Das Verfahren TRSCP wurde im Programm TRSCP1.0 implementiert. TRSCP1.0 ist eine Weiterentwicklung von SCPIP3.0 bzw. SCPIP4.0 (Zillober [76, 77]) und baut auf dieses Programm auf. Bei den für diese Arbeit durchgeführten Vergleichen von TRSCP mit der SCP-Methode mit Line-Search-Stabilisierung von Zillober [74] (LSSCP) hat sich TRSCP bei den betrachteten Testproblemen zumindest als gleichwertig herausgestellt. Bei den Testproblemen aus dem Bereich der Strukturoptimierung waren klare Vorteile für TRSCP zu erkennen. Im Gegensatz zu dem in SCPIP4.0 implementierten LSSCP gab es bei TRSCP in keinem Fall einen vorzeitigen Iterationsabbruch. Die von TRSCP erzeugten Strukturen weisen, im Vergleich zu den von LSSCP erzeugten Strukturen, wesentlich weniger Bereiche mit einer 'checkerboard-structure' auf. Der benötigte Rechenaufwand beider Verfahren ist bei diesen Topologieoptimierungsproblemen ähnlich, allerdings benötigt TRSCP nur bei 11 der 32 betrachteten Aufgaben mehr Funktions- und Gradientenauswertungen als LSSCP, in 16 Fällen ist dies umgekehrt.

Bei dem SCP-Verfahren mit Line-Search kann es vorkommen, daß mit einer festen Anzahl von Teiliterationen mit Hilfe der Line-Search kein Iterationspunkt gefunden werden kann, der einen für die Konvergenz hinreichenden Abstieg bewirkt. In diesem Fall nimmt SCPIP3.0 die aktuelle Lösung des Teilproblems als neuen Iterationspunkt, obwohl der hinreichende Abstieg nicht gewährleistet ist. In ähnlichen Situationen hat das vorgestellte SCP-Verfahren mit Trust-Region-Stabilisierung einen Vorteil. Hier wird eine sogenannte innere Iteration durchgeführt, d.h. der aktuelle Entwicklungspunkt des Teilproblems wird beibehalten und der Trust-Region-Radius wird reduziert, bis ein Iterationspunkt gefunden wird, der einen hinreichenden Abstieg bewirkt. Zwar wird die Anzahl von aufeinanderfolgenden inneren Iterationen bei TRSCP auch beschränkt, aber diese Schranke wird wesentlich seltener aktiv, als bei der Line-Search-Methode. Bei TRSCP wird durch die Reduzierung des Trust-Region-Radius nicht nur der Bereich, in dem nach einem neuen Iterationspunkt gesucht wird verkleinert, es werden durch die Änderung der Asymptoten auch die Approximationen der Zielfunktion und Ungleichungsnebenbedingungen geändert. Als Folge wird, im Gegensatz zur Line-Search, auch die Suchrichtung angepaßt und nicht nur die Schrittweite. Weiter ist der Iterationspunkt bei einer abgebrochenen inneren Iteration im Vergleich näher am aktuellen Entwicklungspunkt als bei einer abgebrochenen Line-Search, die dann mit der vollen Schrittweite fortfährt. Es ist zu erwarten, daß der Iterationspunkt in diesen Fällen bei TRSCP „weniger falsch“ ist.

Hieraus ergeben sich mehrere Ansätze für eine Fortführung des behandelten Themas, die MMA-Approximation mit einer Trust-Region-Stabilisierung zu kombinieren.

- Ein SCP-Verfahren ist zu untersuchen, das die Vorzüge der MMA-Strategie, die MMA-Asymptoten zu aktualisieren, mit einer Line-Search und, falls diese fehlschlägt, mit der Möglichkeit innere Iterationen durchzuführen kombiniert. Ein solches neues Verfahren sollte außerdem die kombinierte MMA-/Trust-Region-Strategie zum erweitern des Abstands der Asymptoten vom Entwicklungspunkt in äußeren Iterationen benutzen, da diese in geeigneten Situationen eine größere Schrittweite ermöglicht.

- Das Teilproblem von TRSCP und die von TRSCP erzeugten Asymptoten können ohne weiteres mit einer Line-Search kombiniert werden. Ein Vergleich der Leistungsfähigkeit von einem SCP-Verfahren mit Line-Search, welches das Teilproblem und die Asymptoten von TRSCP benutzt, mit dem Verfahren von Zillober [74] wird zeigen, ob sich hierdurch die Leistungsfähigkeit der SCP-Methode von Zillober [74] ändert.
- Es ist zu untersuchen, in wie weit das reduzierte Auftreten von Schachbrettmustern in topologieoptimierten Strukturen genutzt werden kann, auf den Einsatz eines Topologie-Filters zu verzichten. TRSCP stützt sich stark auf den Funktionswert der Gradienten. Es ist weiter zu untersuchen, in wie fern ein Topologie-Filter die Leistung von TRSCP beeinflusst.

Diese Arbeit hat gezeigt, daß TRSCP ein sehr gutes Verfahren ist, Strukturoptimierungsprobleme zu lösen. Weiter ist TRSCP genauso gut wie das Verfahren von Zillober [74] geeignet allgemeine Optimierungsprobleme zu lösen. Eine Weiterführung der in dieser Arbeit entwickelten Vorgehensweisen und Strategien verspricht, allgemein SCP-Verfahren zu verbessern.

Literaturverzeichnis

- [1] BATHE,K.-J., *Finite-Elemente-Methoden*, 2. vollständig neu bearbeitete und erweiterte Auflage, Springer, Berlin 2002
- [2] BENDSØE,M.P., GUADES,J.M., HABER,R.B., PEDERSEN,P., TAYLOR,J.E., *An analytical model to predict optimal material properties in the context of optimal structural design*, Journal on Applied Mechanics 61 (S.930-937), 1994
- [3] BENDSØE,M.P., SIGMUND,O., *Topology Optimization - Theory, Methods and Applications*, Springer, Berlin 2003
- [4] BEN-TAL,A., KOČVARA,M., NEMIROVSKI,A., ZOWE,J., *Free material design via semidefinite programming. The multi-load case with contact conditions*, SIAM Journal on Optimization 9 (S.813-832), 1997
- [5] BERTSEKAS,D.P., *Constrained optimization and Lagrange multiplier methods*, Academic Press, 1982
- [6] BYRD,R.H., SCHNABEL,R.B., SCHULTZ,G.A., *A trust region algorithm for nonlinearly constrained optimization*, SIAM Journal on Numerical Analysis 24 (S.1152-1170), 1987
- [7] CELIS,M.R., DENNIS,J.E.,TAPIA,R.A., *A trust region algorithm for nonlinear equality constrained optimization*, in: Boggs,R.T., Byrd,R.H., Schnabel,R.B. (eds.): Numerical Optimization, SIAM (S.71-82), 1985
- [8] CONN,A., GOULD,N. AND TOINT,P., *A globally convergent augmented Lagrangian algorithm for optimization with general constraints and simple bounds*, SIAM Journal on Numerical Analysis 28 (2) (S.545-572), 1991
- [9] CONN,A., GOULD,N. AND TOINT,P., *Trust Region Methods*, MPS/SIAM Series on Optimization, 2000
- [10] DENNIS,J.E., EL-ALEM,M., MACIEL,M.C., *A global convergence theory for general trust-region based algorithms for equality constrained optimization*, SIAM Journal on Optimization 7 (S.177-207), 1997
- [11] ERTEL,S., *Anwendungen von Filtermethoden auf das Optimierungsverfahren SCP*, Diplomarbeit, Mathematisches Institut, Universität Bayreuth, 2006

- [12] ERTEL,S., SCHITTKOWSKI,K., ZILLOBER, C., *D26 - Sequential Convex Programming for Free Material Optimization*, Report, 2008
- [13] ERTEL,S., SCHITTKOWSKI,K., ZILLOBER, C., *A Feasible Sequential Convex Programming Method for Free Material Optimization*, 8th World Congress on Structural and Multidisciplinary Optimization, June 1-5, 2009, Lisbon, Portugal
- [14] FIACCO,A.V., *Introduction to Sensitivity and Stability Analysis in Nonlinear Programming*, Academic Press, Inc., New York, 1982
- [15] FLETCHER,R., *An exact penalty function for nonlinear programming with equalities*, Mathematical Programming 5 (S.129-150), 1973
- [16] FLETCHER,R., *An ideal penalty function for constrained optimization*, Nonlinear Programming 2 (S.121-164), 1975
- [17] FLETCHER,R., *Practical Methods of Optimization*, John Wiley and Sons, 1985
- [18] FLETCHER,R., LEYFFER,S., *Nonlinear programming without a penalty function*, Mathematical Programming 91 (2) (S.239-269), 2002
- [19] FLETCHER,R., GOULD,N., LEYFFER,S., TOINT,P., *Global convergence of trust-region SQP-filter algorithms for general nonlinear programming*, SIAM Journal on Optimization 13 (3) (S.635-659), 2002
- [20] FLEURY,C., BRAIBANT,V., *Structural Optimization - a new dual method using mixed variables*, International Journal for Numerical Methods in Engineering 23 (S.409-428), 1986
- [21] GOULD,N., TOINT,P., *Nonlinear programming without a penalty function or a filter*, Mathematical Programming 122 (1) (S.155-196), 2010
- [22] HAFTKA,R.T., GÜRDAL,Z., *Elements of Structural Optimization*, 3. Auflage, Kluwer Academic Publishers, 1991
- [23] HAHN,H.G., *Methoden der finiten Elemente in der Festigkeitslehre*, 2. unv. Auflage, Akademische Verlagsgesellschaft, Frankfurt am Main 1982
- [24] HARZHEIM,L., *Strukturoptimierung - Grundlagen und Anwendung*, 1. Auflage, Harri Deutsch Verlag, 2007
- [25] HAUG,E.J., ARORA,J.S., *Applied Optimal Design: Mechanical and Structural Systems*, Wiley & Sons, 1979
- [26] HESTENES,M.R., *Multiplier and gradient methods*, Journal of Optimization Theory and Applications 4 (S.303-320), 1969
- [27] HOCK,W., SCHITTKOWSKI,K., *Test Examples for Nonlinear Programming Codes*, Lecture Notes in Economics and Mathematical Systems (187), Springer, Berlin 1981

- [28] KOČVARA, M., ZOWE, J., *Free material optimization: An overview*, in: Siddiqi, A.H., Kočvara, M. (eds.): Trends in Industrial and Applied Mathematics (S.181-215), Kluwer Academic Publishers, Dordrecht 2002
- [29] KRÄTZIG, W., B., WITTEK, U., HARTE, R., MESKOURIS, K., *Tragwerke 1-Theorie und Berechnungsmethoden statisch bestimmter Stabtragwerke*, 4. Auflage Springer, 1999
- [30] LEHMANN, S., *A Strictly Feasible Sequential Convex Programming Method*, Dissertation, Fakultät für Mathematik, Physik und Informatik der Universität Bayreuth, Bayreuth 2011 (eingereicht)
- [31] LEVENBERG, K., *A method for the solution of certain nonlinear problems in least squares*, Quarterly Journal of Applied Mathematics 2 (S.164-168), 1944
- [32] MARQUARDT, D.W., *An algorithm for least-squares estimation of nonlinear inequalities*, SIAM Journal of Applied Mathematics 11 (S.431-441), 1963
- [33] MORÈ, J.J., *The Levenberg-Marquardt algorithm: implementation and theory*, Watson, G.A.: Proc. Dundee Conference on Numerical Analysis, Springer, Berlin 1978
- [34] MORÈ, J.J., *Recent developments in algorithms and software for trust region methods*, in: Bachem, A., Grötschel, M., Korte, B. (eds.): Mathematical Programming: The State of the Art (S.258-287), Springer, Berlin 1983
- [35] NI, Q., *A globally convergent Method of Moving Asymptotes with Trust Region Technique*, Optimization Methods and Software 18 (S.283-297), 2003
- [36] NOCEDAL, J., YUAN, Y., *Combining trust region and line search techniques*, in: Yuan, Y. (ed.): Advances in Nonlinear Programming (S.153-175), Kluwer, 1998
- [37] OMOJOKUN, E.O., *Trust Region Algorithms for Optimization with Nonlinear Equality and Inequality Constraints*, Ph.D. Thesis, University of Colorado at Boulder, 1989
- [38] PAPALAMBROS, P.Y., WILDE, D.J., *Principles of Optimal Design: Modeling and Computation*, 2. Auflage, Cambridge University Press, Oxford 2000
- [39] POWELL, M.J.D., *A method for nonlinear constraints in minimization problems*, in: R.Fletcher (ed.): Optimization (S.283-298), Academic Press, 1969
- [40] POWELL, M.J.D., *A new algorithm for unconstrained optimization*, in: Rosen, J.B., Mangassarian, O.L., Ritter, K. (eds.): Nonlinear Programming (S.31-66), Academic Press, New York 1970
- [41] POWELL, M.J.D., *Convergence properties of a class of minimization algorithms*, in: Mangassarian, O.L., Meyer, R.R., Robinson, S.M. (eds.): Nonlinear Programming 2 (S.1-27), Academic Press, New York 1975

- [42] POWELL,M.J.D., *Algorithms for nonlinear constraints that use Lagrangian functions*, Mathematical Programming 14 (S.224-248), 1978
- [43] POWELL,M.J.D., *On the global convergence of trust region algorithm for unconstrained optimization*, Mathematical Programming 49 (S.297-303), 1984
- [44] POWELL,M.J.D., YUAN,Y., *A trust region algorithm for equally constrained optimization*, Mathematical Programming 49 (S.189-211), 1991
- [45] SCHENK,R., *Strukturoptimierung unter Verwendung dualer Verfahren mit expliziter Approximation der verformungsabhängigen Nebenbedingungen*, Diplomarbeit, Institut für Computer-Anwendungen der Universität Stuttgart, 1989
- [46] SCHITTKOWSKI,K., *The Nonlinear Programming Method of Wilson, Han and Powell with an Augmented Lagrangian Type Line Search Function*, Numerische Mathematik 38 (S.83-114), 1981
- [47] SCHITTKOWSKI,K., *On the Convergence of a Sequential Quadratic Programming Method with an Augmented Lagrangian Line Search Function*, Mathematische Operationsforschung und Statistik, Series Optimization 14 (2) (S.197-216), 1983
- [48] SCHITTKOWSKI,K., *NLPQL: A Fortran subroutine solving constrained nonlinear programming problems*, Annals of Operations Research 5 (S.485-500), 1986
- [49] SCHITTKOWSKI,K., *More Test Examples for Nonlinear Programming*, Lecture Notes in Economics and Mathematical Systems 182, Springer, 1987
- [50] SCHITTKOWSKI,K., ZILLOBER,C.,ZOTEMANTEL,R., *Numerical comparison of nonlinear programming algorithms for structural optimization*, Structural Optimization 7 (S.1-19), 1994
- [51] SCHITTKOWSKI,K., YUAN,Y., *Sequential Quadratic Programming Methods*, erscheint in Wiley Encyclopedia of Operations Research and Management Science, 2010
- [52] SCHITTKOWSKI,K., *An Updated Set of 306 Test Problems for Nonlinear Programming with Validated Optimal Solutions*, User Guide, February 2011
www.ai7.uni-bayreuth.de/test_problems.pdf
www.ai7.uni-bayreuth.de/test_probs_src.zip
- [53] SIGMUND,O., *A 99 line topology optimization code written in MATLAB*, Structural and Multidisciplinary Optimization 21 (S.120-127), 2001
- [54] SIGMUND,O., *Exercises with 'A 99 line topology optimization code written in Matlab'*, DCAMM-course: Topology Optimization - Theory, Methods and Applications (held at DTU, Lyngby, Denmark, June 10-16, 2009),
- [55] SORENSEN,D.C., *Newton's method with a trust region modifications*, SIAM Journal on Numerical Analysis 19 (S.409-426), 1982

- [56] STINGL, M., KOČVARA, M., LEUGERING, G., *A Sequential Convex Semidefinite Programming Algorithm with an Application to Multiple-Load Free Material Optimization*, SIAM Journal on Optimization 20 (1) (S.130-155), 2009
- [57] SVANBERG, K., *The Method of Moving Asymptotes - A New Method For Structural Optimization*, International Journal for Numerical Methods in Engineering 24 (S.359-373), 1987
- [58] SVANBERG, K., *The Method of Moving Asymptotes (MMA) with some Extensions*, Optimization of large structural systems (S.555-566), 1993
- [59] SVANBERG, K., *A globally Convergent Version of MMA without Linesearch*, Proceedings of the First World Congress of Structural and Multidisciplinary Optimizations (S.9-16), 1995
- [60] SVANBERG, K., *A new globally convergent version of the method of moving asymptotes*, Tech. Rep. TRITA/MAT-99-OS2, Department of Mathematics, KTH Stockholm 1999
- [61] SVANBERG, K., *A class of globally convergent Optimization Methods based on Conservative Convex Seperable Approximations*, SIAM Optimization Journals 12 (S.555-573), Stockholm 2002
- [62] ULBRICH, M., ULBRICH, S., *Non-monotone trust-region methods for nonlinear equality constrained optimization without a penalty function*, Mathematical Programming 95 (2) (S.103-135), 2003
- [63] VANDERPLAATS, G.N., *Numerical Optimization Techniques for Engineering Design*, McGraw-Hill, 1984
- [64] VARDI, A., *A trust region algorithm for equality constrained minimization: convergence properties and implementation*, SIAM Journal on Numerical Analysis 22 (S.575-591), 1985
- [65] YUAN, Y., *On the Convergence of a new Trust Region Algorithm*, Numerische Mathematik 70 (S.515-539), 1995
- [66] YUAN, Y., *A review of trust region algorithms for optimization*, in: J.M.Ball and J.C.R.Hunt (eds.), ICM99: Proceedings of the Fourth International Congress on Industrial and Applied Mathematics, Oxford University Press (S.271-282), 2000
- [67] ZIENKIEWICZ, O.C., TAYLOR, R.L., *The Finite Element Method Set*, 6. Auflage, Butterworth-Heinemann, Oxford 2005
- [68] ZIENKIEWICZ, O.C., TAYLOR, R.L., *The Finite Element Method for Solid and Structural Mechanics*, 6. Auflage, Butterworth-Heinemann, Oxford 2005

- [69] ZILLOBER,C., *Eine global konvergente Methode zur Lösung von Problemen aus der Strukturoptimierung*, Dissertation, Institut für Angewandte Mathematik und Statistik der Technischen Universität München, München 1992
- [70] ZILLOBER,C., *A globally convergent version of the method of moving asymptotes*, Structural Optimization 6 (S.166-174), 1993
- [71] ZILLOBER,C., *A practical interior-point method for a Nonlinear Programming problem arising in Sequential Convex Programming*, Technical Report Nr. TR98-1, Informatik, Universität Bayreuth, 1998
- [72] ZILLOBER,C.,VOGEL,F., *Solving large scale structural optimization problems*, in: Proceedings of the 2nd ASMO UK/ISSMO conference on Engineering Design Optimization (S.273-280), J.Sienz ed., University of Swansea, Wales, 2000
- [73] ZILLOBER,C., *Global convergence of a nonlinear programming method using convex approximations*, Numerical Algorithms 27 (S.265-289), 2001
- [74] ZILLOBER,C., *Numerical Solution of Nonlinear Programming Problems by Convex Approximation Methods*, Habilitationsschrift, Fakultät für Mathematik und Physik der Universität Bayreuth, Bayreuth 2001
- [75] ZILLOBER,C., *A combined convex approximation - interior point approach for large scale nonlinear programming*, Optimization and Engineering 2 (S.51-73), 2001
- [76] ZILLOBER C., *Software manual for SCIP3.0*, Technical Report, Informatik, Universität Bayreuth, 2004
www.mathematik.uni-wuerzburg.de/~zillober/pubs/manual30.pdf
- [77] ZILLOBER,C., *SCIP - an efficient tool for the solution of structural optimization problems*, Structural and Multidisciplinary Optimization 24 (S.362-371), 2002
- [78] ZILLOBER,C., SCHITTKOWSKI,K., MORITZEN,K., *Very large scale optimization by sequential convex programming*, Optimization Methods and Software 19 (1) (S.103-120), 2004
- [79] ZOWE,J., KOČVARA,M., BENDSØE,M.P., *Free material optimization via mathematical programming*, Mathematical Programming Series B 79 (S.445-466), 1997

Danksagung

Ich danke allen Leuten, die mich bei der Erstellung dieser Arbeit unterstützt und begleitet haben. Leuten, die mich in über neun Jahre lang ermuntert haben weiter zu arbeiten und, die nie den Glauben an mich verloren haben. Es war eine nicht immer leichte, aber sehr lehrreiche Zeit für mich!

Im besonderen möchte ich mich bei meinem Betreuer Herrn Prof. Dr. Klaus Schittkowski bedanken, der mich zu dieser Arbeit ermuntert und seinen externen Doktoranden nie aus den Augen verloren hat. Die fachlichen Diskussionen haben mir einen tiefes Verständnis der Materie gegeben, die ich ansonsten nie erlangt hätte. Ich bedanke mich bei Herrn PD Dr. Christian Zillober für die Bereitstellung seines Programms SCPIP3.0 und dafür, jederzeit meine Fragen zu seinem Programm, sowie zur sequentiellen konvexen Optimierung umfassend zu beantworten.

Ich möchte mich bei meinen Vorgesetzten bei der EADS Deutschland GmbH bedanken, die es mir durch einen Teilzeitarbeitsvertrag ermöglicht haben, die Zeit zu finden mich wissenschaftlich zu betätigen.

Außerdem danke ich meinem Kollegen Dr. Heinz Ower für die anregenden Diskussionen zum Thema wissenschaftliches Arbeiten und den Anforderungen einer Dissertation.

Ich danke meinem Kollegen Thorsten Schröteler, der mich bei programmiertechnischen Problemen beraten hat und mir geholfen hat in der Linux-Welt nicht verloren zu gehen.

Ich bedanke mich bei meinen Bayreuther Mitstreitern Sonja Lehmann, Thomas Lehmann und Oliver Exler dafür, daß sie den externen seltenen „Gast“ in ihrer Gruppe aufgenommen haben. Durch euch bin ich mir an der Uni niemals fremd vorgekommen. Außerdem waren die Diskussionen zu SCP mit Sonja und zu Trust-Regions mit Oliver von unschätzbarem Wert für mich.

Abschließend danke ich meiner Frau Eva, für die moralische Unterstützung und das Verständnis für meine Launen, wenn die Arbeit nicht so wollte, wie ich das gerne gehabt hätte und ich danke meinen Eltern und meinem Bruder ebenfalls für die Unterstützung und die vielen Kleinigkeiten, die sie für mich aus meinem Weg geräumt haben.

Axel