

# Mathematische Grundlagen von Optimierungsverfahren

K. Schittkowski <sup>1</sup>

## 1. Einführung

### 1.1 Das nichtlineare Optimierungsproblem

Es werden Optimierungsaufgaben betrachtet, die dadurch charakterisiert sind, daß eine Zielfunktion  $f$  unter nichtlinearen Gleichungs- und Ungleichungsnebenbedingungen minimiert wird, d.h.

$$\begin{aligned} & \min f(x) \\ & g_j(x) = 0 \quad , \quad j = 1, \dots, m_e \\ x \in \mathbb{R}^n : & \quad g_j(x) \geq 0 \quad , \quad j = m_e + 1, \dots, m \\ & x_l \leq x \leq x_u \end{aligned}$$

Für die nachfolgenden theoretischen Untersuchungen soll vorausgesetzt werden, daß die oberen und unteren Schranken nicht existieren bzw. als allgemeine Restriktionen formuliert werden. Das obige Problem wird im Folgenden mit (NLP) bezeichnet.

Besitzen Zielfunktion  $f$  und der zulässige Bereich bzw. die Nebenbedingungen  $g_j$  eine spezielle Gestalt, so können zur Lösung des obigen Problems u.U. Spezialverfahren herangezogen werden.

Für die Zielfunktion sind folgende Strukturen interessant:

Allgemeine nichtlineare Zielfunktion  $f(x)$

Lineare Zielfunktion  $f(x) = c^T x$

Quadratische Zielfunktion  $f(x) = \frac{1}{2}x^T Cx + d^T x$

Summe von Quadraten (Regression)  $f(x) = \sum_{i=1}^m (y_i - f(x, t_i))^2,$   
( $y_i$  Meßwert zum Zeitpunkt  $t_i$ )

Maximum von Funktionen  $f(x) = \max_{j=1, \dots, m} f_j(x)$

---

<sup>1</sup>Mathematisches Institut, Universität Bayreuth, D-95440 Bayreuth

In Bezug auf die Nebenbedingungen sind folgende Situationen typisch:

allgemeine nichtlineare Nebenbedingungen

lineare Nebenbedingungen

keine Nebenbedingungen

Obgleich viele Softwaresysteme zur mathematischen Optimierung als geschlossene Systeme konzipiert werden, sollte auch der Anwender einige theoretische Grundkenntnisse besitzen, um die Dokumentation besser zu verstehen und um den Output insbesondere im Fehlerfall analysieren zu können. Für spezielle Anwendungsbereiche oder aus organisatorischen Gründen kann der Entwurf eines eigenen Programms erforderlich sein, was ebenfalls Grundkenntnisse aus der Optimierungstheorie voraussetzt.

## 1.2 Definitionen und Beispiele

Zum grundlegenden Begriff der Ableitung einer Funktion  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  benötigen wir folgende Bezeichnungen:

- Gradient einer Funktion:

$$\nabla f(x) := \left( \frac{\partial}{\partial x_1} f(x), \dots, \frac{\partial}{\partial x_n} f(x) \right)^T$$

- Hesse-Matrix einer Funktion:

$$\nabla^2 f(x) := \left( \frac{\partial^2}{\partial x_i \partial x_j} f(x) \right)_{i,j=1,\dots,n}$$

Die Optimierungstheorie für den vorliegenden Problemtyp basiert auf dem Begriff der Lagrange-Funktion, mit deren Hilfe erst die notwendigen und hinreichenden Optimalitätskriterien aufgestellt werden können.

### Definition:

- a) Der zulässige Bereich  $P$  des Problems (NLP) ist die Menge aller zulässigen Lösungen, d.h.

$$P := \{x : g_j(x) = 0, j = 1, \dots, m_e, g_j(x) \geq 0, j = m_e + 1, \dots, m\}$$

- b) Die aktiven Restriktionen bzgl.  $x \in P$  sind charakterisiert durch die Indexmenge

$$I(x) := \{j : g_j(x) = 0, m_e < j \leq m\}$$

- c) Die Lagrange-Funktion von (NLP) ist durch

$$L(x, u) := f(x) - \sum_{j=1}^m u_j g_j(x)$$

für jedes  $x \in \mathbb{R}^n$  und  $u = (u_1, \dots, u_m)^T \in \mathbb{R}^m$  definiert. Die Variablen  $u_j$  heißen Lagrange-Multiplikatoren.

**Beispiel:** Ein Optimierungsproblem sei durch die Funktionen

$$\begin{aligned} f(x_1, x_2) &:= x_1^2 + x_2 \\ g_1(x_1, x_2) &:= 9 - x_1^2 - x_2^2 \geq 0 \\ g_2(x_1, x_2) &:= 1 - x_1 - x_2 \geq 0 \end{aligned}$$

gegeben. Die Lagrange-Funktion des Problems ist

$$L(x, u) = x_1^2 + x_2 - u_1(9 - x_1^2 - x_2^2) - u_2(1 - x_1 - x_2)$$

Für die Optimallösung  $x^*$  gilt

$$x^* = (0, -3)^T, \quad I(x^*) = \{1\}$$

Man kann i.a. nur erwarten, daß ein Optimierungsverfahren ein lokales Minimum und nicht ein globales berechnet, d.h. einen Punkt  $x^*$  mit  $f(x^*) \leq f(x)$  für alle  $x \in P \cap U(x^*)$ ,  $U(x^*)$  geeignete Umgebung von  $x^*$ . Jedes lokale Minimum von (NLP) ist dagegen ein globales, wenn das Ausgangsproblem konvex ist, d.h. wenn  $f$  konvex,  $g_j$  linear für  $j = 1, \dots, m_e$  und  $g_j$  konvex für  $j = m_e + 1, \dots, m$  ist. Hierdurch wird erzwungen, daß der zulässige Bereich  $P$  eine konvexe Menge ist.

**Definition:** Eine Funktion  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  heißt konvex, falls

$$f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y)$$

für alle  $x, y \in \mathbb{R}^n$  und  $\lambda \in (0, 1)$  gilt, und konvex, falls in obiger Ungleichung ' $\leq$ ' durch ' $\geq$ ' ersetzt wird.

Für eine zweimal stetig differenzierbare Funktion  $f$  ist Konvexität gleichbedeutend damit, daß  $\nabla^2 f(x)$  positiv semidefinit ist, d.h.  $z^T \nabla^2 f(x) z \geq 0$  für alle  $z \in \mathbb{R}^n$ . Die Bedeutung der Konvexität für Optimierungsalgorithmen besteht darin, daß häufig nur für diesen Spezialfall Konvergenzaussagen bewiesen werden können.

Zur Formulierung der nachfolgenden Charakterisierungssätze benötigt man eine Zusatzbedingung, um pathologische Fälle von  $P$  in einer Optimallösung auszuschließen. Sie heißt *constraint qualification*, Slater-Bedingung oder im allgemeinsten Fall Regularitätsbedingung. Der Einfachheit halber beschränken wir uns auf folgende Formulierung:

**Definition:** Unter einer *constraint qualification* in  $x^* \in P$  verstehen wir die lineare Unabhängigkeit der Vektoren  $\nabla g_j(x^*)$  für alle  $j \in \{1, \dots, m_e\} \cup I(x^*)$ .

**Beispiel:** Nebenbedingungen seien durch

$$\begin{aligned} g_1(x_1, x_2) &:= x_2 && \geq 0 \\ g_2(x_1, x_2) &:= -x_2 + x_1^2 && \geq 0 \end{aligned}$$

gegeben, und es sei  $x^* := (0, 0)^T$ . Wegen  $\nabla g_1(x^*) = (0, 1)^T$ ,  $\nabla g_2(x^*) = (0, -1)^T$  ist  $\nabla g_1(x^*) + \nabla g_2(x^*) = 0$  und damit die *constraint qualification* nicht erfüllt.

### 1.3 Sattelpunkt und Dualität

Wir beginnen den Abschnitt mit einer einfach zu beweisenden Aussage über die Bedeutung der Sattelpunkteigenschaft für die nichtlineare Programmierung.

**Satz:** Falls  $x^*, u^*$  Sattelpunkt der Lagrange-Funktion mit  $u_j^* \geq 0$  für  $j = m_e + 1, \dots, m$  ist, d.h. falls

$$L(x^*, u) \leq L(x^*, u^*) \leq L(x, u^*)$$

für alle  $x \in \mathbb{R}^n$  und  $u \in \mathbb{R}^m$  mit  $u_j \geq 0$  für  $j = m_e + 1, \dots, m$  gilt, ist  $x^*$  globale Lösung des Problems (NLP).

*Beweis:* Es gilt

$$\begin{aligned} f(x^*) - \sum_{j=1}^m u_j g_j(x^*) &\leq f(x^*) - \sum_{j=1}^m u_j^* g_j(x^*) \\ &\leq f(x) - \sum_{j=1}^m u_j^* g_j(x). \end{aligned}$$

Zuerst ist  $x^* \in P$  zu zeigen. Dazu nehme man an, daß  $g_j(x^*) \neq 0$  für ein  $j \leq m_e$  oder  $g_j(x^*) < 0$  für ein  $j > m_e$  gilt. Setze  $u^{(r)} = r\sigma e_j$ , wobei  $e_j$  der  $j$ -te Einheitsvektor im  $\mathbb{R}^m$  ist und  $\sigma := -1$ , falls  $g_j(x^*) > 0$ ,  $\sigma := +1$ , falls  $g_j(x^*) < 0$ . Dann gilt

$$L(x^*, u^{(r)}) = f(x^*) - r\sigma g_j(x^*) \rightarrow \infty \text{ für } r \rightarrow \infty$$

was zu einem Widerspruch zur Beschränktheit von  $L(x^*, u^{(r)})$  durch  $L(x^*, u^*)$  führt. Für  $u := 0$  und  $x \in P$  gilt

$$f(x^*) \leq f(x) - \sum_{j=1}^m u_j^* g_j(x) = f(x) - \sum_{j=m_e+1}^m u_j^* g_j(x) \leq f(x)$$

Damit ist  $x^*$  optimal für (NLP).

Die Bedeutung dieses Satzes liegt darin, daß bei hinreichend genauer Kenntnis der optimalen Lagrange-Multiplikatoren  $u^*$  die Lösung des restringierten Problems (NLP) auf die Lösung eines unrestringierten Problems zurückgeführt werden kann. Allgemein spielen die Lagrange-Multiplikatoren bei der Entwicklung neuerer Optimierungsverfahren eine bedeutende Rolle. Man kann davon ausgehen, daß ein vergleichbarer Aufwand zur Approximation der optimalen Multiplikatoren  $u^*$  betrieben wird wie zur Approximation der Optimallösung  $x^*$ . Zur Umkehrung des Satzes werden allerdings verschärfte Voraussetzungen benötigt.

**Satz:** Falls (NLP) konvex,  $x^*$  Optimallösung von (NLP) und die constraint qualification in  $x^*$  erfüllt ist, gibt es ein  $u^* = (u_1^*, \dots, u_m^*)^T \in \mathbb{R}^m$  mit  $u_j^* \geq 0$  für  $j > m_e$ , so daß  $x^*, u^*$  Sattelpunkt der Lagrange-Funktion ist.

**Beispiel:** Sei

$$\begin{aligned} \min \quad & x_1^2 + x_2 \\ x_1, x_2 : \quad & 9 - x_1^2 - x_2^2 \geq 0 \\ & 1 - x_1 - x_2 \geq 0 \end{aligned}$$

wie vorher gegeben. Dann bilden  $x^* = (0, -3)$  und  $u^* = (1/6, 0)$  einen Sattelpunkt der Lagrange-Funktion

$$L(x, u) = x_1^2 + x_2 - u_1(9 - x_1^2 - x_2^2) - u_2(1 - x_1 - x_2)$$

da

$$L(x^*, u) = -3 - 4u_2 \leq L(x^*, u^*) = -3,$$

$$L(x^*, u^*) = -3 \leq \frac{7}{6}x_1^2 + \frac{1}{6}(x_2^2 + 3)^2 - 3 = L(x, u^*)$$

für alle  $u_1 \geq 0$ ,  $u_2 \geq 0$  und  $x_1 \in \mathbb{R}$ ,  $x_2 \in \mathbb{R}$ .

Die Sattelpunkteigenschaft führt zur Definition dualer Probleme.

**Definition:**

1. Das primale Problem lautet

$$\min\{\bar{L}(x) : x \in \mathbb{R}^n\},$$

wobei

$$\bar{L}(x) := \max\{L(x, u) : u \in \mathbb{R}^m, u_j \geq 0 \text{ für } j > m_e\}$$

2. Das duale Problem lautet

$$\max\{\underline{L}(u) : u \in \mathbb{R}^m, u_j \geq 0 \text{ für } j > m_e\},$$

wobei

$$\underline{L}(u) := \min\{L(x, u) : x \in \mathbb{R}^n\}$$

Damit kann der folgende Dualitätssatz bewiesen werden, aus dem auch unmittelbar der Dualitätssatz der linearen Programmierung folgt:

**Satz:** *Es gilt*

$$\max\{\underline{L}(u) : u \in \mathbb{R}^m, u_j \geq 0 \text{ für } j > m_e\} \leq \min\{\bar{L}(x) : x \in \mathbb{R}^n\}$$

*Falls  $x^*, u^*$  Sattelpunkt von  $L$  ist, gilt*

$$\underline{L}(u^*) = \bar{L}(x^*)$$

## 1.4 Notwendige und hinreichende Optimalitätsbedingungen

Für die Entwicklung und das Verständnis von Optimierungsverfahren sind die folgenden Charakterisierungssätze fundamental.

**Satz: (Notwendige Optimalitätsbedingungen 2. Ordnung):** *Es seien  $f$  und  $g_j$  für  $j = 1, \dots, m$  zweimal stetig differenzierbar,  $x^*$  ein lokales Minimum von (NLP) und die constraint qualification in  $x^*$  erfüllt. Dann gibt es ein  $u^* \in \mathbb{R}^m$ , so daß für  $z^* = (x^*, u^*)^T$  gilt*

a)

$$\begin{aligned} u_j^* &\geq 0 \quad , \quad j = m_e + 1, \dots, m \\ g_j(x^*) &= 0 \quad , \quad j = 1, \dots, m_e \\ g_j(x^*) &\geq 0 \quad , \quad j = m_e + 1, \dots, m \\ \nabla_x L(x^*, u^*) &= 0 \\ u_j^* g_j(x^*) &= 0 \quad , \quad j = m_e + 1, \dots, m \end{aligned}$$

b)  $s^T \nabla_x^2 L(x^*, u^*) s \geq 0$  für alle  $s \in \mathbb{R}^n$  mit  $\nabla g_j(x^*)^T s = 0$ ,  $j \in \{1, \dots, m_e\} \cup I(x^*)$ .

Die Bedingung a) des obigen Satzes heißt Kuhn-Tucker-Bedingung. Sie besagt, daß in einem Minimum der Gradient der Zielfunktion eine positive Linearkombination der Gradienten der aktiven Restriktionen ist. Die Aussage b) bedeutet, daß die Lagrange-Funktion auf dem Tangentialraum der Gleichheits- und aktiven Nebenbedingungen positiv semidefinit ist. Auf die Zusatzbedingung der *constraint qualification* kann nicht verzichtet werden.

**Beispiel:** Sei

$$\begin{aligned} f(x_1, x_2) &:= x_1 \\ g_1(x_1, x_2) &:= -x_2 \geq 0 \\ g_2(x_1, x_2) &:= x_2 - x_1^2 \geq 0 \end{aligned}$$

Wegen  $P = \{(0, 0)\}$  ist  $x^* = (0, 0)$  optimal. Allerdings gilt wegen

$$\nabla_x L(x^*, u^*) = \begin{pmatrix} 1 \\ u_1^* - u_2^* \end{pmatrix} \neq \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

nicht die Kuhn-Tucker-Bedingung.

**Satz: (Hinreichende Optimalitätsbedingungen 2. Ordnung):** Es seien  $f$  und  $g_j$  für  $j = 1, \dots, m$  zweimal stetig differenzierbar und  $x^* \in \mathbb{R}^n$ ,  $u^* \in \mathbb{R}^m$  gegeben, so daß gilt:

a)

$$\begin{aligned} u_j^* &\geq 0 \quad , \quad j = m_e + 1, \dots, m \\ g_j(x^*) &= 0 \quad , \quad j = 1, \dots, m_e \\ g_j(x^*) &\geq 0 \quad , \quad j = m_e + 1, \dots, m \\ \nabla_x L(x^*, u^*) &= 0 \\ u_j^* g_j(x^*) &= 0 \quad , \quad j = m_e + 1, \dots, m \end{aligned}$$

b)  $s^T \nabla_{xx} L(x^*, u^*) s > 0$  für alle  $s \in \mathbb{R}^n$  mit  $s \neq 0$ ,  $\nabla g_j(x^*)^T s = 0$ ,  $j = 1, \dots, m_e$ , und für alle  $s$  mit  $\nabla g_j(x^*)^T s = 0$ ,  $j = m_e + 1, \dots, m$ , für die  $u_j^* > 0$  gilt.

Dann ist  $x^*$  ein isoliertes lokales Minimum von  $f$  auf  $P$ , d.h. es gibt eine Umgebung  $U(x^*)$  von  $x^*$ , so daß  $f(x^*) < f(x)$  für alle  $x \in U(x^*) \cap P$ ,  $x \neq x^*$ , gilt.

Beim Studium entsprechender Lehrbücher ist darauf zu achten, daß die Bedingungen a) und b) häufig anders formuliert werden (z.B. für Maximierungs- statt Minimierungsprobleme, ' $\leq$ ' statt ' $\geq$ '-Restriktionen, zusätzliche Schranken der Form ' $x \geq 0$ '), und daß von modifizierten Voraussetzungen ausgegangen werden kann.

**Beispiel:**

$$\begin{aligned} f(x) &:= x_1^2 + x_2 \\ g_1(x) &:= 9 - x_1^2 - x_2^2 \geq 0 \\ g_2(x) &:= 1 - x_1 - x_2 \geq 0 \end{aligned}$$

Die Kuhn-Tucker-Bedingung

$$\nabla_x L(x, u) = \begin{pmatrix} 2x_1 \\ 1 \end{pmatrix} - u_1 \begin{pmatrix} +2x_1 \\ +2x_2 \end{pmatrix} - u_2 \begin{pmatrix} +1 \\ +1 \end{pmatrix} = \begin{pmatrix} 2x_1(1 + u_1) + u_2 \\ 1 + 2u_1x_2 + u_2 \end{pmatrix} = 0$$

ergibt für  $x^* = (0, -3)^T$  die Multiplikatoren  $u_1^* = 1/6$  und  $u_2^* = 0$ . Ferner ist die Hesse-Matrix der Lagrange-Funktion

$$\nabla_x^2 L(x^*, u^*) = \begin{pmatrix} 7/3 & 0 \\ 0 & 1/3 \end{pmatrix}$$

positiv definit.

## 2. Unrestringierte Optimierung

### 2.1 Problemstellung

Abgesehen von ihrer praktischen Bedeutung werden Verfahren zur restringierten Programmierung hier behandelt, weil die Lösung von restringierten Problemen häufig auf die von unrestringierten Problemen zurückgeführt wird, und weil sie ferner als Ansatzpunkt zur Entwicklung von Verfahren zur restringierten Programmierung herangezogen werden. Das Problem lautet also

$$\min_{x \in \mathbb{R}^n} f(x)$$

mit einer (der Einfachheit halber) zweimal stetig differenzierbaren Funktion  $f$ . Die behandelten Verfahren erzeugen eine Folge von Näherungslösungen nach der Iterationsvorschrift

$$x_{k+1} := x_k + \alpha_k d_k,$$

wobei der Startpunkt  $x_0 \in \mathbb{R}^n$  beliebig vorbestimmt wird und  $\alpha_k$  nach einem geeigneten Schrittweitenalgorithmus berechnet wird, auf den wir hier nicht näher eingehen wollen.

Eine intuitiv naheliegende Idee wäre, das Verfahren des steilsten Abstiegs zur Bestimmung der Suchrichtung zu wählen, d.h.

$$d_k := -\nabla f(x_k)$$

zu setzen. In diesem Fall gilt folgendes Resultat:

**Satz:** Es sei  $f(x) := \frac{1}{2}x^T A x$ ,  $A$  positiv definit,  $d_k := -\nabla f(x_k)$  und  $\alpha_k$  durch exakte eindimensionale Minimierung bestimmt. Dann gilt

$$f(x_{k+1}) \leq \left( \frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1} \right)^2 f(x_k)$$

für alle  $k \geq 0$ , wobei  $\lambda_1$  der kleinste und  $\lambda_n$  der größte Eigenwert von  $A$  ist.

Exakte eindimensionale Minimierung bedeutet hier, daß  $\alpha_k$  die Funktion  $f(x_k + \alpha d_k)$  exakt minimiert. Die Abschätzung des Satzes ist scharf in dem Sinn, daß es Startwerte  $x_0$  und Matrizen  $A$  gibt, die Gleichheit für alle  $k$  erzwingen. Für schlecht konditionierte Matrizen ist  $\frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1} \sim 1$ , so daß das in der Praxis beobachtete langsame Konvergenzverhalten theoretisch verifizierbar ist.

## 2.2 Verfahren mit konjugierten Richtungen

Ziel ist jetzt die Beschreibung von Verfahren, die die Nachteile des steilsten Abstiegs vermeiden. Dazu müssen die Suchrichtungen zusätzlich Information über das Krümmungsverhalten der Zielfunktion *ansammeln*.

**Definition:** Es sei  $s_1, \dots, s_n \in \mathbb{R}^n, s_i \neq 0, A$  positiv definit. Dann heißen die Vektoren  $s_1, \dots, s_n$  bzgl.  $A$  konjugiert, falls

$$s_i^T A s_j = 0$$

für  $i, j = 1, \dots, n, i \neq j$ .

Man sieht leicht, daß konjugierte Vektoren linear unabhängig sind. Ein Verfahren, das konjugierte Richtungen erzeugt, muß also bei Anwendung auf quadratische Probleme endlich sein.

**Satz:** Es sei  $f(x) := \frac{1}{2}x^T A x - b^T x + c, A$  positiv definit, und die Suchrichtungen  $d_0, \dots, d_{n-1}$  seien bzgl.  $A$  konjugiert. Bei exakter Schrittweitenbestimmung gilt dann

$$x_n = A^{-1}b ,$$

d.h.  $x_n$  ist Optimallösung des Problems.

Das wohl einfachste und bekannteste Verfahren zur Erzeugung konjugierter Richtungen stammt von Fletcher und Reeves. Für die Anwendung auf das allgemeine unrestringierte Problem hat es folgende Form:

**Algorithmus:** Setze  $d_0 := -\nabla f(x_0)$  für einen beliebigen Startpunkt  $x_0$  und für  $k = 1, 2, \dots$  bestimme

$$d_k := -\nabla f(x_k) + \gamma_k d_{k-1}$$

mit  $\gamma_k := (\|\nabla f(x_k)\| / \|\nabla f(x_{k-1})\|)^2$ .

**Satz:** Es sei  $f(x) := \frac{1}{2}x^T A x - b^T x + c, A$  positiv definit und  $\alpha_k$  durch exakte Schrittweitenbestimmung definiert. Dann sind die Suchrichtungen des Fletcher-Reeves-Verfahrens bzgl.  $A$  konjugiert, und es gibt ein  $l \leq n$  mit

$$x_l = A^{-1}b .$$

Der Vorteil des Verfahrens liegt darin, daß zur Berechnung der Suchrichtung keine Matrizen gespeichert werden müssen. Damit kann es zur Lösung großer Optimierungsprobleme herangezogen werden. Für kleine Probleme scheint es sensitiv gegenüber der Genauigkeit bei der Schrittweitenbestimmung zu sein und ist den nachfolgenden Verfahren numerisch unterlegen.

## 2.3 Newton-Verfahren

Quadratische Probleme mit  $f(x) := \frac{1}{2}x^T Ax - b^T x + c$  lassen sich in einem Iterationsschritt lösen, setzt man

$$x^* := A^{-1}b = x_0 - A^{-1}(Ax_0 - b)$$

für ein beliebiges  $x_0 \in \mathbb{R}^n$ . Dabei soll  $A$  positiv definit sein. Da  $A = \nabla^2 f(x_0)$  und  $Ax_0 - b = \nabla f(x_0)$ , führt diese Überlegung zur Formulierung des Newton-Verfahrens, das auch als eine Verallgemeinerung des bekannten eindimensionalen Verfahrens zur Nullstellenbestimmung anzusehen ist. Hierbei wird die Suchrichtung durch Invertierung der Hesse-Matrix gewonnen:

$$d_k := -\nabla^2 f(x_k)^{-1} \nabla f(x_k) .$$

Der Vorteil des Verfahrens liegt im quadratischen Konvergenzverhalten und im Verzicht auf eine Schrittweitenbestimmung in der Nähe einer Optimallösung begründet.

**Satz:** *Es sei  $x^*$  optimal für das unrestringierte Optimierungsproblem, d.h.  $\nabla f(x^*) = 0$  und  $\nabla^2 f(x^*)$  positiv definit. Falls  $x_0$  hinreichend nahe bei  $x^*$  ist, konvergiert das Newton-Verfahren*

$$x_{k+1} = x_k - \nabla^2 f(x_k)^{-1} \nabla f(x_k)$$

*quadratisch gegen  $x^*$ , d.h. es gilt*

$$\|x_{k+1} - x^*\| < \gamma \|x_k - x^*\|^2 .$$

Dieses Konvergenzresultat wird jedoch durch einige Nachteile des Newton-Verfahrens beeinträchtigt:

- a) Für jeden Iterationsschritt müssen zweite Ableitungen berechnet werden.
- b) Die Lösung des Gleichungssystems  $\nabla^2 f(x_k)d_k = -\nabla f(x_k)$  ist aufwendig.
- c) Die Hesse-Matrix kann nichtsingulär oder schlecht konditioniert werden, so daß zusätzliche Sicherungsvorkehrungen erforderlich sind.
- d) Das Verfahren konvergiert nur lokal.
- e) Bei ungünstigen Startwerten ist u.U.  $\nabla^2 f(x_k)$  nicht mehr positiv definit, so daß  $d_k$  keine Abstiegsrichtung mehr ist.

## 2.4 Quasi-Newton-Verfahren

Für Quasi-Newton-Verfahren soll auf die Berechnung zweiter Ableitungen verzichtet werden. Die Suchrichtung wird durch Multiplikation des Gradienten mit einer sogenannten Quasi-Newton-Matrix  $H_k$  berechnet, d.h.

$$d_k := -H_k \nabla f(x_k) ,$$

wobei  $H_k$  eine geeignete Näherung von  $\nabla^2 f(x_k)^{-1}$  darstellen soll. Folgende Forderungen werden deshalb an  $H_k$  gestellt:

### Bedingungen:

- a)  $H_k$  ist positiv definit, sofern  $H_0$  positiv definit ist.
- b)  $H_k$  erfüllt die Quasi-Newton-Bedingung

$$H_{k+1}(\nabla f(x_{k+1}) - \nabla f(x_k)) = x_{k+1} - x_k .$$

- c)  $H_{k+1}$  entsteht aus den Daten  $H_k$ ,  $x_{k+1}$ ,  $x_k$ ,  $\nabla f(x_{k+1})$  und  $\nabla f(x_k)$  durch eine Rang-2-Korrektur, d.h.

$$H_{k+1} = H_k + \beta_k u_k u_k^T + \gamma_k v_k v_k^T$$

mit  $\beta_k, \gamma_k \in \mathbb{R}$ ,  $u_k, v_k \in \mathbb{R}^n$ .

- d) Falls  $f(x) = \frac{1}{2}x^T A x + b^T x$ ,  $A$  positiv definit,  $\alpha_k$  exakte Schrittweite und  $m = 0$ , sind die Suchrichtungen  $d_0, d_1, \dots$  konjugiert, d.h.  $d_i^T A d_j = 0$ ,  $0 \leq i < j$ .

Da die Hesse-Matrix von  $f$  in einem Minimum positiv definit ist und  $H_k$  ihr Inverses approximieren soll, wird die positive Definitheit von  $H_k$  verlangt. Zusätzlich erzwingt diese Bedingung, daß es längs der Suchrichtung  $d_k$  *bergab* geht, d.h. daß

$$d_k^T \nabla f(x_k) = -\nabla f(x_k)^T H_k \nabla f(x_k) < 0 .$$

Zur Motivation der Quasi-Newton-Bedingung betrachte man eine quadratische Funktion  $f(x) := \frac{1}{2}x^T A x - b^T x + c$  mit einer positiv definiten Matrix  $A$ . Dann gilt für  $H_k := A^{-1}$

$$\begin{aligned} H_{k+1}(\nabla f(x_{k+1}) - \nabla f(x_k)) &= A^{-1}(A x_{k+1} - b - A x_k + b) \\ &= x_{k+1} - x_k . \end{aligned}$$

Bedingung c) bewirkt eine einfache Berechnungsformel für die Matrizen  $H_k$ , während d) die Endlichkeit des Verfahrens nach höchstens  $n$  Iterationen für quadratische Probleme garantiert.

Unter Verzicht auf eine allgemeine Darstellung von Verfahren, die obige Bedingungen erfüllen, sollen hier nur die zwei bekanntesten und gebräuchlichsten Algorithmen angegeben werden.

**Definition:** Sei  $q_k := \nabla f(x_{k+1}) - \nabla f(x_k)$ ,  $p_k := x_{k+1} - x_k$ . Dann heißen die Rekursionsformeln

a)

$$H_{k+1} := H_k + \frac{p_k p_k^T}{p_k^T q_k} - \frac{H_k q_k q_k^T H_k}{q_k^T H_k q_k}$$

das Verfahren von Davidon-Fletcher-Powell (DFP-Verfahren) und

b)

$$H_{k+1} := H_k + \left(1 + \frac{q_k^T H_k q_k}{p_k^T q_k}\right) \frac{p_k p_k^T}{p_k^T q_k} - \frac{1}{p_k^T q_k} \left(p_k q_k^T H_k + H_k q_k p_k^T\right)$$

das Verfahren von Broyden-Fletcher-Goldfarb-Shanno (BFGS-Verfahren).

**Satz:** Es sei  $H_0$  positiv definit,  $x_0$  beliebiger Startpunkt und  $\{x_k\}$  eine Iterationsfolge des DFP- bzw. BFGS-Quasi-Newton-Verfahrens mit  $p_k^T q_k > 0$ . Dann ist  $H_k$  positiv definit für alle  $k$ , und es gilt die Quasi-Newton-Gleichung

$$H_{k+1}(\nabla f(x_{k+1}) - \nabla f(x_k)) = x_{k+1} - x_k .$$

Für quadratische Zielfunktionen können wesentlich weitergehende Eigenschaften der Quasi-Newton-Verfahren gezeigt werden.

**Satz:** Es sei  $f(x) := \frac{1}{2}x^T A x - b^T x + c$ ,  $A$  positiv definit,  $x_0 \in \mathbb{R}^n$ ,  $H_0$  positiv definit und  $\{x_k\}$  eine Iterationsfolge des DFP- oder BFGS-Verfahrens mit exakter Schrittweitenbestimmung. Dann gibt es ein  $l \leq n$ , so daß:

- a) Die Suchrichtungen  $d_0, \dots, d_{l-1}$  sind bzgl.  $A$  konjugiert.
- b)  $x_l = A^{-1}b$  ist die Optimallösung.
- c) Falls  $l = n$ , gilt  $H_n = A^{-1}$ .

Für allgemeine nichtquadratische Zielfunktionen können nur lokale Konvergenzaussagen gezeigt werden:

**Satz:** Es sei  $x^*$  lokales Minimum von  $f$ ,  $\nabla^2 f(x)$  positiv definit in  $x^*$  und Lipschitz-stetig, d.h.

$$\|\nabla^2 f(x) - \nabla^2 f(x^*)\| \leq \lambda \|x - x^*\|$$

für alle  $x$  aus einer Umgebung von  $x^*$ . Dann konvergiert das Quasi-Newton-Verfahren mit exakter Schrittweitenbestimmung lokal superlinear gegen  $x^*$ , d.h. falls  $H_0$  positiv definit ist und  $x_0$  hinreichend nahe bei  $x^*$  gewählt wird, gilt

$$\lim_{k \rightarrow \infty} \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|} = 0 .$$

Dieser Satz gilt auch für etwas verallgemeinerte Schrittlängenverfahren. Die superlineare Konvergenz besagt, daß es eine Nullfolge  $\{\gamma_k\}$  gibt mit

$$\|x_{k+1} - x^*\| \leq \gamma_k \|x_k - x^*\| .$$

Die bisher erwähnten Eigenschaften der Quasi-Newton-Verfahren gelten in der Regel nur für exakte Schrittweitenbestimmungen. Theoretisch erzeugen in diesem Fall die beiden hier besprochenen Verfahren sogar identische Iterationsfolgen. In der Praxis jedoch kann eine exakte Schrittweitenbestimmung nicht durchgeführt werden, so daß alle bisher erwähnten Eigenschaften nur mit Einschränkungen gültig bleiben. Insbesondere besteht die Gefahr, daß die Matrizen  $H_k$  nicht mehr positiv definit bleiben, und daß damit die Suchrichtungen nicht mehr profitabel sind. Um dieser Gefahr zu begegnen, wird in heutigen Implementierungen häufig die inverse Matrix von  $H_k$  iteriert. Setzt man  $B_k := H_k^{-1}$  für alle  $k$ , erhält man ebenfalls eine Rang-2-Korrekturformel für  $B_k$ :

**Satz:** *Für das inverse BFGS-Verfahren*

$$B_{k+1} := B_k + \frac{q_k q_k^T}{p_k^T q_k} - \frac{B_k p_k p_k^T B_k}{p_k^T B_k p_k}$$

*gilt  $B_k = H_k^{-1}$  für alle  $k$ .*

Da die obige Rekursionsformel eng mit der des DFP-Verfahrens für die Matrix  $H_k$  verwandt ist, bezeichnet man auch beide Verfahren als dual oder komplementär zueinander. Da nun die Lösung eines Gleichungssystems

$$B_k d_k = -\nabla f(x_k)$$

in jedem Iterationsschritt erforderlich ist, speichert man  $B_k$  in LDL-Faktorisierung (Choleskey-Verfahren), d.h.

$$B_k = L_k D_k L_k^T$$

mit einer unteren Dreiecksmatrix  $L_k$  und einer Diagonalmatrix  $D_k$ . Allerdings wird nicht mehr diese Choleskey-Zerlegung in jedem Iterationsschritt wiederholt, sondern man bringt das inverse BFGS-Verfahren in die Form

$$L_{k+1} D_{k+1} L_{k+1}^T = L_k D_k L_k^T + u_k u_k^T - v_k v_k^T$$

und modifiziert statt  $B_k$  die Zerlegungsmatrizen  $L_k$  und  $D_k$  nach einem Verfahren von Gill, Murray, Saunders (1975), siehe auch Kraft (1977). Dieser Algorithmus erfordert ungefähr denselben Rechenaufwand wie die ursprüngliche Rekursionsformel und garantiert die positive Definitheit von  $H_k$  bzw.  $B_k$  für alle  $k$ , da  $q_k^T p_k > 0$  durch eine einfache Modifizierung der Korrekturformel garantiert werden kann.

## 3. Schrittlängenbestimmung

### 3.1 Problemstellung

Die Standardalgorithmen zur nichtlinearen Programmierung bestimmen einen neuen Iterationswert  $\bar{x}$  in der Form

$$\bar{x} := x + \bar{\alpha}d ,$$

wobei  $x$  die alte Näherungslösung,  $d$  die sogenannte Suchrichtung und  $\bar{\alpha}$  eine reelle Zahl ist, die eine Funktion  $F$  längs  $d$  minimieren soll, d.h. für die

$$F(\bar{\alpha}) = \min\{F(x + \alpha d) : 0 < \alpha < \infty\}$$

gilt. Die spezielle Struktur von  $F$  hängt vom jeweiligem Optimierungsverfahren ab und wird später untersucht. Der Einfachheit halber kann man sich hier ein unrestringiertes Optimierungsproblem vorstellen, d.h.

$$\min_{x \in \mathbb{R}^n} F(x) .$$

Die Problematik bei der Ermittlung einer Schrittweite  $\alpha$  liegt nun darin, mit möglichst geringem Aufwand, d.h. mit möglichst wenigen Funktionsaufrufen von  $F$  eine hinreichend genaue Näherungslösung zu bestimmen, so daß ein Verfahren zur Minimierung von  $F$  noch konvergieren würde.

## 3.2 Schrittweiten mit garantierter Konvergenz

Schon frühzeitig wurde die exakte Bestimmung von  $\bar{\alpha}$  durch Bedingungen ersetzt, die die Konvergenz des zugrunde gelegten Algorithmus zur unrestringierten Minimierung einer Funktion  $F$  zum Ziel hatten. Zwei der bekanntesten Bedingungen dieser Art sind die folgenden:

**Goldstein-Bedingung:** Es sei  $\bar{\alpha}$  so gewählt, daß

$$0 < -\mu_1 \bar{\alpha} \nabla F(x)^T d \leq F(x) - F(x + \bar{\alpha}d) \leq -\mu_2 \bar{\alpha} \nabla F(x)^T d$$

gilt, wobei  $0 < \mu_1 \leq \frac{1}{2} \leq \mu_2 < 1$  zu setzen ist.

**Armijo-Bedingung:** Es sei  $0 < \mu < 0.5$ ,  $q > 1$ ,  $1 \geq \sigma > 0$  und  $\bar{\alpha}$  die größte Zahl in der Folge  $\{\sigma q^{-j}\}$ , so daß

$$F(x) - F(x + \bar{\alpha}d) \geq -\mu \bar{\alpha} \nabla F(x)^T d .$$

Zur Veranschaulichung dieser Schrittweitenverfahren betrachte man die Funktion

$$\Psi(\alpha) := F(x + \alpha d) .$$

Wegen  $\Psi'(\alpha) = \nabla F(x + \alpha d)^T d$  lassen sich beide Bedingungen in eine einfachere Form bringen.

$$\text{Goldstein: } \Psi(0) + \bar{\alpha} \mu_2 \Psi'(0) \leq \Psi(\bar{\alpha}) \leq \Psi(0) + \bar{\alpha} \mu_1 \Psi'(0)$$

$$\text{Armijo: } \Psi(2^{-j}) \leq \Psi(0) + 2^{-j} \mu \Psi'(0) \quad (\sigma := 1, q := 2)$$

Beide Bedingungen besagen, daß zwar auf der einen Seite die Abnahme von  $F$  groß genug sein muß, damit das Verfahren möglichst rasch konvergiert, jedoch auf der anderen Seite nicht zu klein werden darf, um die Konvergenz gegen einen stationären Punkt nicht zu unterbinden. Ein typisches Konvergenzresultat zur Minimierung einer differenzierbaren Funktion  $F$  ohne Restriktionen ist im folgenden Satz beschrieben.

**Satz:** *Es sei  $\{x_k\}$  eine Iterationsfolge gebildet nach der Vorschrift*

$$x_{k+1} := x_k + \alpha_k d_k ,$$

*wobei  $\alpha_k$  nach der Goldstein- oder Armijo-Bedingung bestimmt wurde, und wobei für die Suchrichtung*

$$-\nabla F(x_k)^T d_k > \sigma \|d_k\| \|\nabla F(x_k)\|$$

*gelten möge. Falls für den Startpunkt  $x_0$  die Niveaumenge*

$$K := \{x \in \mathbb{R}^n : F(x) \leq F(x_0)\}$$

*kompakt ist, ist jeder Häufungspunkt  $x^*$  von  $\{x_k\}$  stationär, d.h. es gilt*

$$\nabla F(x^*) = 0 .$$

Hierbei ist zu beachten, daß die Suchrichtung  $d_k$  in jedem Iterationsschritt beliebig gewählt werden darf, sofern es sich nur um eine Abstiegsrichtung handelt und sie nicht orthogonal zum Gradienten von  $F$  steht.

Numerisch ist ein Verfahren mit den Schrittweitenbestimmungen nach Goldstein oder Armijo ineffizient, so daß es in der Praxis nicht verwandt wird. Allerdings werden wegen der globalen Konvergenzaussage Bedingungen dieser Art häufig als Stoppkriterien in andere Schrittweitenverfahren eingebaut.

### 3.3 Numerische Algorithmen

Die Implementierung eines Schrittweitenalgorithmus ist ein kritischer Punkt in vielen Optimierungsverfahren und beeinflusst die Effizienz erheblich. In der Regel basiert die Schrittweitenbestimmung auf quadratischer Interpolation. Falls man jedoch numerische Schwierigkeiten zu erwarten hat, kombiniert man ein solches Verfahren häufig mit einer Intervallmethode. Als Stoppkriterium kann man die Goldstein-Bedingung oder ähnliche benutzen. Ist jedoch ein pathologisches Kurvenverhalten auszuschließen, genügt in der Regel ein einfaches Interpolationsverfahren. Bei der Wahl eines Schrittweitenalgorithmus ist also auf die Struktur der zu lösenden Probleme und die des übergeordneten Optimierungsverfahrens Rücksicht zu nehmen.

Gerade Algorithmen zur restringierten Optimierung erzwingen häufig einen pathologischen Kurvenverlauf von  $\rho(\alpha)$  am Rand des zulässigen Bereichs. Dies kann zu ineffizienten Interpolationsverfahren führen. Je nach Lage von  $\alpha_1, \alpha_2, \alpha_3$  liegt  $\bar{\alpha}$  entweder zu weit vom tatsächlichen Minimum entfernt oder zu nahe an der ersten Stützstelle.

Für die später zu besprechenden SQP-Verfahren genügt in der Regel eine grobe Schrittweitenbestimmung, da nach Möglichkeit die Schrittweite 1 gewählt werden soll. Hierzu eignet sich folgendes einfaches Verfahren:

#### Algorithmus:

Gegeben seien  $\beta, \mu$  mit  $0 < \beta < 1, 0 < \mu < \frac{1}{2}$  ( $\beta = 0.1, \mu = 0.01$ )

Start:  $\alpha_0 := 1$

Für  $i = 0, 1, 2, \dots$

- 1) Falls  $\Psi(\alpha_i) < \Psi(0) + \mu\alpha_i\Psi'(0)$ , stop
- 2)  $\bar{\alpha}_i := \frac{1}{2}\alpha_i^2\Psi'(0)/(\alpha_i\Psi'(0) - (\Psi(\alpha_i) - \Psi(0)))$
- 3)  $\alpha_{i+1} := \max(\beta\alpha_i, \bar{\alpha}_i)$

Hier wird eine quadratische Interpolation basierend auf  $\Psi(0), \Psi'(0)$  und einer Zwischenstelle  $\Psi(\alpha_i)$  verbunden mit einer Armijo-Schrittweitenreduzierung. Damit werden auch die Spezialfälle abgefangen, die bei einer ungeeigneten quadratischen Interpolation auftreten.

**Beispiel:** Es seien  $f(x) = x_1^2 + x_2, x_k := \begin{pmatrix} 2 \\ 0 \end{pmatrix}, d_k := \begin{pmatrix} -4 \\ -1 \end{pmatrix}$  und die Konstanten  $\mu := 1/4, \beta := 1/2$  gegeben. Dann ist  $\Psi(\alpha) = F(x_k + \alpha d_k) = 16\alpha^2 - 17\alpha + 4$ , d.h.  $\Psi(0) = 4, \Psi'(0) = -17$ . Setze  $\alpha_0 := 1$ .

$$i = 0 : \Psi(\alpha_0) = 3 > \Psi(0) + \frac{1}{4}\Psi'(0) = 4 - \frac{17}{4}$$

$$\bar{\alpha}_0 = \frac{17}{32} \text{ (exaktes Minimum von } \Psi(\alpha) \text{ längs } d_k)$$

$$\alpha_1 = \max\left(\frac{1}{2}, \frac{17}{32}\right) = \frac{17}{32}$$

$$i = 1 : \Psi(\alpha_1) \sim -0.1 < \Psi(0) + \frac{1}{4}\alpha_1\Psi'(0) \sim 0.9$$

$\alpha_k = \frac{17}{32}$  ist die gesuchte Schrittweite.

## 4. Numerische Verfahren

### 4.1 Voraussetzungen

Wir wenden uns der Lösung von restringierten Optimierungsproblemen der Form

$$\begin{aligned} \min f(x) \\ x \in \mathbb{R}^n : \quad g_j(x) = 0 \quad , \quad j = 1, \dots, m_e \\ g_j(x) \geq 0 \quad , \quad j = m_e + 1, \dots, m \end{aligned}$$

zu. Zur Vereinfachung der Beschreibung des Verfahrens werden obere und untere Schranken für die Variablen ausgelassen. Folgende Voraussetzungen seien getroffen:

- Das Problem ist glatt, d.h. die Problemfunktionen  $f$  und  $g_1, \dots, g_m$  sind auf dem  $\mathbb{R}^n$  stetig differenzierbar.
- Das Problem ist klein, d.h. die Anzahl der Variablen und Nebenbedingungen ist nicht so groß, daß Spezialverfahren herangezogen werden müssen (z.B.  $n, m < 100 - 200$ ).
- Das Problem ist gut skaliert, d.h. Änderungen in den Variablen erzeugen Änderungen in den Problemfunktionen in derselben Größenordnung.
- Das Problem ist wohldefiniert, d.h. der zulässige Bereich ist nicht leer, und es existiert eine Optimallösung.

Obwohl in der Vergangenheit eine Vielzahl mathematischer Optimierungsverfahren entwickelt und auch implementiert wurden (z.B. Penalty-, Multiplikator-, verallgemeinerte reduzierte Gradientenverfahren), beschäftigen wir uns jetzt nur mit sequentiellen quadratischen Programmierungsverfahren (SQP-Verfahren) auf Grund ihrer Effizienz und Zuverlässigkeit insbesondere für allgemeine Probleme, für die keine Spezialverfahren entwickelt werden können.

## 4.2 Das quadratische Teilproblem

SQP-Verfahren basieren auf der sukzessiven Lösung quadratischer Teilprobleme, die durch eine quadratische Approximation der Lagrange-Funktion und eine Linearisierung der Restriktionen entstehen. Sei  $x_k \in \mathbb{R}^n$  eine Näherung der Optimallösung,  $v_k \in \mathbb{R}^m$  eine Näherung der Multiplikatoren und  $B_k$  eine Näherung der Hessematrix der Lagrange-Funktion im Optimalpunkt. Dann erhält man eine Suchrichtung durch Lösen des folgenden Teilproblems:

$$\begin{aligned} \min \quad & \frac{1}{2}d^T B_k d + \nabla f(x_k)^T d \\ d \in \mathbb{R}^n : \quad & \nabla g_j(x_k)^T d + g_j(x_k) = 0 \quad , \quad j = 1, \dots, m_e \\ & \nabla g_j(x_k)^T d + g_j(x_k) \geq 0 \quad , \quad j = m_e + 1, \dots, m \end{aligned}$$

Das obige Teilproblem, daß in jedem Iterationsschritt des SQP-Verfahrens gelöst werden muß, werde jetzt mit (QP) bezeichnet. Falls  $d_k$  die Optimallösung und  $u_k$  der zugehörige Multiplikator von (QP) ist, erhält man die neuen Iterationswerte

$$\begin{pmatrix} x_{k+1} \\ v_{k+1} \end{pmatrix} = \begin{pmatrix} x_k \\ v_k \end{pmatrix} + \alpha_k \begin{pmatrix} d_k \\ u_k - v_k \end{pmatrix},$$

wobei  $\alpha_k$  eine geeignete Schrittlänge darstellt. Man beachte, daß  $v_{k+1} = u_k$  im Falle  $\alpha_k = 1$  gilt.

*Motivation:* Sei  $m_e = m$ , d.h. es mögen keine Ungleichungen existieren. Ziel ist die Lösung der Kuhn-Tucker-Bedingungen

$$\begin{pmatrix} \nabla_x L(x, v) \\ g(x) \end{pmatrix} = 0$$

mit  $g = (g_1, \dots, g_m)^T$ , d.h. die Lösung eines Gleichungssystems  $F(z) = 0$  mit  $n + m$  Unbekannten  $z = (x, v)$  und  $n + m$  nichtlinearen Funktionen  $F(z) = (\nabla_x L(x, v), g(x))$ . Gegeben sei nun eine Näherungslösung  $z_k = (x_k, v_k)$ . Mit  $B_k := \nabla_{xx} L(x_k, v_k)$  erhält man aus dem Newton-Verfahren

$$\nabla F(z_k) p_k + F(z_k) = 0$$

die Bedingung

$$\begin{pmatrix} B_k & : & -\nabla g(x_k) \\ \nabla g(x_k)^T & : & 0 \end{pmatrix} \begin{pmatrix} d_k \\ y_k \end{pmatrix} + \begin{pmatrix} \nabla f(x_k) - \nabla g(x_k) v_k \\ g(x_k) \end{pmatrix} = 0$$

Hierbei ist  $p_k = (d_k, y_k)$ . Dann ist mit  $u_k := y_k - v_k$

$$B_k d_k - \nabla g(x_k) u_k + \nabla f(x_k) = 0$$

$$\nabla g(x_k)^T d_k + g(x_k) = 0$$

Dies sind die Optimalitätsbedingungen für das quadratische Teilproblem (QP), d.h. durch (QP) wird das Newton-Verfahren zur Lösung der notwendigen Optimalitätsbedingungen von (NLP) definiert, sofern  $B_k$  die Hessematrix der Lagrange-Funktion ist.

**Beispiel:** Gegeben sei das Problem

$$\begin{aligned} \min \quad & x_1^2 + x_2 \\ x_1, x_2 : \quad & 9 - x_1^2 - x_2^2 \geq 0 \\ & 1 - x_1 - x_2 \geq 0 \end{aligned}$$

Gegeben seien die Startwerte  $x_0 = (2, 0)^T$  und  $B_0 = I$ , d.h.  $B_0$  sei die Einheitsmatrix. Dann lautet das quadratische Teilproblem

$$\begin{aligned} \min \quad & \frac{1}{2}d_1^2 + \frac{1}{2}d_2^2 + 4d_1 + d_2 \\ d_1, d_2 : \quad & -4d_1 + 5 \geq 0 \\ & -d_1 - d_2 - 1 \geq 0 \end{aligned}$$

Da die linearen Restriktionen nicht aktiv werden, erhält man durch einfaches Nachrechnen die Lösung  $d_0 = (-4, -1)^T$ ,  $u_0 = (0, 0)^T$ .

Die neue Matrix  $B_1$ , die die Hesse-Matrix der Lagrange-Funktion approximieren soll, wird mit Hilfe des BFGS-Verfahrens bestimmt:

$$\begin{aligned} B_1 &= B_0 + \frac{q_0 q_0^T}{p_0^T q_0} - \frac{B_0 p_0 p_0^T B_0}{p_0^T B_0 p_0} \\ q_0 &= \nabla_x L(x_1, u_0) - \nabla_x L(x_0, u_0) = \nabla f(x_1) - \nabla f(x_0) \\ &= \begin{pmatrix} -4 \\ 1 \end{pmatrix} - \begin{pmatrix} 4 \\ 1 \end{pmatrix} = \begin{pmatrix} -8 \\ 0 \end{pmatrix}, \\ p_0 &= x_1 - x_0 = \begin{pmatrix} -4 \\ -1 \end{pmatrix} \end{aligned}$$

Damit ist

$$B_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} + \frac{1}{32} \begin{pmatrix} 64 & 0 \\ 0 & 0 \end{pmatrix} - \frac{1}{17} \begin{pmatrix} 16 & 4 \\ 4 & 1 \end{pmatrix} \approx \begin{pmatrix} 2 & -1/4 \\ -1/4 & 1 \end{pmatrix}.$$

Das neue quadratische Teilproblem ergibt sich zu

$$\begin{aligned} \min \quad & d_1^2 + \frac{1}{2}d_2^2 - \frac{1}{4}d_1 d_2 - 4d_1 + d_2 \\ d_1, d_2 : \quad & 4d_1 + 2d_2 + 4 \geq 0 \\ & -d_1 - d_2 + 4 \geq 0 \end{aligned}$$

Da das unrestringierte Minimum von (QP) zulässig ist, ist

$$d_1 = \frac{1}{21} \begin{pmatrix} 20 \\ -16 \end{pmatrix}, \quad u_1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

ein Kuhn-Tucker-Punkt und damit Optimallösung von (QP).

Der tatsächlich berechnete Iterationsverlauf wurde mit Hilfe des SQP-Programms NLPQL nachgerechnet und ist in der nachfolgenden Tabelle aufgelistet.

$k$	$x_1^{(k)}$	$x_2^{(k)}$	$f(x_k)$	$r(x_k)$	$s(x_k)$
0	2.0000000	0.0000000	4.0000000	1.0	17.
1	-2.0000000	-1.0000000	3.0000008	.0	6.56
2	-1.0000000	-1.81250000	-.81249925	.0	33.59
3	-.1370218	-2.2942287	-2.2754537	.0	.69
4	.2791080	-2.8742998	-2.7963985	.0	.27
5	.2301523	-3.0153240	-2.9623539	.15	.36
6	.21 <sub>10</sub> - 1	-3.0237105	-3.0232876	.14	.47 <sub>10</sub> - 1
7	.99 <sub>10</sub> - 3	-3.0001562	-3.0001552	.94 <sub>10</sub> - 3	.31 <sub>10</sub> - 3
8	.21 <sub>10</sub> - 3	-3.0000001	-3.0000001	.63 <sub>10</sub> - 6	.11 <sub>10</sub> - 6
9	.15 <sub>10</sub> - 5	.3.0000000	-3.0000000	.42 <sub>10</sub> - 7	.14 <sub>10</sub> - 7

Hierbei ist  $r(x_k)$  die Summe der Restriktionsverletzungen, und  $s(x_k)$  dient als Stoppkriterium:

$$s(x_k) := | \nabla f(x_k)^T d_k | + \sum_{i=1}^m | u_i^{(k)} g_i(x^{(k)}) |$$

Aus der Formulierung des quadratischen Teilproblems folgt der folgende Satz:

**Satz:** *Es sei  $d_k = 0$  Optimallösung des quadratischen Teilproblems (QP) und  $u_k$  der zugehörige Multiplikator. Dann erfüllen  $x_k$  und  $u_k$  die notwendigen Optimalitätsbedingungen des nichtlinearen Programms (NLP).*

*Beweis:* Die Optimalitätsbedingungen für das quadratische Teilproblem (QP) lauten:

$$B_k d_k + \nabla f(x_k) - \sum_{j=1}^m u_j^{(k)} \nabla g_j(x_k) = 0$$

$$\nabla g_j(x_k)^T d_k + g_j(x_k) = 0, \quad j = 1, \dots, m_e,$$

$$\nabla g_j(x_k)^T d_k + g_j(x_k) \geq 0, \quad j = m_e + 1, \dots, m,$$

$$u_k \geq 0,$$

$$\sum_{j=1}^m u_j^{(k)} (\nabla g_j(x_k)^T d_k + g_j(x_k)) = 0.$$

Für  $d_k = 0$  folgt die Behauptung.

Numerische Schwierigkeiten können jedoch durch die Unlösbarkeit von (QP) entstehen, obwohl das Ausgangsproblem wohldefiniert ist.

**Beispiel:** Es sei das folgende Problem gegeben:

$$\begin{aligned}
& \min (x_1 - 2)^2 + (x_2 - 3)^2 \\
& x_1 + x_2^2 \geq 0 \\
x_1, x_2 : & x_1^2 + x_2 \geq 0 \\
& -0.5 \leq x_1 \leq 0.5 \\
& x_2 \leq 1
\end{aligned}$$

Für  $x_0 = (-2, 1)$ ,  $B_0 = I$  erhält man ein unlösbares Teilproblem:

$$\begin{aligned}
& \min \frac{1}{2}d_1^2 + \frac{1}{2}d_2^2 - 8d_1 - 4d_2 \\
& d_1 + 2d_2 - 1 \geq 0 \\
d_1, d_2 : & -4d_1 + d_2 + 5 \geq 0 \\
& 1.5 \leq d_1 \leq 2.5 \\
& d_2 \leq 0
\end{aligned}$$

Ein weiterer Nachteil ist die Berechnung aller Restriktionsgradienten für das Teilproblem, wodurch Rechenzeit für nichtaktive Nebenbedingungen vergeudet wird. Beide Nachteile lassen sich z.B. durch folgendes modifizierte Teilproblem beheben, bei dem eine zusätzliche Variable  $\delta$  eingeführt wird.

$$\begin{aligned}
& \min \frac{1}{2}d^T B_k d + \nabla f(x_k)^T d + \frac{1}{2}\sigma_k^2 \delta^2 \\
d \in \mathbb{R}^n, \delta \in \mathbb{R} : & \nabla g_j(x_k)^T d + (1 - \delta)g_j(x_k) \begin{cases} = \\ \geq \end{cases} 0, j \in J_k, \\
& \nabla g_j(x_{j(k)})^T d + g_j(x_k) \geq 0, j \in K_k
\end{aligned}$$

Hierbei ist  $\sigma_k$  ein geeigneter Penaltyparameter, siehe Schittkowski (1983), und

$$J_k := \{1, \dots, m_e\} \cup \{j : m_e < j \leq m, g_j(x_k) < \epsilon \text{ oder } v_j^{(k)} > 0\}$$

$$K_k := \{1, \dots, m\} \setminus J_k$$

Hierbei ist  $\epsilon$  eine vorgegebene Fehlertoleranz,  $v_j^{(k)}$  der j-te Koeffizient von  $v_k$ , und  $k(j)$  kennzeichnet 'alte' Gradienten. Da  $d = 0$ ,  $\delta = 1$  stets zulässig ist, ist das modifizierte Teilproblem stets lösbar. Wir wollen es im Folgenden mit (QPM) bezeichnen. Man beachte, daß u.U.  $d = 0$ ,  $\delta = 1$  die einzige zulässige Lösung von (QPM) darstellt, wenn die *constraint qualification* nicht erfüllt ist.

### 4.3 Die Meritfunktion

Zur Bestimmung einer Schrittweite  $\alpha_k$  mit

$$\begin{pmatrix} x_{k+1} \\ v_{k+1} \end{pmatrix} = \begin{pmatrix} x_k \\ v_k \end{pmatrix} + \alpha_k \begin{pmatrix} d_k \\ u_k - v_k \end{pmatrix}$$

benötigt man eine geeignete eindimensionale Meritfunktion

$$\Psi_k(\alpha) := \Phi_{r_k} \left( \begin{pmatrix} x_k \\ v_k \end{pmatrix} + \alpha \begin{pmatrix} d_k \\ u_k - v_k \end{pmatrix} \right)$$

wobei  $\Phi_r(x, v)$  von  $n + m$  Variablen  $x$  und  $v$  abhängt. Der Index  $r$  bezeichnet einen Vektor von Penaltyparametern.

a) *Exakte  $L_1$ -Penaltyfunktion:*

$$\Phi_r(x, v) := f(x) + \sum_{j=1}^{m_e} r_j |g_j(x)| + \sum_{j=m_e+1}^m r_j |\min(0, g_j(x))|$$

Obige Funktion 'bestraft'  $f(x)$  im Sinne der  $L_1$ -Norm, sobald der zulässige Bereich verlassen wird, und hängt nicht von  $v$  ab.  $\Phi$  heißt exakt, da für hinreichend großes  $r$  (größer als optimaler Lagrange-Multiplikator) ein unrestringiertes Minimum von  $\Phi$  eine Lösung des restringierten Problems (NLP) darstellt. Die Funktion ist allerdings nicht differenzierbar und die untere Schranke für  $r$  im voraus nicht bekannt.

**Beispiel:** Sei

$$\min \{2x : x \geq 0\}$$

Dann ist

$$\Phi_r(x, v) = 2x + r |\min(0, x)|$$

die  $L_1$ -Penaltyfunktion. Die Optimallösung ist

$$x^* = 0, \quad u^* = 2$$

Im SQP-Verfahren werden die Penalty-Parameter wie folgt bestimmt, vgl. Powell (1978a):

$$r_j^{(k)} := \max(|u_j^{(k)}|, \frac{1}{2}(r_j^{(k-1)} + |u_j^{(k)}|))$$

Allerdings konvergiert das resultierende Verfahren ohne zusätzliche Sicherungsvorkehrungen i.a. nicht (Gegenbeispiel), und lokal ist  $\alpha_k = 1$  nicht mehr gewährleistet, was die schnelle superlineare Konvergenzrate verhindern kann. Obwohl die numerische Implementierung eines SQP-Verfahrens mit der  $L_1$ -Penaltyfunktion als zuverlässig und robust angesehen werden kann, betrachten wir eine andere Meritfunktion, um die genannten Nachteile zu vermeiden.

b) *Erweiterte Lagrange-Funktion:*

$$\Phi_r(x, v) := f(x) - \sum_{j \in J} (v_j g_j(x) - \frac{1}{2} r_j g_j(x)^2) - \frac{1}{2} \sum_{j \in K} v_j^2 / r_j,$$

$$J := \{1, \dots, m_e\} \cup \{j : m_e < j \leq m, g_j(x) \leq v_j / r_j\},$$

$$K := \{1, \dots, m\} \setminus J.$$

Jetzt wird die Lagrange-Funktion mit Hilfe der euklidischen  $L_2$ -Norm 'bestraft', sobald der zulässige Bereich verlassen wird. Die Funktion  $\Phi_r(x, v)$  ist jetzt bzgl.  $x$  und  $v$  stetig differenzierbar.

**Beispiel:** Gegeben sei das einfache Problem

$$\min \{2x : x \geq 0\}$$

Dann ist

$$\Phi_r(x, v) = 2x - \begin{cases} vx - \frac{1}{2}rx^2 & , \text{ falls } x \leq v/r \\ \frac{1}{2}v^2/r & , \text{ sonst} \end{cases}$$

die zugehörige erweiterte  $L_2$ -Lagrange-Funktion.

Der Penalty-Parameter ist hierbei wie folgt zu wählen:

$$r_j^{(k)} := \max \left( \frac{2m(u_j^{(k)} - v_j^{(k)})^2}{(1 - \delta_k) d_k^T B_k d_k}, r_j^{(k-1)} \right).$$

Dabei bezeichnet  $\delta_k$  die zusätzlich in (QPM) eingeführte Variable zur Vermeidung von Inkonsistenz. Für beide Meritfunktionen gilt die folgende Abstiegsseigenschaft:

**Satz:** Für die genannte Meritfunktionen und Penaltyparameter gilt

$$\Psi'_k(0) = \nabla \Phi_{r_k}(x_k, v_k)^T \begin{pmatrix} d_k \\ u_k - v_k \end{pmatrix} < 0.$$

*Beweis:* Der Satz soll der Einfachheit halber nur für die erweiterte Lagrange-Meritfunktion mit  $\delta_k = 0$  und  $m_e = m$  bewiesen werden. Ferner wird jetzt der Index  $k$  weggelassen. Mit  $g(x) := (g_1(x), \dots, g_m(x))^T$  und

$$A(x) := (\nabla g_1(x), \dots, \nabla g_m(x))$$

lauten die Optimalitätsbedingungen für (QP):

$$\begin{aligned} Bd + \nabla f(x) &= A(x)u, \\ A(x)^T d + g(x) &= 0. \end{aligned}$$

Hieraus folgt mit  $R := \text{diag}(r_1, \dots, r_m)$

$$\begin{aligned}
\Psi'_r(0) &= \nabla \Phi_r(x, v)^T \begin{pmatrix} d \\ u - v \end{pmatrix} \\
&= \nabla f(x)^T d - d^T A(x)v + d^T A(x)Rg(x) - g(x)^T(u - v) \\
&= -d^T B d + d^T A(x)(u - v) - g(x)^T Rg(x) - g(x)^T(u - v) \\
&= -d^T B D - 2g(x)^T(u - v) - g(x)^T Rg(x) \\
&= -d^T B d - \|R^{1/2}g(x) + R^{-1/2}(u - v)\|^2 + (u - v)^T R^{-1}(u - v) \\
&\leq -d^T B d + \sum_{j=1}^m \frac{1}{r_j} (u_j - v_j)^2 \\
&\leq -\frac{1}{2} d^T B d \\
&< 0,
\end{aligned}$$

da  $\frac{1}{r_j} (u_j - v_j)^2 \leq \frac{d^T B d}{2m}$ .

## 4.4 Das SQP-Verfahren

Zur Realisierung eines speziellen SQP-Verfahrens werden einige konstante Daten benötigt, wobei die numerischen Zahlen in Klammern als Standardwerte übernommen werden können.

- $\epsilon > 0$  : Stoptoleranz ( $10^{-7}$ )
- $0 < \beta < 1$  : Reduktionsfaktor in Schrittweitenbestimmung (0.1)
- $0 < \mu < 1/2$  : Armijo – Bedingung (0.001)
- $0 < \bar{\delta} < 1$  : Obere Schranke für  $\delta$  (0.9)
- $\bar{\Psi} > 1$  : Multiplikator für  $\Psi_k$  (10.0)
- $\bar{\sigma} > 1$  : Multiplikator für  $r_k$  (10.0)

### Algorithmus:

0) Start: Wähle Startwerte

$$\begin{aligned}
 x_0 &\in R^n && \text{(Benutzervorgabe)} \\
 v_0 &\in R^m && (= (0, \dots, 0)^T) \\
 B_0 &\in R^{n \times n} && (= I) \\
 \Psi_0 &\in R^m && (= 1) \\
 r_0 &\in R^m && (= (.01, \dots, .01)^T)
 \end{aligned}$$

Berechne  $f(x_0)$ ,  $\nabla f(x_0)$ ,  $g_j(x_0)$ ,  $\nabla g_j(x_0)$ ,  $j = 1, \dots, m$ ,  $J_0$  und setze  $k(j) := 0$  für alle  $j \in K_0$ .

Für  $l = 0, 1, 2, \dots$

- 1) Löse das quadratische Teilproblem (QP). Falls die Restriktionen inkonsistent sind, führe neue Variable  $\delta$  ein und löse (QPM). Es sei  $d_k, \delta_k$  die Optimallösung und  $u_k$  der zugehörige Multiplikator. Falls  $\delta_k > \bar{\delta}$ , setze  $\Psi_k := \Psi_k \bar{\Psi}$  und löse (QPM) erneut. Falls diese Schleife nicht erfolgreich ist bzgl. einer oberen Schranke für  $\Psi_k$ , setze

$$\begin{aligned}
 d_k &:= -B_k^{-1} \nabla_x \Phi_{r_{k-1}}(x_k, v_k) \\
 u_k &:= v_k - \nabla_v \Phi_{r_{k-1}}(x_k, v_k) .
 \end{aligned}$$

- 2) Definiere neuen Penaltyparameter  $r_k$ .
- 3) Falls  $\Psi'_k(0) > 0$ , setze  $r_k := \sigma r_k$  und gehe nach 1).
- 4) Bestimme  $\sigma_{k+1}$ .
- 5) Führe Schrittweitenbestimmung bzgl.  $\Psi_k(\alpha)$  durch, wobei  $\Phi_r(x, v)$  die erweiterte Lagrangefunktion ist. Es sei  $\alpha_k$  die berechnete Schrittweite.

6) Setze

$$\begin{aligned}x_{k+1} &:= x_k + \alpha_k d_k \\v_{k+1} &:= v_k + \alpha_k (u_k - v_k)\end{aligned}$$

und berechne  $f(x_{k+1}), g_j(x_{k+1}), j = 1, \dots, m, \nabla f(x_{k+1}), J_{k+1}$  und  $\nabla g_j(x_{k+1}), j \in J_{k+1}$ .

7) Bestimme  $B_{k+1}$  nach dem BFGS-Verfahren mit

$$q_k := \nabla_x L(x_{k+1}, u_k) - \nabla_x L(x_k, u_k) .$$

Weitere Einzelheiten findet man in Schittkowski (1983) und Schittkowski (1985/86). Im Algorithmus können folgende Stoppkriterien, z.T. alternativ eingesetzt werden:

$$\begin{aligned}d_k^T B_k d_k &< \epsilon^2 \\|\nabla f(x_k)^T d_k| + \sum_{j=1}^m |u_j^{(k)} g_j(x_k)| &< \epsilon \\\|\nabla_x L(x_k, u_k)\| &< \sqrt{\epsilon} \\\sum_{j=1}^m |g_j(x_k)| + \sum_{j=m_e+1}^m |\min(0, g_j(x_k))| &< \sqrt{\epsilon} .\end{aligned}$$

## 4.5 Konvergenzresultate

Man unterscheidet zwischen globalen und lokalen Konvergenzaussagen, wobei jeweils von unterschiedlichen Voraussetzungen ausgegangen wird. Eine Übersicht über bekannte Konvergenzuntersuchungen findet man bei Stoer (1985). Stellvertretend werden vier Sätze herausgegriffen.

**Satz (Han (1977)):** *Es sei  $\{(x_k, v_k)\}$  eine Iterationsfolge eines SQP-Verfahrens, wobei (QP) stets lösbar sei,  $\Psi_k(\alpha)$  die exakte  $L_1$ -Penaltyfunktion ist und  $r_k = r$  für alle  $k$  gilt. Ferner sei*

- (i)  $\Psi_k(\alpha) \leq \min\{\Psi_k(\alpha) : 0 < \alpha \leq 1\} + \epsilon_k, \sum_{k=1}^{\infty} \epsilon_k < \infty, \epsilon_k \geq 0$  für alle  $k$ ,
- (ii)  $\alpha d^T d \leq d^T B_k d \leq \beta d^T d$  für alle  $d \in R^n$  und für alle  $k$ ,
- (iii)  $\|u_k\|_{\infty} \leq \|r\|_{\infty}$  für alle  $k$ .

*Dann ist jeder Häufungspunkt von  $\{(x_k, v_k)\}$ , der die constraint qualification erfüllt, ein Kuhn-Tucker-Punkt von (NLP).*

**Satz (Schittkowski (1983)):** *Es sei  $\{(x_k, v_k)\}$  eine Iterationsfolge des Algorithmus (4.4), und es gelte*

- (i)  $d_k^T B_k d_k \geq \gamma d_k^T d_k$  für alle  $k$  und ein  $\gamma > 0$ ,
- (ii)  $\delta_k \leq \bar{\delta}$  für alle  $k$ ,
- (iii)  $\sigma_k \geq \|A(x_k)v_k\|^2 / \gamma(1 - \bar{\delta})^2$  für alle  $k$ ,
- (iv)  $\{x_k\}, \{d_k\}, \{u_k\}, \{B_k\}$  seien beschränkt.

*Dann gibt es einen Häufungspunkt von  $\{(x_k, v_k)\}$ , der die Kuhn-Tucker-Bedingung für (NLP) erfüllt.*

Man beachte, daß die spezielle Wahl der Matrizen  $B_k$  noch unbedeutend ist. Lokale Konvergenzaussagen dienen der Ermittlung der Konvergenzgeschwindigkeit in der Nähe einer Optimallösung. Es seien folgende Voraussetzungen erfüllt:

- a)  $z^* = (x^*, u^*)$  sei ein strenges lokales Minimum von (NLP).
- b)  $m_e = m$ , d.h. es seien alle aktiven Restriktionen bekannt.
- c)  $f, g_1, \dots, g_m$  seien zweimal stetig differenzierbar.
- d) Für  $z_k := (x_k, v_k)$  gelte  $\lim_{k \rightarrow \infty} z_k = z^*$ .
- e) Die Gradienten  $\nabla g_1(x^*), \dots, \nabla g_m(x^*)$  seien linear unabhängig (*constraint qualification*).
- f)  $d^T B_k d \geq \gamma d^T d$  für alle  $d \in R^n$  mit  $A(x_k)^T d = 0$ .

**Satz** (Han (1976)): *Es gelte zusätzlich:*

- (i)  $\nabla_{xx}L(x^*, u^*)$  ist positiv definit,
- (ii)  $\alpha_k = 1$  für alle  $k$ ,
- (iii)  $B_k$  ist nach der DFP-Formel bestimmt,
- (iv)  $\|B_0 - \nabla_{xx}L(x^*, u^*)\|$  ist hinreichend klein.

Dann konvergiert  $\{z_k\}$   $Q$ -superlinear, d.h.

$$\lim_{k \rightarrow \infty} \frac{\|z_{k+1} - z^*\|}{\|z_k - z^*\|} = 0$$

Wichtig ist hier, daß (i) für viele Probleme nicht erfüllt ist. Ferner kann (iv) i.R. nicht gewährleistet werden. Die Konvergenzaussage bedeutet, daß es eine Nullfolge  $\{\beta_k\}$  gibt mit

$$\|z_{k+1} - z^*\| \leq \beta_k \|z_k - z^*\| .$$

Hieraus folgt allerdings noch nicht die superlineare Konvergenz von  $\{x_k\}$ .

**Satz** (Powell (1978b)): *Es sei  $\alpha_k = 1$  für alle  $k$ . Dann konvergiert  $\{x_k\}$   $R$ -superlinear gegen  $x^*$ , d.h.*

$$\lim_{k \rightarrow \infty} \|x_k - x^*\|^{1/k} = 0$$

Es wird im obigen Satz vom BFGS-Verfahren ausgegangen, wobei die Startmatrix beliebig gewählt werden darf. Allerdings ist die  $R$ -superlineare Konvergenz schwächer als die  $Q$ -superlineare Konvergenz, jedoch noch völlig ausreichend, um das praktische Konvergenzverhalten theoretisch zu fundieren. Es gibt also eine Nullfolge  $\{\beta_k\}$  mit

$$\|x_k - x^*\| \leq \beta_k^k .$$

## 5. Numerische Vergleichstests

### 5.1 Testbeispiele

Zur numerischen Berechnung von Leistungsdaten benötigt man geeignete Testbeispiele. Hierzu stehen zwei grundsätzlich verschiedene Ansätze zur Verfügung.

#### a) Künstliche und anwendungsorientierte Probleme

Hierzu zählen in der Regel die Testbeispiele, die in der Vergangenheit zur Entwicklung und zum Test von Optimierungsprogrammen benutzt werden und die in Form von Beispielsammlungen vorliegen, siehe z.B. Hock und Schittkowski (1981) oder Schittkowski (1987). Sie wurden entweder *per Hand* entwickelt, z.B.

$$\begin{aligned} \text{TP1} \quad & \text{(Rosenbrock - Problem) :} \\ & \min_{x_1, x_2} 100(x_2 - x_1^2)^2 + (1 - x_1)^2 \\ & x_0 = (-1.2, 1)^T, x^T = (1, 1)^T, \end{aligned}$$

oder besitzen einen praktischen Hintergrund, z.B.

$$\begin{aligned} \text{TP104} \quad & \text{(Reaktorentwurf) :} \\ & \min .4x_1^{.67}x_7^{-.67} + .4x_2^{.67}x_8^{-.67} + 10 - x_1 - x_2 \\ & .0588x_5x_7 + .1x_1 \leq 1 \\ & .0588x_6x_8 + .1x_1 + .1x_2 \leq 1 \\ & 4x_3x_5^{-1} + 2x_3^{-.71}x_5^{-1} + .0588x_3^{-1.3}x_7 \leq 1 \\ & 4x_3x_6^{-1} + 2x_4^{-.71}x_6^{-1} + .0588x_4^{-1.3}x_8 \leq 1 \\ & 0.01 \leq x_i \leq 10, i = 1, \dots, 8 \\ & x_0 = (6, 3, .4, .2, 6, 6, 1, .5)^T. \end{aligned}$$

Der Schwierigkeitsgrad eines Testbeispiels läßt sich nicht vom optischen Eindruck herleiten. So benötigt man z.B. zur Lösung von TP1 ca. 31 Iterationen, zur Lösung von TP104 ca. 16 Iterationen mit einem SQP-Verfahren.

#### b) Zufallsmäßig erzeugte Testbeispiele

Die bisher besprochenen Beispiele besitzen den Nachteil, daß nur unzureichende Informationen über die Struktur der Lösung vorliegen. Außerdem ist die Programmierung einer großen Anzahl von Testproblemen aufwendig. Dagegen können für zufallsmäßig erzeugte Testbeispiele nicht nur der optimale Variablenvektor  $x^*$  vorgegeben werden, sondern auch die zugehörigen Multiplikatoren und die Hessematrix der Lagrange-Funktion. Hierzu benötigt man eine Möglichkeit, zufallsmäßig nichtlineare Funktionen zu erzeugen, z.B. verallgemeinerte Polynome der Form

$$s(x) = \sum_{r=1}^k c_r \prod_{i=1}^n x_i^{a_{ir}}$$

mit  $c_i, a_{ir} \in \mathbb{R}$ .

### Konstruktionsverfahren:

- a) Wähle  $x^* \in \mathbb{R}^n$  zufallsmäßig innerhalb von vorgegebenen Schranken  $x_l$  und  $x_u$ .
- b) Bestimme  $m + 1$  verallgemeinerte Polynome  $s_0, s_1, \dots, s_m$ , wobei die Koeffizienten und Exponenten zufallsmäßig ermittelt werden.
- c) Setze

$$\begin{aligned} g_j(x) &:= s_j(x) - s_j(x^*), & j = 1, \dots, m_e + m_a \\ g_j(x) &:= s_j(x) - s_j(x^*) + \mu_j, & j = m_e + m_a + 1, \dots, m, \end{aligned}$$

wobei  $\mu_j \in (0, m)$  beliebig ist.

- d) Wähle positiv definite Matrix  $P$ , z.B.  $P = LL^T$ ,  $L$  zufallsmäßig bestimmte untere Dreiecksmatrix.
- e) Bestimme zufallsmäßig Multiplikatorvektor  $u^*$ , so daß  $u_j^* > 0$  für  $j = m_e + 1, \dots, m_e + m_a$  und  $u_j^* = 0$  für  $j = m_e + m_a + 1, \dots, m$ .
- f) Setze

$$f(x) := s_0(x) + \frac{1}{2}x^T Hx + c^T x + \alpha$$

mit

$$\begin{aligned} H &:= -\nabla^2 s_0(x^*) + \sum_{j=1}^m u_j^* \nabla^2 g_j(x^*) + P, \\ c &:= -\nabla s_0(x^*) - Hx^T + \sum_{j=1}^m u_j^* \nabla g_j(x^*), \\ \alpha &:= -\frac{1}{2}x^{*T} Hx^* - c^T x^* \end{aligned}$$

Die vorgegebenen Vektoren  $x^*$  und  $u^*$  erfüllen damit die notwendigen und hinreichenden Optimalitätsbedingungen einer isolierten lokalen Lösung:

$$\begin{aligned} g_j(x^*) &= 0, & j = 1, \dots, m_e + m_a, \\ g_j(x^*) &> 0, & j = m_e + m_a + 1, \dots, m, \\ u_j^* &\geq 0, & j = m_e + 1, \dots, m, \\ \nabla_x L(x^*, u^*) &= 0, \\ f(x^*) &= 0. \end{aligned}$$

Konvexe Problemfunktionen erhält man, falls die verallgemeinerten Polynome durch Exponentialfunktionen der Form

$$s(x) = \sum_{r=1}^k c_r \exp\left(\sum_{i=1}^n a_{ir} x_i\right)$$

mit  $c_r > 0$  ersetzt werden.

## 5.2 Einige allgemeine Vergleichsresultate

Es sollen Effizienz- und Zuverlässigkeitsresultate für einige Optimierungsprogramme vorgestellt werden, die typisch sind für das benutzte mathematische Verfahren. Zu ihrer Charakterisierung werden folgende Abkürzungen benutzt:

- PE : Penalty- oder Strafkostenverfahren
- MU : Multiplikator- oder erweitertes Lagrangeverfahren
- GRG : Verallgemeinertes reduziertes Gradientenverfahren
- SQP : Sequentielles quadratisches Programmierungsverfahren

In der nachfolgenden Tabelle sind die Programme aufgeführt, deren Leistungsdaten numerisch miteinander verglichen werden sollen:

Programm	Quelle	Verfahren
SUMT	Fiacco, McCormick (1968)	PE
NLP	Rufer (1978)	PE
LPNLP	Pierre, Lowe (1975)	MU
VF01A	Fletcher (1975)	MU
GRGA	Abadie (1978)	GRG
GRG2	Lasdon e.al. (1978)	GRG
VF0LAD	Powell (1978a)	SQP
NLPQL	Schittkowski (1985/86)	SQP

Die vorgestellten Ergebnisse beziehen sich auf 240 Testläufe mit zufallsmäßig erzeugten Testproblemen und können in Schittkowski (1980) nachgelesen werden. Für die in 5.1 erwähnten künstlichen und anwendungsbezogenen Testbeispiele gelten analoge Resultate, vergleiche hierzu Hock und Schittkowski (1981). Die Testläufe basieren auf 80 Testbeispielen, die mit jeweils drei verschiedenen Startwerten versehen werden. Für die Dimensionsparameter gelten die Einschränkungen

$$4 \leq n \leq 20, 3 \leq m \leq 18, 0 \leq m_e \leq 3, 0 \leq m_a \leq 12 .$$

Folgende durchschnittliche Effizienz- und Zuverlässigkeitsresultate werden aufgelistet:

- ET – Rechenzeit in Sekunden,
- NF – Anzahl der Funktionsauswertungen von  $f$ ,
- NG – Anzahl der Funktionsauswertungen von  $g_1, \dots, g_m$ ,
- NDF – Anzahl der Gradientenauswertungen von  $f$ ,
- NDG – Anzahl der Gradientenauswertungen von  $g_1, \dots, g_m$ ,
- PNS – Prozentsatz der nicht erfolgreichen Testläufe,
- F – Anzahl der Fehlläufe.

Die Mittelwerte beziehen sich jeweils auf die bzgl. vorgegebener Toleranzen erfolgreichen Testläufe.

Programm	ET	NF	NG	NDF	NDG	PNS	F
SUMT	270.1	2335	24046	99	1053	69.9	19
NLP	88.1	1043	8635	111	957	15.6	31
LPNLP	57.4	252	2518	101	1014	26.7	10
VF01A	42.2	158	1596	158	603	23.9	12
GRGA	37.7	204	2946	67	378	12.1	3
GRG2	52.6	297	3368	38	423	10.4	0
VF02AD	31.5	16	179	16	179	6.2	5
NLPQL	14.1	18	181	16	64	3.3	0

Die in NLPQL auftretenden quadratischen Teilprobleme werden mit dem Unterprogramm QPSOL von Gill, Murray, Saunders und Wright (1982) gelöst. Die wichtigsten Schlußfolgerungen sind einmal, daß unterschiedliche Implementierungen derselben mathematischen Methode zu Programmen mit unterschiedlichen Leistungsdaten führen können und zum anderen, daß die SQP-Verfahren den anderen deutlich bzgl. Effizienz und Zuverlässigkeit überlegen sind.

### 5.3 Vergleichsresultate für Strukturoptimierungsprobleme

Die im letzten Abschnitt angegebenen Resultate lassen sich nicht ohne weiteres auf die Lösung von Strukturoptimierungsproblemen übertragen. In Abhängigkeit vom dominierenden Typ der Restriktionen wurden Spezialverfahren konstruiert, die in vielen Fällen den SQP-Verfahren überlegen sind, siehe Fleury (1989) oder Svanberg (1987). Der Grund liegt in der speziellen Struktur der Restriktionen, bei denen häufig eine Einführung von reziproken Variablen die Nebenbedingung *linearer* macht.

Es sollen hier nur einige wenige Resultate einer umfangreichen numerischen Vergleichsstudie dargestellt werden, s. Schittkowski, Zillober und Zotemantel (1994). Sie beruhen auf einer Sammlung von 79 Testbeispielen, die von MBB-UF Flugzeuge zur Verfügung gestellt wurden und in der nachfolgenden Tabelle 1 zusammengefaßt werden. Sie enthält neben den Namen der Beispiele Informationen über charakterisierende Daten, z.B.:

NET	-	Anzahl der Finiten Elemente
NOD	-	Anzahl der Knoten
NLY	-	Anzahl der Lagen bei Kompositelementen
NLC	-	Anzahl der Lastfälle
NDT	-	Anzahl der Freiheitsgrade
NSV	-	Anzahl der Strukturvariablen
NDV	-	Anzahl der Entwurfsvariablen
NDG	-	Gesamtzahl aller Restriktionen

Während ca. 50% der Beispiele aus der Literatur stammen, repräsentieren die restlichen Beispiele reale Anwendungsprobleme bzw. deren Modifikationen. Sie wurden im Zusammenhang mit der Entwicklung des Strukturoptimierungssystems MBB-LAGRANGE für Teststudien gesammelt.

MBB-LAGRANGE ist ein Strukturoptimierungssystem, das insbesondere für Anwendungen aus der Luft- und Raumfahrt entwickelt wurde, siehe Kneppe, Krammer und Winkler (1987). Die folgenden Restriktionstypen sind definierbar, die sich alle in der ein oder anderen Form in den Testbeispielen wiederfinden:

- Verformungen
- Dehnungen
- Spannungen
- Stabilität (lokal und global)
- aeroelastische Wirksamkeiten
- Flattergeschwindigkeiten
- Divergenzgeschwindigkeiten
- Eigenwerte und Eigenformen
- dynamische Systemantworten

- Fertigungsbedingungen
- minimale und maximale Abmessungen

Die zugehörigen Entwurfsvariablen sind

- Querschnittsflächen von Stäben und Balken
- Wanddicken von Membran- und Schalenelementen
- Lagendicken und -winkel von Faserverbundwerkstoffen
- Knotenkoordinaten
- Ausgleichsmassen

In das System MBB-LAGRANGE wurde eine Vielzahl mathematischer Optimierungsverfahren eingebaut basierend auf der Erkenntnis, daß es kein Verfahren gibt, daß auch nur annähernd alle auftretenden Strukturoptimierungsprobleme in der gewünschten Weise lösen könnte:

- |        |  |
|--------|--|
| IBF    | - inverses Penaltyverfahren, s. Fiacco, McCormick (1969)                         |
| MOM    | - Multiplikatorverfahren, s. Reklaitis, Ravindran und Ragsdell (1983)            |
| SLP    | - sequentielle lineare Programmierung, s. Knepe (1986)                           |
| SRM    | - <i>stress ratio</i> -Methode, s. Berke und Khot (1974)                         |
| RQP1   | - sequentielle quadratische Programmierung, s. Schittkowski (1985/86)            |
| RQP2   | - sequentielle quadratische Programmierung, s. Powell (1978a)                    |
| GRG    | - verallgemeinertes reduziertes Gradientenverfahren, s. Bremicker (1986)         |
| CONLIN | - konvexes Approximationsverfahren, s. Fleury (1989)                             |
| SCP    | - konvexes Approximationsverfahren mit Schrittlängenwahl, s. Zillober (1993)     |
| MMA    | - Methode der <i>Moving Asumptotes</i> , s. Zillober (1994) bzw. Svanberg (1987) |

Die Leistungsdaten der Tabellen 2 und 3 beziehen sich auf die bzgl. einer Fehlerschranke  $\epsilon > 0$  erfolgreichen Testläufe. Ein Iterationsvektor  $x_k$  erfüllt dabei die Genauigkeitsschranke  $\epsilon$ , falls

$$f(x_k) \leq (1 + \epsilon)f(x^*) , \quad r(x_k) \leq \epsilon.$$

Hierbei ist  $f(x_k)$  der Zielfunktionswert,  $r(x_k)$  die Gesamtsumme aller Restriktionsverletzungen und  $x^*$  die Optimallösung, die entweder der Literatur entnommen wurde oder durch separate Testläufe mit hoher Endgenauigkeit ermittelt wurde.

In Tabelle 2 soll ein Eindruck von der Zuverlässigkeit und die erzielbare Endgenauigkeit der betrachteten Verfahren gegeben werden. Hierzu wird jeweils die Zahl der ungelösten Beispiele für vier verschiedene Genauigkeitsschranken angegeben.

In Tabelle 3 werden bezogen auf diverse Endgenauigkeiten jeweils die erzielten Leistungszahlen der betrachteten Verfahren angegeben, um einen Eindruck von der Effizienz der numerischen Verfahren zu geben. Dabei beziehen sich die angegebenen Werte auf die Rechenzeit bzw. die Zahl der durchgeführten Analysen bis zum Erreichen der Stoppbedingung. Zur Herleitung der Leistungskriterien sei auf Schittkowski, Zillober und Zotemantel (1994) verwiesen.

Aus den numerischen Resultaten lassen sich folgende Schlußfolgerungen ableiten:

- Nur mit Hilfe der SQP-Verfahren läßt sich eine hohe Endgenauigkeit erzielen.
- Die SQP-Verfahren sind die zuverlässigsten Verfahren, d.h. lösen die meisten Testbeispiele.
- Falls ein konvexes Approximationverfahren konvergiert, ist es deutlich effizienter als alle anderen Verfahren.
- Sequentielle Linearisierungsverfahren lassen sich alternativ zur Lösung von Strukturoptimierungsproblemen einsetzen. Sie sind effizient und hinreichend zuverlässig.
- Penalty- und Multiplikatorverfahren sind unzuverlässig und ineffizient. Sie sollten nur in Spezialfällen benutzt werden.

No	Modell	Testbeispiel	NET	NOD	NLY	NLC	NDT	NSV	NDV	NDG
1	1	APLATE2	49	64	1	1	175	49	7	51
2	1	APLCON	49	64	1	1	175	49	32	51
3	2	ARIANB	310	357	1	5	2136	310	24	1020
4	3	B4SIZE	130	88	4	1	240	520	72	522
5	3	B4SIZELA	130	88	4	1	240	520	83	522
6	4	BALKEN2	2	4	1	2	10	2	2	4
7	5	BARBIG	9	11	1	0	54	9	9	91
8	6	BARDISP	3	4	1	1	12	3	2	3
9	6	BARDYN2	3	4	1	1	12	3	3	4
10	6	BARDYN5	3	4	1	1	12	3	3	4
11	7	BARMASS	8	10	1	1	48	8	8	1
12	8	BAROF9	7	4	1	1	18	7	7	7
13	9	BUST	8	15	1	2	54	8	8	16
14	9	BUSTB	8	9	1	2	30	8	8	16
15	9	BUSTM	4	6	1	2	8	4	4	8
16	9	BUSTQ	8	15	1	2	54	8	8	16
17	9	BUSTQM	2	6	1	2	8	2	2	4
18	9	BUSTT	10	15	1	2	54	10	10	20
19	10	CANE	40	44	1	1	120	40	10	41
20	11	CELAS4	22	29	1	1	87	18	2	18
21	12	COMPKRAGS	4	7	4	2	10	16	10	44
22	13	COOLPLATE	177	180	1	1	971	177	8	177
23	14	CORDS	63	35	4	1	87	135	8	96
24	15	DREI	3	4	1	2	2	3	3	6
25	15	DREIDISP	3	4	1	2	2	3	2	6
26	15	DREISLP	3	4	1	2	2	3	3	6
27	16	DYNPLT	63	60	1	1	250	63	10	157
28	17	EFA2CLAG	492	385	1	1	1859	492	54	153
29	18	F1F3	24	28	9	1	52	100	13	67
30	19	GARTEUR1	83	79	1	0	231	83	13	21
31	20	GBOX	97	89	1	1	473	97	6	98
32	21	GENTEST	3	4	1	2	18	1	1	2
33	22	GRADPLA	4	9	1	2	41	4	4	10
34	23	HEXA	9	52	1	1	148	9	6	13
35	24	JPATS16	252	214	2	1	406	744	144	416
36	25	KALA3_1	9	9	4	1	18	75	37	40
37	26	KRAG	4	7	3	2	10	7	7	14
38	26	KRAGBA	8	10	4	2	16	12	10	26
39	26	KRAGBAD	8	10	4	2	16	12	10	25
40	26	KRAGBAM	8	10	4	2	16	12	10	20
41	26	KRAGDYN	8	10	4	2	16	12	8	27
42	26	KRAGIBF	4	7	3	2	10	7	7	14

*Tabelle 1:* Zusammenfassung aller Testbeispiele (fortgesetzt)

No	Model	Test Example	NET	NOD	NLY	NLC	NDT	NSV	NDV	NDG
43	26	KRAGMAN	4	7	4	2	10	16	10	44
44	26	KRAGMOM	4	7	3	2	10	7	7	14
45	27	KRAN0	54	20	1	1	112	54	54	54
46	27	KRAN1	54	20	1	1	112	54	54	55
47	28	LAGTEST	40	36	1	1	158	40	8	40
48	10	MCANE	40	44	1	1	120	40	40	41
49	29	MOMENT	8	9	1	2	48	8	8	16
50	30	MPC	18	29	1	1	113	18	2	18
51	31	MPLATE	6	10	1	2	26	6	6	12
52	31	MPLATEH	6	10	1	2	26	6	6	12
53	32	OPM2	208	133	1	0	516	208	11	2
54	33	PLATTE	10	16	4	1	27	40	10	41
55	34	POINTMASS	8	10	1	0	48	8	8	1
56	35	PUNCHTEST	50	31	4	1	60	86	20	86
57	36	QUARTOPLT	4	9	1	4	10	4	4	20
58	37	QUATRIA	39	49	8	1	84	92	35	99
59	38	RBAR	25	29	1	1	134	25	2	25
60	39	SCHEIT210	8	9	4	1	15	32	4	32
61	40	SPANT	20	15	1	1	26	20	13	20
62	41	SPSLP	414	196	1	1	389	414	108	414
63	41	SPSLPB	414	196	1	1	389	414	108	415
64	42	T3D066	9	52	1	1	148	9	6	13
65	43	TBDYN	10	6	1	0	8	10	10	1
66	44	TEMPER2	16	25	1	1	18	16	12	12
67	43	TENBAR	10	6	1	1	8	10	10	10
68	43	TENCON	10	6	1	1	8	10	10	10
69	43	TENMOM	10	6	1	1	8	10	10	10
70	43	TENRQP	10	6	1	1	8	10	10	10
71	43	TENSRM	10	6	1	1	8	10	10	10
72	2	TESTCORD1	173	207	1	2	1017	173	11	176
73	2	TESTCORD4	173	207	1	2	1027	173	11	176
74	45	TESTNAS0	30	18	4	2	72	42	30	84
75	46	TESTPARDYN	9	11	1	0	54	9	9	91
76	47	TR4X4	32	25	1	1	56	32	32	32
77	48	TRIOM	10	18	1	1	32	10	5	13
78	49	TSHELL3	1	3	1	2	3	1	1	2
79	50	TUBE	60	32	4	1	72	132	8	98

*Tabelle 1: Zusammenfassung aller Testbeispiele*

Code	$\epsilon = 0.01$		$\epsilon = 0.001$		$\epsilon = 0.0001$		$\epsilon = 0.00001$	
MOM	30	(38 %)	51	(65 %)	60	(76 %)	67	(85 %)
SLP	20	(25 %)	27	(34 %)	31	(39 %)	37	(47 %)
RQP1	13	(16 %)	15	(19 %)	19	(24 %)	23	(29 %)
RQP2	18	(23 %)	20	(25 %)	22	(28 %)	26	(33 %)
GRG	26	(33 %)	30	(38 %)	31	(39 %)	31	(39 %)
QPRLT	15	(19 %)	20	(25 %)	24	(30 %)	28	(35 %)
CONLIN	34	(43 %)	37	(47 %)	42	(53 %)	43	(54 %)
SCP	19	(24 %)	24	(30 %)	25	(32 %)	28	(35 %)
MMA	21	(27 %)	25	(32 %)	25	(32 %)	27	(34 %)

Table 3: Anzahl ungelöster Beispiele

Code	$\epsilon = 0.01$	$\epsilon = 0.001$	$\epsilon = 0.0001$	$\epsilon = 0.00001$
MOM	27.16	31.70	31.58	28.79
	43.72	48.88	47.76	49.64
	55.67	61.55	63.20	54.68
SLP	7.28	6.82	7.30	6.91
	3.66	3.29	4.22	3.21
	6.62	5.84	7.25	6.85
RQP1	14.15	12.97	13.07	13.63
	5.59	4.47	4.59	4.51
	7.69	6.15	5.75	7.43
RQP2	9.39	8.25	8.31	8.97
	4.49	3.64	3.88	3.93
	7.05	5.38	5.23	6.89
GRG	11.20	10.80	10.68	11.06
	17.34	15.87	15.71	15.56
	3.62	3.03	2.71	3.41
QPRLT	8.61	8.13	8.17	8.36
	13.45	12.56	12.72	11.77
	3.02	2.66	2.53	2.96
CONLIN	5.82	5.68	5.33	5.66
	2.63	3.12	2.35	2.45
	4.59	5.18	3.49	4.60
SCP	9.54	9.06	9.03	9.59
	6.39	5.59	6.01	6.07
	7.11	5.94	5.65	7.43
MMA	6.86	6.59	6.54	7.04
	2.73	2.59	2.75	2.86
	4.64	4.26	4.19	5.75

Table 8: Leistungsindizes für Rechenzeit, Anzahl der Analysen und Gradientenauswertungen, jeweils untereinander

## Literatur:

ABADIE J. (1978): *The GRG method for nonlinear programming*, in: Design and Implementation of Optimization Software, H. Greenberg ed., Sijthoff and Noordhoff

BERKE L., KHOT N.S. (1974): *Use of optimality criteria methods for large scale systems*, AGARD Lecture Series No.70 on Structural Optimization

BREMICKER M. (1986): *Entwicklung eines Optimierungsalgorithmus der generalisierten reduzierten Gradienten*, Report, Forschungslaboratorium für angewandte Strukturopti- Strukturoptimierung, Universität-GH Siegen

FIACCO A.V., MCCORMICK G.P. (1968): *Nonlinear sequential unconstrained minimization techniques*, John Wiley & Sons

FLETCHER R. (1975): *An ideal penalty function for constrained optimization*, in: Nonlinear Programming 2, O.L. Mangasarian, R.R. Meyer, S.M. Robinson eds., Academic Press

FLEURY C. (1989): *CONLIN: An efficient dual optimizer based on convex approximation concepts*, Structural Optimization, Vol.1, 81-89

GILL P.E., MURRAY W., SAUNDERS M.A. (1975): *Methods for computing and modifying the LDV factors of a matrix*, Mathematics of Computation, Vol.29, 1051-1077

GILL P.E., MURRAY W., SAUNDERS M.A., WRIGHT M. (1982): *User's guide for SOL/QPSOL: A FORTRAN package for quadratic programming*, Report SOL 82-7, Dept. of Operations Research, Stanford University, USA

HAN S.-P. (1976): *Superlinearly convergent variable metric algorithms for general nonlinear programming problems*, Mathematical Programming, Vol.11, 263-282

HAN S.-P. (1977): *A globally convergent method for nonlinear programming*, Journal of Optimization Theory and Applications, Vol.22, 297-309

HOCK W., SCHITTKOWSKI K. (1981): *Test examples for nonlinear programming codes*, Lecture Notes in Economics and Mathematical Systems, Vol.187, Springer

KNEPPE G., KRAMMER H., WINKLER F. (1987): *Structural optimization of large scale problems using MBB-LAGRANGE*, 5th World Congress and Exhibition on FEM, Salzburg, Austria

KNEPPE G. (1986): *Direkte Lösungsstrategien zur Gestaltoptimierung von Fächentragwerken*, VDI-Verlag

KRAFT D. (1977): *Nichtlineare Programmierung - Grundlagen, Verfahren, Beispiele*, Forschungsbericht, DFVLR Oberpfaffenhofen

- LASDON L.S., WARREN A.D., RATNER M.W. (1978): *GRG2 user's guide*, Report, School of Business Administration, The University of Texas at Austin, USA
- PIERRE D.A., LOWE M.J. (1975): *Mathematical programming via augmented Lagrangian. An introduction with computer programs*, Addison-Wesley
- POWELL M.J.D. (1978A): *A fast algorithm for nonlinearly constrained optimization calculations*, in: Numerical Analysis, G.A. Watson ed., Lecture Notes in Mathematics, Vol.630, Springer
- POWELL M.J.D. (1978B): *The convergence of variable metric methods for nonlinearly constrained optimization calculations*, in: Nonlinear Programming 3, O.L. Mangasarian, R.R. Meyer, S.M. Robinson eds., Academic Press
- REKLAITIS G.V., RAVINDRAN A., RAGSDALL K.M. (1983): *Engineering Optimization*, John Wiley
- RUFER D. (1979): *Implementation and properties of a method for the identification of nonlinear continuous time models*, Proceeding of the 7th IFAC World Congress, A. Niemi ed., Pergamon Press
- SCHITTKOWSKI K. (1980): *Nonlinear programming codes*, Lecture Notes in Economics and Mathematical Systems, Vol.183, Springer
- SCHITTKOWSKI K. (1983): *On the convergence of a sequential quadratic programming method with an augmented Lagrangian line search function*, Optimization, Mathematische Operationsforschung und Statistik, Vol.14, 197-216
- SCHITTKOWSKI K. (1985/86): *NLPQL: A FORTRAN subroutine solving constrained nonlinear programming problems*, Annals of Operations Research, Vol.5, 485-500
- SCHITTKOWSKI K. (1987): *More test examples for nonlinear programming codes*, Lecture Notes in Economics and Mathematical Systems, Vol.282, Springer
- SCHITTKOWSKI K., ZILLOBER C., ZOTEMANTEL R. (1994): *Numerical Comparison of Nonlinear Programming Algorithms for Structural Optimization* Structural Optimization, Vol. 7, No. 1, 1-28
- STOER J. (1985): *Principles of sequential quadratic programming methods for solving nonlinear programs*, in: Computational Mathematical Programming, K. Schittkowski ed., NATO ASI Series F: Computer and System Sciences, Vol.15, 165-208
- SVANBERG K. (1987): *Method of moving asymptotes - a new method for structural optimization*, International Journal on Numerical Methods in Engineering, Vol.24, 359-373
- ZILLOBER C. (1993): *SCP - An implementation of a sequential convex programming algorithm for nonlinear programming*, Report No. 470, Mathematisches Institut, Universität Bayreuth