

# Kapitel 5

## Lösungsverfahren

Dieses Kapitel gibt einen Überblick über Lösungsverfahren für nichtlineare Optimierungsprobleme. Es wird vor allem auf die wesentlichen Ideen der Verfahren eingegangen und weniger auf Details.

### 5.1 Projektionsverfahren

Projektionsverfahren sind recht einfache Verfahren, die das Konzept von Abstiegsverfahren zur Minimierung von Funktionen ohne Nebenbedingungen auf Minimierungsprobleme mit konvexen Nebenbedingungen übertragen. Bei einem Abstiegsverfahren wird, ausgehend von einer Iterierten  $\mathbf{x}^{(k)}$ , die nächste Iterierte  $\mathbf{x}^{(k+1)}$  so gewählt, dass sich der Wert der zu minimierenden Funktion verkleinert. Hat man ein Problem mit Nebenbedingungen, so muss man natürlich zusätzlich darauf achten, dass  $\mathbf{x}^{(k+1)}$  zum zulässigen Bereich gehört. Anderenfalls kann es zum Beispiel vorkommen, dass die Zielfunktion gar nicht definiert ist. Projektionsverfahren projizieren die Abstiegsrichtung für die Zielfunktion in geeigneter Weise in die zulässige Menge.

Wir betrachten das Optimierungsproblem

$$z = \min\{f(\mathbf{x}) : \mathbf{x} \in \Omega\} \quad (5.1)$$

mit  $f \in C^1(\mathbb{R}^n)$  und die zulässige Menge  $\Omega$  sei nichtleer, konvex und abgeschlossen.

Von besonderer Bedeutung sind die Probleme, bei denen  $f(\mathbf{x})$  quadratisch und

$$\Omega = \{\mathbf{x} \in \mathbb{R}^n : A\mathbf{x} \leq \mathbf{b}\} \quad (5.2)$$

ein Polyeder ist,  $A \in \mathbb{R}^{m \times n}$ ,  $\mathbf{b} \in \mathbb{R}^m$ . Solche Probleme treten als Teilprobleme bei den sogenannten SQP-Verfahren (sequential quadratic programming) auf, wo sie wiederholt mit unterschiedlichen Daten  $A, \mathbf{b}, f$  gelöst werden müssen, siehe Abschnitt 5.4.

In anderen Anwendungen ist der zulässige Bereich sogar nur ein  $n$ -dimensionaler Quader („box constraints“):

$$\Omega = \times_{i=1}^n [l_i, u_i]. \quad (5.3)$$

Für  $\mathbf{x} \in \mathbb{R}^n$  sei  $P_\Omega(\mathbf{x})$  die Lösung von

$$\inf_{\mathbf{y} \in \Omega} \|\mathbf{y} - \mathbf{x}\|_2,$$

die Projektion von  $\mathbf{x}$  auf  $\Omega$  bezüglich der Euklidischen Norm. Da die Menge  $\Omega$  nach Voraussetzung abgeschlossen ist, kann man das Infimum durch das Minimum ersetzen, also

$$P_\Omega(\mathbf{x}) = \arg \min_{\mathbf{y} \in \Omega} \|\mathbf{y} - \mathbf{x}\|_2.$$

**Beispiel 5.1** Falls  $\Omega$  durch (5.2) gegeben ist, muss man zur Berechnung der Projektion  $\bar{\mathbf{y}} = P_{\Omega}(\mathbf{x})$  das konvexe quadratische Programm

$$\bar{\mathbf{y}} = \arg \min_{\mathbf{y} \in \Omega} \{\|\mathbf{y} - \mathbf{x}\|_2^2 : \mathbf{A}\mathbf{y} \leq \mathbf{b}\} = \arg \min_{\mathbf{y} \in \Omega} \{\mathbf{y}^T \mathbf{y} - 2\mathbf{y}^T \mathbf{x} : \mathbf{A}\mathbf{y} \leq \mathbf{b}\}$$

lösen. Die Menge  $\Omega$  ist ein konvexes Polyeder. Die Zielfunktion  $\|\mathbf{y} - \mathbf{x}\|_2^2$  ist eine konvexe Funktion.

Ist der zulässige Bereich ein  $n$ -dimensionaler Quader (5.3), kann man die Projektion komponentenweise berechnen:

$$(\bar{\mathbf{y}})_i = \begin{cases} x_i & \text{falls } x_i \in [l_i, u_i], \\ u_i & \text{falls } x_i > u_i, \\ l_i & \text{falls } x_i < l_i. \end{cases}$$

□

**Definition 5.2 Stationärer Punkt.** Der Punkt  $\mathbf{x}_0$  wird stationärer Punkt des Problems (5.1) genannt, falls für alle Punkte des Tangentenkegels  $\mathbf{y} \in T(\mathbf{x}_0)$  die Ungleichung

$$\mathbf{y}^T \nabla f(\mathbf{x}_0) \geq 0$$

gilt.

□

Ein stationärer Punkt erfüllt also die im Satz 4.8 bewiesene notwendige Bedingung für ein lokales Minimum bezüglich  $\Omega$ .

### Algorithmus 5.3 Projektionsverfahren.

1. *Initialisierung.*

Bestimme  $\mathbf{x}^{(0)} \in \Omega$  und wähle drei reelle Parameter  $0 < \beta$ ,  $\mu < 1$  und  $\gamma > 0$ .

2. *Iteration.*  $\mathbf{k} = 0, 1, 2, \dots$

Falls  $\mathbf{x}^{(\mathbf{k})}$  ein stationärer Punkt ist, dann

Stopp

sonst

Betrachte für  $\alpha > 0$  den Pfad

$$\mathbf{x}^{(\mathbf{k})}(\alpha) := P_{\Omega}(\mathbf{x}^{(\mathbf{k})} + \alpha \nabla f(\mathbf{x}^{(\mathbf{k})}))$$

Setze

$$\mathbf{x}^{(\mathbf{k}+1)}(\alpha) := \mathbf{x}^{(\mathbf{k})}(\alpha^{(\mathbf{k})}),$$

wobei  $\alpha^{(\mathbf{k})} = \gamma \beta^{m^{(\mathbf{k})}}$  und  $m^{(\mathbf{k})}$  die kleinste natürliche Zahl größer oder gleich Null ist mit

$$\mathbf{f}(\mathbf{x}^{(\mathbf{k}+1)}) \leq \mathbf{f}(\mathbf{x}^{(\mathbf{k})}) + \mu \nabla \mathbf{f}(\mathbf{x}^{(\mathbf{k})})^T (\mathbf{x}^{(\mathbf{k}+1)} - \mathbf{x}^{(\mathbf{k})})$$

(Armijo-Liniensuche)

□

**Bemerkung 5.4 Armijo-Liniensuche.** Seien  $f(\mathbf{x})$  eine zu minimierende Funktion,  $\mathbf{x}^{(\mathbf{k})}$  die gegenwärtige Iterierte,  $\mathbf{s}^{(\mathbf{k})}$  eine irgendwie berechnete Abstiegsrichtung,  $\mu \in (0, 1)$  und  $\gamma > 0$  eine Konstante. Im Falle, dass man keine Nebenbedingungen hat, wählt man  $\lambda^{(0)} \geq \gamma \|\nabla f(\mathbf{x}^{(0)})\|_2$  und bestimme unter den Zahlen  $\lambda^{(j)} = 2^{-j} \lambda^{(0)}$ ,  $\beta = 1/2$ , die erste, für die

$$f(\mathbf{x}^{(\mathbf{k})} + \lambda^{(j)} \mathbf{s}^{(\mathbf{k})}) \leq f(\mathbf{x}^{(\mathbf{k})}) + \mu \lambda^{(j)} \nabla f(\mathbf{x}^{(\mathbf{k})})^T \mathbf{s}^{(\mathbf{k})}$$

gilt. Die Motivation für diese Herangehensweise kommt von der nach dem linearen Glied abgebrochenen Taylor-Entwicklung

$$f(\mathbf{x}^{(k)} + \lambda^{(j)} \mathbf{s}^{(k)}) \approx f(\mathbf{x}^{(k)}) + \lambda^{(j)} \nabla f(\mathbf{x}^{(k)})^T \mathbf{s}^{(k)}.$$

Ist  $\mathbf{s}^{(k)}$  eine Abstiegsrichtung, dann findet man in dieser Richtung einen Funktionswert von  $f(\mathbf{x})$ , der kleiner als der Funktionswert  $f(\mathbf{x}^{(k)})$  ist, falls  $\lambda^{(j)}$  nur hinreichend klein ist. Man fängt mit irgendeinem hinreichend großen  $\lambda^{(0)}$  an. Dieses hängt von  $\|\nabla f(\mathbf{x}^{(k)})\|_2$  ab. Ist  $\|\nabla f(\mathbf{x}^{(k)})\|_2$  klein, kann man vermuten in der Nähe eines Minimums zu sein (besitzt  $f(\mathbf{x})$  in  $\mathbf{x}^{(k)}$  ein Minimum, so ist  $\nabla f(\mathbf{x}^{(k)}) = \mathbf{0}$ ) und man braucht vielleicht nur einen kleinen Schritt. Man testet ob der Funktionswert sich verkleinert. Ist das nicht der Fall, wird der Parameter  $\lambda^{(j)}$  sukzessive halbiert, bis er klein genug ist.  $\square$

Im Algorithmus 5.3 hat man einen gekrümmten Pfad  $\mathbf{x}^{(k)}(\alpha)$ . Für jedes  $\alpha$  hat man im allgemeinen ein konvexes, quadratisches Minimierungsproblem über  $\Omega$  zu lösen. Das ist recht teuer.

Falls  $\Omega$  ein Polyeder ist, kann man eine Startiterierte  $\mathbf{x}^{(0)}$  mittels eines linearen Programms bestimmen.

Da der Algorithmus 5.3 im wesentlichen ein Gradientenverfahren ist, kann man im allgemeinen nur langsame Konvergenz erwarten. Außerdem ist die Berechnung von  $P_\Omega(\mathbf{x}^{(k)} + \alpha \nabla f(\mathbf{x}^{(k)}))$  für allgemeine Polyeder aufwendig. Wir betrachten jetzt noch eine Variante von Algorithmus 5.3, die zusätzlich leichter berechenbare Zwischenwerte  $\mathbf{x}^{(k)} \in \Omega$  berechnet, bei denen der Funktionswert zumindest nicht ansteigt.

### Algorithmus 5.5 Modifiziertes Projektionsverfahren.

1. *Initialisierung.*

Bestimme  $\mathbf{x}^{(0)} \in \Omega$  und wähle Parameter falls nötig.

2. *Iteration.*  $k = 0, 1, 2, \dots$

Bestimme  $\mathbf{x}^{(k+1)} = P_\Omega(\mathbf{x}^{(k)} + \alpha \nabla f(\mathbf{x}^{(k)}))$  entweder wie im Algorithmus 5.3 oder

bestimme  $\mathbf{x}^{(k+1)} \in \Omega$ , so dass  $f(\mathbf{x}^{(k+1)}) \leq f(\mathbf{x}^{(k)})$ .

$\square$

Die Umsetzung der zweiten Strategie hängt von der Art der Nebenbedingungen ab. Wir betrachten affine Nebenbedingungen (5.2). Bezeichne  $\hat{\mathbf{a}}_i$  die Zeilen von  $A$ . Für  $\mathbf{x} \in \Omega$  sei die Menge der aktiven Nebenbedingungen

$$I(\mathbf{x}) = \{i \in \{1, \dots, m\} : \hat{\mathbf{a}}_i^T \mathbf{x} = b_i\}.$$

Dann wird die zweite Strategie häufig so realisiert, dass  $I(\mathbf{x}^{(k)}) \subseteq I(\mathbf{x}^{(k+1)})$  gilt. Man wählt dazu in  $\mathbf{x}^{(k)}$  eine Abstiegsrichtung

$$\mathbf{v}^{(k)} \in L(\mathbf{x}^{(k)}) := \{\mathbf{v} : A_{I(\mathbf{x}^{(k)})} \mathbf{v} = \mathbf{0}\},$$

wobei  $A_{I(\mathbf{x}^{(k)})}$  die Teilmatrix von  $A$  mit den Zeilen ist, deren Indizes in  $I(\mathbf{x}^{(k)})$  enthalten sind. Die Abstiegseigenschaft wird dann nur in schwacher Form

$$\nabla f(\mathbf{x}^{(k)})^T \mathbf{v}^{(k)} \leq 0$$

verlangt. Die neue Iterierte besitzt die Gestalt  $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha \mathbf{v}^{(k)}$  für ein geeignetes  $\alpha$ . Aus der Wahl von  $\mathbf{v}^{(k)}$  folgt

$$A(\mathbf{x}^{(k+1)}) = A(\mathbf{x}^{(k)} + \alpha \mathbf{v}^{(k)}) = A\mathbf{x}^{(k)} + \alpha A\mathbf{v}^{(k)}.$$

Für die aktiven Nebenbedingungen verschwindet der zweite Summand. Für die anderen Nebenbedingungen,  $i \notin I(\mathbf{x}^{(k)})$ , ist der erste Summand kleiner als  $b_i$  und man findet ein hinreichend kleines  $\alpha > 0$ , so dass die Summe der beiden Summanden kleiner oder gleich  $b_i$  bleibt. Mit

$$\bar{\alpha}^{(k)} := \sup\{\alpha : \mathbf{x}^{(k)} + \alpha \mathbf{v}^{(k)} \in \Omega\}$$

wird nun eine Liniensuche gestartet um einen Faktor  $\alpha^{(k)}$  und damit ein Argument  $\mathbf{x}^{(k+1)}$  zu finden, so dass  $f(\mathbf{x}^{(k+1)}) \leq f(\mathbf{x}^{(k)})$  gilt. Ist  $\alpha^{(k)} < \bar{\alpha}^{(k)}$ , dann ist  $I(\mathbf{x}^{(k)}) = I(\mathbf{x}^{(k+1)})$ , sonst  $I(\mathbf{x}^{(k)}) \subset I(\mathbf{x}^{(k+1)})$ .

## 5.2 Penalty–Verfahren (Strafverfahren)

Wir betrachten wieder das Optimierungsproblem

$$z = \min\{f(\mathbf{x}) : \mathbf{x} \in \Omega\}, \quad (5.4)$$

diesmal aber zunächst mit  $f \in C^0(\mathbb{R}^n)$  und die Menge  $\Omega \subset \mathbb{R}^n$  sei abgeschlossen. Um die Lösung von (5.4) mit einer Folge einfacherer Optimierungsprobleme ohne Nebenbedingungen zu approximieren, wird die Straffunktion

$$l : \mathbb{R}^n \rightarrow \mathbb{R}_+, \quad l(\mathbf{x}) \begin{cases} > 0 & \text{für } \mathbf{x} \notin \Omega, \\ = 0 & \text{für } \mathbf{x} \in \Omega \end{cases}$$

eingeführt. Diese Funktion bestraft Punkte, die nicht zum zulässigen Bereich gehören, mit positiven Funktionswerten.

**Beispiel 5.6** Für

$$\Omega = \{\mathbf{x} \in \mathbb{R}^n : g_i(\mathbf{x}) \leq 0, i = 1, \dots, p; g_i(\mathbf{x}) = 0, i = p + 1, \dots, m\}$$

ist eine mögliche Straffunktion

$$l(\mathbf{x}) = \sum_{i=1}^p (g_i^+(\mathbf{x}))^\alpha + \sum_{i=p+1}^m |g_i(\mathbf{x})|^\alpha$$

mit  $\alpha > 0$ ,  $g_i^+(\mathbf{x}) = \max\{0, g_i(\mathbf{x})\}$ . □

**Definition 5.7 Penalty–Funktion** Die gewichtete Summe aus Zielfunktion und Straffunktion

$$p(\mathbf{x}, r) := f(\mathbf{x}) + rl(\mathbf{x}), \quad r \in \mathbb{R}_+, r > 0,$$

wird Penalty–Funktion genannt. Der Parameter  $r$  heißt Strafparameter. □

Für fest gewählte Parameter  $r$  werden jetzt die Minimierungsprobleme ohne Nebenbedingungen

$$\min_{\mathbf{x} \in \mathbb{R}^n} p(\mathbf{x}, r) \quad (5.5)$$

betrachtet. Der Strafterm belegt die Punkte, die nicht zum zulässigen Bereich gehören, mit positiven Werten, die für große Strafparameter  $r$  groß sind. Deswegen hofft man, dass die Minima von (5.5) für große Strafparameter im zulässigen Bereich liegen.

**Algorithmus 5.8 Allgemeines Penalty–Verfahren.**

1. *Initialisierung.*

Wähle  $r^{(1)} > 0$ .

2. *Iteration.*  $k = 0, 1, 2, \dots$

Bestimme ein lokales Minimum  $\mathbf{x}^{(k)}$  für  $p(\mathbf{x}, r^{(k)})$   
 Falls  $\mathbf{x}^{(k)} \in \Omega$ , dann Stopp  
 sonst wähle  $r^{(k+1)} \geq 2r^{(k)}$

□

Man kann zeigen, dass die  $\mathbf{x}^{(k)}$  unter gewissen Voraussetzungen tatsächlich Näherungen eines lokalen Minimums von (5.4) sind.

**Satz 5.9** *Seien  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  eine stetige Funktion,  $\mathbf{x}_0$  ein striktes lokales Minimum und  $l : \mathbb{R}^n \rightarrow \mathbb{R}_+$  eine stetige Straffunktion. Dann gibt es ein  $r_0 > 0$  so, dass für alle  $r > r_0$  die Penalty-Funktion  $p(\mathbf{x}, r)$  ein lokales Minimum  $\mathbf{x}(r)$  besitzt, dass für  $r \rightarrow \infty$  gegen  $\mathbf{x}_0$  konvergiert.*

**Beweis:** Literatur, [JS04, S. 294]. ■

**Bemerkung 5.10** Zwei Eigenschaften sind für Penalty-Verfahren von Bedeutung:

1. In vielen Fällen ist die Zielfunktion  $f(\mathbf{x})$  differenzierbar. Damit die Anwendung eines Verfahrens vom Newton-Typ zur Bestimmung des Minimums von (5.5) möglich ist, muss die Straffunktion  $l(\mathbf{x})$  auch differenzierbar sein.
2. Damit das Verfahren nach endlich vielen Schritten abbricht, ist es wünschenswert, wenn es bereits einen endlichen Wert  $\bar{r} > 0$  gibt, so dass ein lokales Minimum  $\mathbf{x}_0$  von (5.4) auch lokales Minimum für jedes Problem ohne Nebenbedingungen (5.5) mit  $r \geq \bar{r}$  ist. In diesem Fall nennt man die Penalty-Funktion exakt in  $\mathbf{x}_0$ .

Es stellt sich leider heraus, dass diese beiden wünschenswerten Eigenschaften in der Regel unvereinbar sind. Aus diesem Grunde werden Penalty-Verfahren in der Form von Algorithmus 5.8 praktisch nicht genutzt. Stattdessen betrachtet man modifizierte Penalty-Funktionen, die auf dem Konzept einer erweiterten Lagrange-Funktion (augmented Lagrange-Funktion) beruhen, siehe Literatur. □

## 5.3 Barrieremethoden

Wir betrachten das Problem

$$z = \min_{\mathbf{x} \in \Omega} f(\mathbf{x}) \tag{5.6}$$

mit den Nebenbedingungen

$$g_i(\mathbf{x}) \leq 0 \text{ für } 1 \leq i \leq p, \quad g_i(\mathbf{x}) = 0 \text{ für } p+1 \leq i \leq m. \tag{5.7}$$

Dabei seien  $f, g_i \in C^2(\mathbb{R}^n)$ ,  $i = 1, \dots, m$ , und wir nehmen an, dass (5.6) eine Optimallösung besitzt, die mit  $\mathbf{x}_0$  bezeichnet wird.

Barrieremethoden sind eng verwandt mit den Penalty-Verfahren. Auch bei diesen Methoden betrachtet man eine Folge von Hilfsproblemen, bei denen die Zielfunktion  $f(\mathbf{x})$  durch gewichtete Strafterme erweitert wird. Die Barrieremethoden erzeugen eine Folge von inneren Punkten, das heißt von Punkten welche die Ungleichungsrestriktionen sogar strikt erfüllen,  $g_i(\mathbf{x}) < 0, i = 1, \dots, p$ , während die Gleichungsrestriktionen verletzt sein können.

Bezeichne

$$\begin{aligned} \hat{\Omega} &:= \{\mathbf{x} \in \mathbb{R}^n : g_i(\mathbf{x}) \leq 0, i = 1, \dots, p\}, \\ \hat{\Omega}_0 &:= \{\mathbf{x} \in \mathbb{R}^n : g_i(\mathbf{x}) < 0, i = 1, \dots, p\}. \end{aligned}$$

Man beachte, die Menge  $\hat{\Omega}_0$  muss nicht notwendig die topologischen inneren Punkte von  $\hat{\Omega}$  enthalten, wähle zum Beispiel  $n = p = 1, g_1(x) = 0$ . Dann sind  $\hat{\Omega} = \mathbb{R}$  und  $\hat{\Omega}_0 = \emptyset$ .

Barriereverfahren bestrafen solche Punkte aus  $\hat{\Omega}_0$ , die sich dem Rand von  $\hat{\Omega}_0$  nähern. Die Gleichheitsnebenbedingungen werden direkt mit Hilfe von Linearisierungen behandelt. Diese Gleichheitsnebenbedingungen sind grundsätzlich einfacher zu behandeln als Ungleichungsnebenbedingungen. Die Strafterme in den Barriereverfahren, die sogenannten Barriereterme, sind in  $\hat{\Omega}_0$  endlich und wachsen zum Rand dieser Menge nach unendlich an. Außerhalb von  $\hat{\Omega}$  besitzen sie den Wert  $\infty$ . Im Gegensatz zu den Penalty-Verfahren, bei denen die Strafterme sukzessive immer stärker gewichtet werden, siehe Algorithmus 5.8, muss bei den Barriere-Verfahren der Einfluss der Strafterme immer weiter abgeschwächt werden. Damit wird das Gewicht der Zielfunktion im Barriereproblem erhöht und man kann hoffen, dass die Minima der Barriereprobleme unter geeigneten Voraussetzungen gegen ein Minimum von  $f(\mathbf{x})$  konvergieren. An der Eigenschaft, dass die Barriereterme außerhalb von  $\hat{\Omega}$  unendlich sind, ändert man nichts. Somit ist garantiert, dass die Minima der Barriereprobleme immer in  $\hat{\Omega}_0$  liegen.

**Definition 5.11 Skalare Barrierefunktion.** Eine skalare Barrierefunktion ist eine streng monoton fallende, glatte, konvexe Funktion  $b : (0, \infty) \rightarrow \mathbb{R}$  mit

$$\lim_{t \rightarrow 0+0} b(t) = \infty \quad \text{und} \quad \lim_{t \rightarrow 0+0} b'(t) = -\infty.$$

□

Außerdem wird stets  $b(t) = \infty$  für  $t \leq 0$  gesetzt, so dass  $b(t)$  formal eine auf  $\mathbb{R}$  definierte konvexe Funktion ist,  $b : \mathbb{R} \rightarrow \mathbb{R} \cup \{\infty\}$ .

**Beispiel 5.12** Beispiele für Barrierefunktionen sind

$$b(t) = -\log t, \quad b(t) = \frac{1}{t^\alpha}, \quad \alpha > 0.$$

Die logarithmische Barrierefunktion ist in gewisser Hinsicht optimal. □

Zur Konstruktion von Barriereverfahren zur Lösung von Problem (5.6) mit den Nebenbedingungen (5.7) werden nun Hilfsprobleme der Form

$$\inf_{\mathbf{x} \in \mathbb{R}^n} \left\{ f(\mathbf{x}) + \mu \sum_{i=1}^p b(d_i - g_i(\mathbf{x})) : g_i(\mathbf{x}) = 0, i = p+1, \dots, m \right\} \quad (5.8)$$

betrachtet. In (5.8) ist  $\mu > 0$  ein Gewicht für die Barriereterme und die Zahlen  $d_i \geq 0, i = 1, \dots, p$ , sind Verschiebungen der Ungleichungsnebenbedingungen in (5.7), das heisst anstatt  $g_i(\mathbf{x}) \leq 0$  ist nun  $g_i(\mathbf{x}) \leq d_i$  erlaubt. Diese Verschiebungen gestatten es, dass man das Verfahren auch dann anwenden kann, wenn kein innerer Punkt für (5.6), (5.7) bekannt ist. Die Zielfunktion von (5.8) wird abkürzend mit

$$\Phi(\mathbf{x}; \mu, \mathbf{d}) = f(\mathbf{x}) + \mu \sum_{i=1}^p b(d_i - g_i(\mathbf{x}))$$

bezeichnet.

Wir nehmen an, dass (5.8) ein endliches lokales Minimum besitzt. Die gewichtete Summe der Barriereterme in der Zielfunktion garantiert, dass jedes  $\mathbf{x}$  mit  $\Phi(\mathbf{x}; \mu, \mathbf{d}) \in \mathbb{R}$  die abgeschwächten Nebenbedingungen  $g_i(\mathbf{x}) < d_i, i = 1, \dots, p$ , erfüllt.

**Lemma 5.13** Falls  $f(\mathbf{x})$  und  $g_i(\mathbf{x}), i = 1, \dots, p$  konvex sind, so ist auch  $\Phi(\mathbf{x}; \mu, \mathbf{d})$  konvex.

**Beweis:** Es ist bekannt, dass die Linearkombination konvexer Funktionen mit nichtnegativen Koeffizienten in der Linearkombination eine konvexe Funktion ist. Damit bleibt zu zeigen, dass die Funktionen  $b(d_i - g_i(\mathbf{x}))$ ,  $i = 1, \dots, p$ , konvex sind.

Da die Funktionen,  $g_i(\mathbf{x})$  konvex sind, gilt für  $\lambda \in [0, 1]$

$$\begin{aligned} d_i - g_i(\lambda \mathbf{x}_1 + (1 - \lambda) \mathbf{x}_2) &\geq d_i - (\lambda g_i(\mathbf{x}_1) + (1 - \lambda) g_i(\mathbf{x}_2)) \\ &= \lambda (d_i - g_i(\mathbf{x}_1)) + (1 - \lambda) (d_i - g_i(\mathbf{x}_2)). \end{aligned}$$

Mit dieser Aussage, mit der Monotonie von  $b(t)$  und der Konvexität von  $b(t)$  folgt

$$\begin{aligned} b(d_i - g_i(\lambda \mathbf{x}_1 + (1 - \lambda) \mathbf{x}_2)) &\leq b(\lambda (d_i - g_i(\mathbf{x}_1)) + (1 - \lambda) (d_i - g_i(\mathbf{x}_2))) \\ &\leq \lambda b(d_i - g_i(\mathbf{x}_1)) + (1 - \lambda) b(d_i - g_i(\mathbf{x}_2)). \end{aligned}$$

■

Weiterhin gilt folgende stärkere Aussage.

**Satz 5.14** *Gelten die Voraussetzungen von Lemma 5.13 sowie  $\lim_{t \rightarrow \infty} b'(t) = 0$ . Die Gleichheitsnebenbedingungen  $g_i(\mathbf{x})$ ,  $i = p + 1, \dots, m$  seien affin und die Menge der Optimallösungen von (5.6) sei nicht leer und beschränkt. Dann besitzt das Hilfsproblem (5.8) für jedes  $\mu > 0$  eine Optimallösung und die Minima der Barriereprobleme nähern sich der Optimalmenge von (5.6).*

**Beweis:** Siehe Literatur. ■

Für Probleme (5.6), die die Bedingungen dieses Satzes erfüllen, kann man nun das folgende Verfahren konstruieren.

**Algorithmus 5.15 Barrieremethode für konvexe Probleme.**

1. *Initialisierung.*

Bestimme  $\mathbf{x}^{(0)} \in \mathbb{R}^n$  mit  $g_i(\mathbf{x}^{(0)}) = 0$  für  $i = p + 1, \dots, m$ . Wähle  $\mu^{(0)} > 0$  und  $\mathbf{d}^{(0)} \geq \mathbf{0}$  so dass  $d_i^{(0)} > g_i(\mathbf{x}^{(0)})$  für  $i = 1, \dots, p$ .

2. *Iteration.*  $k = 1, 2, \dots$

Wähle  $\lambda^{(k)} \in (0, 1)$  so, dass mit  $(\mu^{(k)}, \mathbf{d}^{(k)}) := \lambda^{(k)} (\mu^{(k-1)}, \mathbf{d}^{(k-1)})$  gilt

$$g_i(\mathbf{x}^{(k-1)}) < d_i^{(k)}, \quad \text{für } i = 1, \dots, p.$$

Ausgehend von  $\mathbf{x}^{(k-1)}$  führt man nun einige Schritte des Newton-Verfahrens (mit Liniensuche) zum lösen des Barriereproblems aus. Das Ergebnis ist  $\mathbf{x}^{(k)}$ .

□

Im ersten Schritt der Iteration werden sowohl das Gewicht als auch der Verschiebevektor verkleinert. Der Verkleinerungsfaktor wird so gewählt, dass mit dem neuen Verschiebevektor noch alle Ungleichungsnebenbedingungen erfüllt sind. Das Minimum  $\mathbf{x}_0(\mu^{(k)}, \mathbf{d}^{(k)})$  des Barriereproblems (5.8) zu den Parametern  $(\mu^{(k)}, \mathbf{d}^{(k)})$  wird im zweiten Schritt approximiert. Da die Barriereterme das Minimum vom Rand der Menge  $\{\mathbf{x} : g_i(\mathbf{x}) \leq d_i^{(k)}\}$  abstoßen, kann man nach der Berechnung der Näherung  $\mathbf{x}^{(k)}$  von  $\mathbf{x}_0(\mu^{(k)}, \mathbf{d}^{(k)})$  die Verschiebeparameter  $d_i^{(k)}$  in der folgenden Iteration wieder etwas verkleinern.

Die Schwierigkeiten von Algorithmus 5.15 bestehen darin, dass das Newton-Verfahren für  $\mu^{(k)} \rightarrow 0$  oft schlecht konvergiert. Deshalb wird diese Basiserangeheungsweise nicht genutzt. Man kann diese Herangeheungsweise durch Verfeinerung der Barrieremethode verbessern.

## 5.4 SQP–Verfahren

In diesem Abschnitt wird ein Zugang vorgestellt, der Punkte berechnet, die die notwendige Optimalitätsbedingung, die im Satz 4.21 (Kuhn/Tucker) formuliert ist, erfüllen, die sogenannten SQP–Verfahren (sequential quadratic programming). Es wird also eine Iteration durchgeführt, bei welcher in jedem Schritt ein quadratisches Optimierungsproblem gelöst wird.

Wir betrachten wieder das Optimierungsproblem (5.6) mit den Nebenbedingungen (5.7) und den gleichen Regularitätsvoraussetzungen wie im Abschnitt 5.3. Seien  $\mathbf{x}_0$  ein lokales Minimum von (5.6), (5.7) und  $\mathbf{z}_0$  der zugehörige Lagrange–Multiplikator zum Lagrange–Problem (4.10). Insgesamt erfüllen  $(\mathbf{x}_0, \mathbf{z}_0)$  das Problem, siehe (4.10),

$$\Phi(\mathbf{x}_0, \mathbf{z}_0) = \begin{pmatrix} \nabla_{\mathbf{x}} L(\mathbf{x}_0, \mathbf{z}_0) \\ \mathbf{z}_0^T \mathbf{g}(\mathbf{x}_0) \end{pmatrix} = \begin{pmatrix} \nabla f(\mathbf{x}_0) + (\nabla \mathbf{g}(\mathbf{x}_0))^T \mathbf{z}_0 \\ \mathbf{z}_0^T \mathbf{g}(\mathbf{x}_0) \end{pmatrix} = \mathbf{0} \quad (5.9)$$

mit  $\mathbf{z}_0 \geq \mathbf{0}$ . Da die Gleichheitsbedingungen ohnehin verschwinden, kann man (5.9) sogar wie folgt schreiben

$$\Phi(\mathbf{x}_0, \mathbf{z}_0) = \begin{pmatrix} \nabla f(\mathbf{x}_0) + (\nabla \mathbf{g}(\mathbf{x}_0))^T \mathbf{z}_0 \\ z_{0,1} g_1(\mathbf{x}_0) \\ \vdots \\ z_{0,p} g_p(\mathbf{x}_0) \\ g_{p+1}(\mathbf{x}_0) \\ \vdots \\ g_m(\mathbf{x}_0) \end{pmatrix} = \mathbf{0}. \quad (5.10)$$

SQP–Verfahren wollen das nichtlineare Problem (5.10) mit einem Verfahren vom Newton–Typ lösen. Dazu benötigt man die Jacobi–Matrix von  $\Phi(\mathbf{x}, \mathbf{z})$ , die durch

$$\begin{aligned} & \Psi(\mathbf{x}, \mathbf{z}, H_{L,\mathbf{x}}(\mathbf{x}, \mathbf{z})) \\ &= \begin{pmatrix} H_{L,\mathbf{x}}(\mathbf{x}, \mathbf{z}) & \nabla g_1(\mathbf{x}) & \cdots & \nabla g_p(\mathbf{x}) & \nabla g_{p+1}(\mathbf{x}) & \cdots & \nabla g_m(\mathbf{x}) \\ z_1 (\nabla g_1(\mathbf{x}))^T & g_1(\mathbf{x}) & & & & & \\ \vdots & & \ddots & & & & \\ z_p (\nabla g_p(\mathbf{x}))^T & & & g_p(\mathbf{x}) & & & 0 \\ (\nabla g_{p+1}(\mathbf{x}))^T & & & & & & \\ \vdots & & & & & & \\ (\nabla g_m(\mathbf{x}))^T & & & & 0 & & 0 \end{pmatrix} \\ & \in \mathbb{R}^{(n+m) \times (n+m)} \end{aligned}$$

gegeben ist. Ein wesentliches Merkmal eines SQP–Verfahrens besteht darin, dass die teure Hesse–Matrix  $H_{L,\mathbf{x}}(\mathbf{x}, \mathbf{z})$  in der Regel durch eine einfacher zu berechnende Matrix ersetzt wird.

Unter geeigneten Voraussetzungen kann man zeigen, dass die Jacobi–Matrix  $\Psi(\mathbf{x}_0, \mathbf{z}_0, H_{L,\mathbf{x}}(\mathbf{x}_0, \mathbf{z}_0))$  nichtsingulär ist und dass das Newton–Verfahren zur Nullstellenbestimmung von  $\Phi(\mathbf{x}, \mathbf{z})$  quadratisch konvergiert. Sei eine aktuelle Iterierte  $(\mathbf{x}^{(k)}, \mathbf{z}^{(k)})^T$  gegeben. Die Newton–Korrektur  $(\Delta \mathbf{x}^{(k)}, \Delta \mathbf{z}^{(k)})^T$  berechnet sich als Lösung des Gleichungssystems

$$\Psi(\mathbf{x}^{(k)}, \mathbf{z}^{(k)}, H_{L,\mathbf{x}}(\mathbf{x}^{(k)}, \mathbf{z}^{(k)})) \begin{pmatrix} \Delta \mathbf{x}^{(k)} \\ \Delta \mathbf{z}^{(k)} \end{pmatrix}^T = -\Phi(\mathbf{x}^{(k)}, \mathbf{z}^{(k)}).$$

Beim Newton–Verfahren kann man im allgemeinen jedoch nur lokale Konvergenz erwarten, das heißt, der Startwert muss nahe genug an der (unbekannten) Lösung



sein. Speziell für die Funktion  $\Phi(\mathbf{x}, \mathbf{z})$  kann man auch nicht garantieren, dass alle Iterierten die Ungleichungen  $g_i(\mathbf{x}^{(k)}) \leq 0$  und die Nichtnegativitätsbedingung  $z_i^{(k)} \geq 0$  erfüllen. Damit ist es möglich, dass das Newton-Verfahren gegen eine nicht zulässige Lösung von  $\Phi(\mathbf{x}, \mathbf{z}) = \mathbf{0}$  konvergiert, bei der Nebenbedingungen nicht erfüllt sind oder die Lagrange-Multiplikatoren negativ sind. Die Konvergenz gegen eine solche Lösung muss verhindert werden. Dazu wird anstelle des Newton-Verfahrens das System

$$\Psi\left(\mathbf{x}^{(k)}, \mathbf{z}^{(k+1)}, B^{(k)}\right)\left(\Delta\mathbf{x}^{(k)}, \Delta\mathbf{z}^{(k)}\right)^T = -\Phi\left(\mathbf{x}^{(k)}, \mathbf{z}^{(k)}\right) \quad (5.11)$$

betrachtet, wobei  $\Delta\mathbf{x}^{(k)}$ ,  $\Delta\mathbf{z}^{(k)}$  und  $\mathbf{z}^{(k+1)} = \mathbf{z}^{(k)} + \Delta\mathbf{z}^{(k)}$  die zusätzlichen Forderungen

$$z_i^{(k+1)} \geq 0 \quad \text{für } i = 1, \dots, p, \quad (5.12)$$

$$g_i\left(\mathbf{x}^{(k)}\right) + \nabla\left(g_i\left(\mathbf{x}^{(k)}\right)\right)^T \Delta\mathbf{x}^{(k)} \leq 0 \quad \text{für } i = 1, \dots, p \quad (5.13)$$

erfüllen. Im Vergleich zum Newton-Verfahren ersetzt man  $H_{L,\mathbf{x}}(\mathbf{x}^{(k)}, \mathbf{z}^{(k)})$  durch eine Matrix  $B^{(k)}$ , die in der Regel durch gewisse Quasi-Newton-Korrekturen (sogenannte Broyden-Verfahren) erzeugt wird. Des Weiteren wird der Vektor  $\mathbf{z}^{(k)}$  auf der linken Seite durch  $\mathbf{z}^{(k+1)}$  ersetzt. Man erhält ein implizites Gleichungssystem, welches nicht mehr linear bezüglich der Lagrange-Multiplikatoren ist. Außerdem werden noch die linearen Ungleichungsbedingungen (5.12), (5.13) an  $\Delta\mathbf{x}^{(k)}$  und  $\Delta\mathbf{z}^{(k)}$  gestellt.

Ausgeschrieben besagt (5.11) – (5.13)

$$\begin{aligned} \nabla f\left(\mathbf{x}^{(k)}\right) + B^{(k)} \Delta\mathbf{x}^{(k)} + \sum_{i=1}^m z_i^{(k+1)} \nabla g_i\left(\mathbf{x}^{(k)}\right) &= 0, \\ z_i^{(k+1)} \left( g_i\left(\mathbf{x}^{(k)}\right) + \left( \nabla g_i\left(\mathbf{x}^{(k)}\right) \right)^T \Delta\mathbf{x}^{(k)} \right) &= 0, \quad i = 1, \dots, p, \quad (5.14) \\ g_i\left(\mathbf{x}^{(k)}\right) + \left( \nabla g_i\left(\mathbf{x}^{(k)}\right) \right)^T \Delta\mathbf{x}^{(k)} &= 0, \quad i = p+1, \dots, m. \end{aligned}$$

Wir betrachten das folgende quadratische Programm

$$z = \min_{\mathbf{s} \in \mathbb{R}^n} \left( \nabla f\left(\mathbf{x}^{(k)}\right) \right)^T \mathbf{s} + \frac{1}{2} \mathbf{s}^T B^{(k)} \mathbf{s} \quad (5.15)$$

unter den Nebenbedingungen

$$g_i\left(\mathbf{x}^{(k)}\right) + \left( \nabla g_i\left(\mathbf{x}^{(k)}\right) \right)^T \mathbf{s} \leq 0, \quad i = 1, \dots, p, \quad (5.16)$$

$$g_i\left(\mathbf{x}^{(k)}\right) + \left( \nabla g_i\left(\mathbf{x}^{(k)}\right) \right)^T \mathbf{s} = 0, \quad i = p+1, \dots, m. \quad (5.17)$$

Erfülle dieses Problem die Voraussetzungen des Satzes 4.21 (Kuhn/Tucker). Dann sind die notwendigen Bedingungen für ein Minimum gerade die Gleichungen (5.14). *Übungsaufgabe* Damit ergibt sich folgendes Verfahren:

#### Algorithmus 5.16 SQP-Algorithmus (Grundform).

1. *Initialisierung.*

Wähle  $\mathbf{x}^{(0)} \in \mathbb{R}^n$ ,  $\mathbf{B}^{(0)} = (\mathbf{B}^{(0)})^T (\approx H_{L,\mathbf{x}}(\mathbf{x}^{(0)}, \mathbf{z}^{(0)}))$  für ein  $\mathbf{z}^{(0)} \in \mathbb{R}^m$  mit  $z_i^{(0)} > 0$ ,  $i = 1, \dots, p$

2. *Iteration.*  $k = 0, 1, 2, \dots$

Bestimme die Lösung  $\mathbf{s}$  von (5.15) – (5.17) und einen zugehörigen Lagrange-Multiplikator  $\mathbf{z}$ . Setze

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \mathbf{s}, \quad \mathbf{z}^{(k+1)} = \mathbf{z}.$$

Bestimme eine symmetrische Matrix

$$\mathbf{B}^{(k+1)} \approx \mathbf{H}_{L, \mathbf{x}} \left( \mathbf{x}^{(k+1)}, \mathbf{z}^{(k+1)} \right).$$

□

Falls  $B^{(k)}$  positiv semidefinit ist, dann ist (5.15) – (5.17) ein konvexes quadratisches Programm. In diesem Fall sind die Bedingungen des Satzes 4.21 notwendig und hinreichend für ein globales Minimum, siehe auch Satz 3.24. Zur Lösung kann man beispielsweise ein Projektionsverfahren nehmen, siehe Abschnitt 5.1.