

**Optimierung
für Wirtschaftsingenieure**
Lösungen der Übungsaufgaben

Prof. Dr. Martin Bünner

Skriptenreihe Wirtschaftsingenieurwesen

Optimierung für Wirtschaftsingenieure

Lösungen der Übungsaufgaben

Prof. Dr. Martin Bünner

Herausgeber: Lukas Schmid

Bei Fragen und Anregungen erreichen Sie
den Autor über <martin.buenner@ntb.ch>.

Prof. Dr. Martin Bünner, 2019
Optimierung
für Wirtschaftsingenieure
FHS St.Gallen

Copyright © 2019, Prof. Dr. Martin Bünner
Veröffentlicht durch: FHS St.Gallen
Layout: Tobias Weh (tobiw.de)

Für meine Söhne A. und M.

Inhaltsverzeichnis

1	Grundlagen und Wiederholung	9
2	Univariate Optimierung	17
3	Multivariate Optimierung	23
4	Numerische Mathematik, Algorithmen und das Newton-Verfahren	43
5	Das Simplex-Verfahren zur Lösung von LP-Problemen	49
6	Das SQP-Verfahren zur Lösung von NLP-Problemen	55
7	Relaxation und Enumeration zur Lösung von Integer- und mixed-integer Problemen	59
8	Ausblick	65

Vorwort

Dieses Buch enthält die Lösungen der Übungsaufgaben aus dem Buch *Optimierung für Wirtschaftsingenieure* von Martin Bünner.

Teilweise enthalten die Lösungen Matlab-Code (Version R2016). Dieser ist in weiten Teilen auch unter den lizenz-frei erhältlichen Entwicklungs- und Programmier-Umgebungen *Octave* und *SciLab* lauffähig.

Den in einigen Lösungen auftauchenden Matlab-Code habe ich versucht so weit wie möglich selbsterklärend zu formulieren, dass der Leser und die Leserin, diesen wie Pseudo-Code verwenden kann. In der Folge, lässt sich der Matlab-Code problemlos in jede beliebige andere Programmiersprache oder Entwicklungs-Umgebung einfach umsetzen.

St. Gallen im März 2019

Prof. Dr. Martin Bünner



Kapitel 1

Grundlagen und Wiederholung

1.1.5.1

Die Funktion $f(x)$ besitzt 2 lokale Minima.
Die Funktion $f(x)$ besitzt 1 lokale Maxima.
Die Funktion $f(x)$ besitzt 1 globale Minima.
Die Funktion $f(x)$ besitzt 1 globale Maxima.
Die Funktion $f(x)$ besitzt 1 Extremwerte mit einer horizontalen Tangente.
Die Funktion $f(x)$ besitzt 2 Extremwerte am Rand des Definitionsbereiches.
Die Funktion $f(x)$ besitzt 0 Extremwerte an einer Sprungstelle.
Die Funktion $f(x)$ besitzt 0 Extremwerte an einem „Knick“.
Die Funktion ist stetig. Die Funktion ist differenzierbar.

1.1.5.2

Die Funktion $f(x)$ besitzt 2 lokale Minima.
Die Funktion $f(x)$ besitzt 2 lokale Maxima.
Die Funktion $f(x)$ besitzt 1 globale Minima.
Die Funktion $f(x)$ besitzt 1 globale Maxima.
Die Funktion $f(x)$ besitzt 1 Extremwerte mit einer horizontalen Tangente.
Die Funktion $f(x)$ besitzt 2 Extremwerte am Rand des Definitionsbereiches.
Die Funktion $f(x)$ besitzt 1 Extremwerte an einer Sprungstelle.
Die Funktion $f(x)$ besitzt 0 Extremwerte an einem „Knick“.

1.1.5.3

Die Funktion $f(x)$ besitzt 1 lokale Minima.
Die Funktion $f(x)$ besitzt 2 lokale Maxima.
Die Funktion $f(x)$ besitzt 1 globale Minima.

Die Funktion $f(x)$ besitzt 1 globale Maxima.
Die Funktion $f(x)$ besitzt 1 Extremwerte mit einer horizontalen Tangente.
Die Funktion $f(x)$ besitzt 2 Extremwerte am Rand des Definitionsbereiches.
Die Funktion $f(x)$ besitzt 0 Extremwerte an einer Sprungstelle.
Die Funktion $f(x)$ besitzt 0 Extremwerte an einem „Knick“.
Die Funktion ist stetig. Die Funktion ist differenzierbar.

1.1.5.4

Die Funktion $f(x)$ besitzt 2 lokale Minima.
Die Funktion $f(x)$ besitzt 3 lokale Maxima.
Die Funktion $f(x)$ besitzt 2 globale Minima.
Die Funktion $f(x)$ besitzt 2 globale Maxima.
Die Funktion $f(x)$ besitzt 1 Extremwerte mit einer horizontalen Tangente.
Die Funktion $f(x)$ besitzt 2 Extremwerte am Rand des Definitionsbereiches.
Die Funktion $f(x)$ besitzt 0 Extremwerte an einer Sprungstelle.
Die Funktion $f(x)$ besitzt 2 Extremwerte an einem „Knick“.
Die Funktion ist stetig.

1.1.5.5

Die Funktion $f(x)$ besitzt 1 lokale Minima.
Die Funktion $f(x)$ besitzt 2 lokale Maxima.
Die Funktion $f(x)$ besitzt 1 globale Minima.
Die Funktion $f(x)$ besitzt 1 globale Maxima.
Die Funktion $f(x)$ besitzt 1 Extremwerte mit einer horizontalen Tangente.
Die Funktion $f(x)$ besitzt 2 Extremwerte am Rand des Definitionsbereiches.
Die Funktion $f(x)$ besitzt 0 Extremwerte an einer Sprungstelle.
Die Funktion $f(x)$ besitzt 0 Extremwerte an einem „Knick“.
Die Funktion ist stetig. Die Funktion ist differenzierbar.

1.1.5.6

Die Funktion $f(x)$ besitzt 1 lokale Minima.
Die Funktion $f(x)$ besitzt 2 lokale Maxima.
Die Funktion $f(x)$ besitzt 1 globale Minima.
Die Funktion $f(x)$ besitzt 2 globale Maxima.
Die Funktion $f(x)$ besitzt 1 Extremwerte mit einer horizontalen Tangente.
Die Funktion $f(x)$ besitzt 2 Extremwerte am Rand des Definitionsbereiches.
Die Funktion $f(x)$ besitzt 0 Extremwerte an einer Sprungstelle.
Die Funktion $f(x)$ besitzt 1 Extremwerte an einem „Knick“.
Die Funktion ist stetig.

1.1.5.7

Die Funktion $f(x)$ ist stetig für $a = -1$.

Die Funktion $f(x)$ ist differenzierbar für kein $a \in \mathbb{R}$.

Die Funktion $f(x)$ besitzt ein globales Maximum für $a \geq -1$.

Die Funktion $f(x)$ besitzt genau zwei globale Minima für $a = -1.5$.

1.1.5.8**1.**

$$\begin{aligned} f(x) &= 3x^4 - 2x^2 - \frac{1}{3}x - 1 \\ f'(x) &= 12x^3 - 4x - \frac{1}{3} \\ f''(x) &= 36x^2 - 4 \end{aligned}$$

Matlab:

```
x=(-1:0.01:1);
f=3*x.^4-2*x.^2-x/3-1;
fs=12*x.^3-4*x-1/3;
fss=36*x.^2-4;
figure(1);subplot(3,1,1);plot(x,f);xlabel('x');ylabel('f(x)');
subplot(3,1,2);plot(x,fs);xlabel('x');ylabel('1. Ableitung');
subplot(3,1,3);plot(x,fss);xlabel('x');ylabel('2. Ableitung');
```

2.

$$\begin{aligned} f(x) &= ax^2 - (a-2)x - a^2 \\ f'(x) &= 2ax - a + 2 \\ f''(x) &= 2a \end{aligned}$$

Matlab:

```
a=1;
x=(-2:0.01:1);
f=a*x.^2-(a-2)*x-a^2;
fs=2*a*x-a+2;
fss=2*a*ones(length(x));
figure(2);subplot(3,1,1);plot(x,f);xlabel('x');ylabel('f(x)');
subplot(3,1,2);plot(x,fs);xlabel('x');ylabel('1. Ableitung');
subplot(3,1,3);plot(x,fss);xlabel('x');ylabel('2. Ableitung');
```

3.

$$\begin{aligned}
 F(x) &= xe^{-x} \\
 F'(x) &= e^{-x} - xe^{-x} = (1-x)e^{-x} \\
 F''(x) &= -e^{-x} - (1-x)e^{-x} = (x-2)e^{-x}
 \end{aligned}$$

Matlab:

```

x=(-2:0.01:2);
F=x.* exp(-x);
Fs=(1-x).* exp(-x);
Fss=(x-2).* exp(-x);
figure(3);subplot(3,1,1);plot(x,F);xlabel('x');ylabel('F(x)');
subplot(3,1,2);plot(x,Fs);xlabel('x');ylabel('1. Ableitung');
subplot(3,1,3);plot(x,Fss);xlabel('x');ylabel('2. Ableitung');

```

4.

$$\begin{aligned}
 h(x) &= \sin(x)(\cos(x) - A) \\
 h'(x) &= \cos(x)(\cos(x) - A) + \sin(x)(-\sin(x)) = \cos^2(x) - \sin^2(x) - A \cos(x) \\
 h''(x) &= 2 \cos(x)(-\sin(x)) - 2 \sin(x) \cos(x) + A \sin(x) \\
 &= -4 \cos(x) \sin(x) + A \sin(x)
 \end{aligned}$$

Matlab:

```

A=2;
x=(-5:0.01:5);
h=sin(x).*(cos(x)-A);
hs=cos(x).^2-sin(x).^2-A*cos(x);
hss=-4*cos(x).*sin(x)+A*sin(x);
figure(4);subplot(3,1,1);plot(x,h);xlabel('x');ylabel('h(x)');
subplot(3,1,2);plot(x,hs);xlabel('x');ylabel('1. Ableitung');
subplot(3,1,3);plot(x,hss);xlabel('x');ylabel('2. Ableitung');

```

5.

$$\begin{aligned}
 s(a) &= \frac{2a}{a-1} \\
 s'(a) &= \frac{2(a-1) - 2a}{(a-1)^2} = \frac{-2}{(a-1)^2} \\
 s''(a) &= \frac{0 - (-4)(a-1)}{(a-1)^4} = \frac{4}{(a-1)^3}
 \end{aligned}$$

Matlab:

```

a_1=(0:0.01:1-0.01);
a_2=(1+0.01:0.01:2);
s_1=2*a_1./(a_1-1);s_2=2*a_2./(a_2-1);
s_abl1_1=-2./((a_1-1).^2);s_abl1_2=-2./((a_2-1).^2);
s_abl2_1=4./((a_1-1).^3);s_abl2_2=4./((a_2-1).^3);
figure(5);
subplot(3,1,1);hold off;plot(a_1,s_1);hold on;plot(a_2,s_2);
xlabel('a');ylabel('s(a)');
subplot(3,1,2);hold off;plot(a_1,s_abl1_1);hold on;plot(a_2,s_abl1_2);
xlabel('a');ylabel('1. Ableitung');
subplot(3,1,3);hold off;plot(a_1,s_abl2_1);hold on;plot(a_2,s_abl2_2);
xlabel('a');ylabel('2. Ableitung');

```

6.

$$\begin{aligned}
 y(x) &= x - \frac{k-x}{k+x} \\
 y'(x) &= 1 - \frac{-(k+x) - (k-x)}{(k+x)^2} = 1 + \frac{2k}{(k+x)^2} \\
 y''(x) &= \frac{-4k}{(k+x)^3}
 \end{aligned}$$

Matlab:

```

k=1;
x_1=(-3:0.01:-1-0.01);
x_2=(-1+0.01:0.01:1);
y_1=x_1-(k-x_1)./(k+x_1);y_2=x_2-(k-x_2)./(k+x_2);
ys_1=1+2*k./((k+x_1).^2);ys_2=1+2*k./((k+x_2).^2);
yss_1=-4*k./((k+x_1).^3);yss_2=-4*k./((k+x_2).^3);
figure(6);
subplot(3,1,1);hold off;plot(x_1,y_1);hold on;plot(x_2,y_2);
xlabel('x');ylabel('y(x)');
subplot(3,1,2);hold off;plot(x_1,ys_1);hold on;plot(x_2,ys_2);
xlabel('x');ylabel('1. Ableitung');
subplot(3,1,3);hold off;plot(x_1,yss_1);hold on;plot(x_2,yss_2);
xlabel('x');ylabel('2. Ableitung');

```

7.

$$\begin{aligned}
 z(t) &= 4 \sin\left(\omega t - \frac{\pi}{4}\right) \\
 \dot{z}(t) &= 4\omega \cos\left(\omega t - \frac{\pi}{4}\right) \\
 \ddot{z}(t) &= -4\omega^2 \sin\left(\omega t - \frac{\pi}{4}\right)
 \end{aligned}$$

1 Grundlagen und Wiederholung

Matlab:

```
w=2;
t=(-3:0.01:3);
z=4*sin(w*t-pi/4);
zs=4*w*cos(w*t-pi/4);
zss=4*w^2*sin(w*t-pi/4);
figure(7);subplot(3,1,1);plot(t,z);xlabel('t');ylabel('z(t)');
subplot(3,1,2);plot(t,zs);xlabel('t');ylabel('1. Ableitung');
subplot(3,1,3);plot(t,zss);xlabel('t');ylabel('2. Ableitung');
```

8.

$$\begin{aligned}u(x) &= \frac{1}{2}e^{-x^2} \\u'(x) &= -xe^{-x^2} \\u''(x) &= -e^{-x^2} + 2x^2e^{-x^2} = (2x^2 - 1)e^{-x^2}\end{aligned}$$

Matlab:

```
x=(-2:0.01:2);
u=0.5*exp(-x.^2);
us=-x.*exp(-x.^2);
uss=(2*x.^2-1).*exp(-x.^2);
figure(8);subplot(3,1,1);plot(x,u);xlabel('x');ylabel('u(x)');
subplot(3,1,2);plot(x,us);xlabel('x');ylabel('1. Ableitung');
subplot(3,1,3);plot(x,uss);xlabel('x');ylabel('2. Ableitung');
```

1.2.3.1

1.

$$\begin{aligned}f(x,y) &= 2xy - x - 1 \\ \nabla f(x,y) &= \begin{pmatrix} (2y-1) & 2x \end{pmatrix} \\ H(x,y) &= \begin{pmatrix} 0 & 2 \\ 2 & 0 \end{pmatrix}\end{aligned}$$

2.

$$\begin{aligned}
 f(x, y) &= ax^2 - y^2 \\
 \nabla f(x, y) &= \begin{pmatrix} 2ax & 2y \end{pmatrix} \\
 H(x, y) &= \begin{pmatrix} 2a & 0 \\ 0 & 2 \end{pmatrix}
 \end{aligned}$$

3.

$$\begin{aligned}
 f(x, y) &= x \cdot \sin(y) \\
 \nabla f(x, y) &= \begin{pmatrix} \sin(y) & x \cos(y) \end{pmatrix} \\
 H(x, y) &= \begin{pmatrix} 0 & \cos(y) \\ \cos(y) & -x \sin(y) \end{pmatrix}
 \end{aligned}$$

4.

$$\begin{aligned}
 f(x_1, x_2, x_3) &= -x_1 - \frac{x_2}{2x_3} \\
 \nabla f(x_1, x_2, x_3) &= \begin{pmatrix} -1 & -\frac{1}{2x_3} & \frac{x_2}{2x_3^2} \end{pmatrix} \\
 H(x_1, x_2, x_3) &= \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & \frac{1}{2x_3^2} \\ 0 & \frac{1}{2x_3^2} & -\frac{x_2}{x_3^3} \end{pmatrix}
 \end{aligned}$$

1.2.3.2

- 2 kritische Punkte: (0;0) ist ein Minimum und (2;0) ist keine Extremalstelle.
Matlab-Plot mit:

```

[X,Y]=meshgrid(-2:0.1:2,-2:0.1:2);
Z=(X.*X+Y.*Y).*exp(-X);
surf(X,Y,Z);

```

- 2 kritische Punkte: (1;2) und (1;-2) sind keine Extremalstellen, (0;0) ist ein Maximum und (2;0) ist ein Minimum.
- 1 kritischer Punkt: (3;-3) ist ein Maximum.
- 1 kritischer Punkt: (0;0) ist ein Minimum.
- 2 kritische Punkte: (0;0) ist keine Extremalstelle und (0.44;0.38) ist ein Minimum.

6. 1 kritischer Punkt: $(1; 1)$ ist ein Minimum.



Kapitel 2

Univariate Optimierung

2.3.1.1

1. reellwertiges univariates Optimierungsproblem ohne Nebenbedingungen.
2. Es gibt keine Lösung, da die Funktion für $x \rightarrow \infty$ gegen unendlich strebt.

2.3.1.2

1. reellwertiges univariates Optimierungsproblem mit 2 Ungleichheits-Nebenbedingungen.
2. Eine Lösung kann nur an einem kritischen Punkt mit waagrechtter Tangente entstehen oder am Rand der Feasible Region. Die Funktion besitzt ein lokales Maximum bei $x_1 = 2 - \frac{2}{3}\sqrt{6}$ und ein lokales Minimum bei $x_2 = 2 + \frac{2}{3}\sqrt{6}$. Der Funktionswert am lokalen Maximum ist $f(x_1) = -0.2907$. Die Funktionswerte an den Rändern sind: $f(-1) = -12$ und $f(1) = -2$. Somit ist die Lösung $x^* = 2 - \frac{2}{3}\sqrt{6}$ mit $f^* = -0.2907$.

2.4.1.1

1. univariates, reellwertiges Optimierungsproblem mit 2 Ungleichheitsnebenbedingungen
2. Plot mit Matlab nur in der Feasible Region:

```
x=(-2:0.1:4);  
y=exp(-x)+2*x.^4;  
plot(x,y);
```

Wir verwenden das Vergrößerungstool und betrachten die Funktion in einem geeigneten Bereich um aus Grafik das Minimum abzulesen: $x^* \approx 0.42$.

2.4.1.2

1. univariates, reellwertiges Optimierungsproblem mit 2 Ungleichheitsnebenbedingungen
2. Plot mit Matlab nur in der Feasible Region:

```
x=(1:0.1:4);
y=exp(-x)+2*x.^4;
plot(x,y);
```

Wir können das Minimum aus der Grafik ablesen: $x^* = 1$.

2.5.1.1

1. 2 Möglichkeiten:
 - (i)

$$\max_{v \in \mathbb{R}^+} K(v).$$

Reellwertiges Optimierungsproblem ohne Nebenbedingungen.

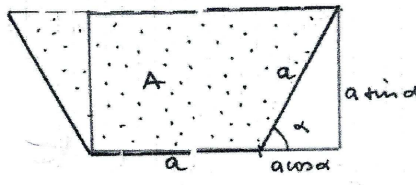
- (ii)

$$\max K(v), v \geq 0.$$

Reellwertiges Optimierungsproblem mit einer Nebenbedingung.

2. Die 1. Ableitung $K'(v) = 4 \frac{1-v^2}{(v^2+1)^2}$ besitzt eine Nullstelle bei $v = 1$. Dort wechselt $K'(v)$ das Vorzeichen von + nach -, deshalb handelt es sich um ein lokales Maximum. Dieses ist auch die Lösung. Maximale Bremskraft $K_{\max} = 2$

2.5.1.2



- 1.
2. ...

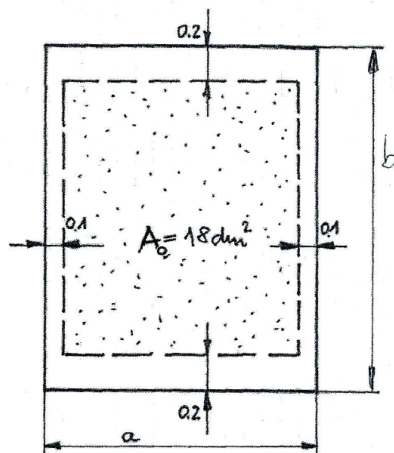
3. Wir bestimmen nun die Fläche A in Abhängigkeit von α und erhalten $A(\alpha) = a^2 \sin(\alpha)(1 + \cos(\alpha))$. Somit ergibt sich ein reellwertiges, univariates Optimierungsproblem mit 2 Nebenbedingungen:

$$A^* = \max \sin(\alpha)(1 + \cos(\alpha))$$

$$0 \leq \alpha \leq \frac{2\pi}{3}$$

4. Wir bestimmen das Maximum der Zielfunktion mit Hilfe der 1. Ableitung und erhalten $\alpha^* = 60^\circ$ und $A^* = \frac{3\sqrt{3}}{4}a^2$.

2.5.1.3



1.

2. ...

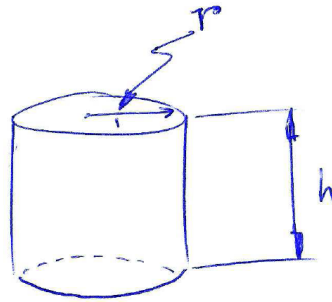
3. Es gilt: $(a - 0.2)(b - 0.4) = 18$. Dies können wir umformen zu: $a = \frac{18}{b - 0.4} + 0.2$. Es soll $A = a \cdot b$ maximal werden und somit erhalten wir ein reellwertiges, univariates Optimierungsproblem mit einer Nebenbedingung:

$$A^* = \max\left(\frac{18b}{b - 0.4} + 0.2b\right),$$

$$b \geq 0.4$$

4. Wir bestimmen mit Hilfe der 1. Ableitung das Maximum: Breite = 3.2 dm, Höhe = 6.4 dm und die Fläche ist $A^* = 20.48 \text{ dm}^2$.

2.5.1.4



- 1.
2. ...
3. Es gilt: $V = \pi r^2 h = 1000$ und damit können wir h durch r ausdrücken: $h = \frac{1000}{\pi r^2}$. Wir erhalten als Kosten für die Büchse $K(r) = \frac{4000}{r} + 6\pi r^2$. Somit erhalten wir ein reellwertiges, univariates Optimierungsproblem mit einer Nebenbedingung:

$$K^* = \max\left(\frac{4000}{r} + 6\pi r^2\right),$$

$$r \geq 0$$

4. Mit Hilfe der 1. Ableitung erhalten wir als Lösung $r^* = \frac{10}{\sqrt[3]{3}\pi} = 4.734$ dm, Kosten sind 12,67 SFr für 1 Büchse und 12'673'842 SFr für 1 Mio Büchsen.

2.6.3.1

1. univariates Integer-Optimierungs-Problem.
2. Unendlich viele Kandidaten. Rechenzeit wäre unendlich lange. Durch Probieren: $n^* = -100$.
3. Durch die Relaxation erhalten wir ein reellwertiges Optimierungsproblem, das sich mit Hilfe der 1. Ableitung lösen lässt: $x^* = -100$. Wir gehen zurück in den Integer-Zahlenraum und erhalten: $n^* = -100$. Für diese Aufgabe ist die Relaxation sehr viel effizienter als die Enumeration.

2.6.3.2

1. Integer-Optimierungs-Problem.

2. Es gibt 101 Kandidaten. Rechenzeit gering.
Matlab:

```
n=(0:100);
f=cos(10000*n-7.2);
figure(2);plot(f,'+');
format long;
[ fs,ns ]=min(f)
```

Lösung: $f^* = -0.999993110344365$ bei $n^* = 94$.

3. Die Cosinus-Funktion hat lokale Minima bei $n = 2\pi * 10^{-4} * k, k \in \mathbb{Z}$. Aber: Diese sind so zahlreich und vergleichsweise weit weg von den Werten von n , dass die Relaxation keinen Vorteil bringt. Bei dieser Aufgabe ist die Enumeration sehr viel effizienter als die Relaxation.

2.6.3.3

1. Die Kostenfunktion ist (s. Aufgabe zuvor): $K(r) = \frac{4000}{r} + 6\pi r^2$. Somit erhalten wir eine univariate, integer Optimierungsaufgabe:

$$K^* = \min K(r_i),$$

$$r_i \in \{3.0; 4.1; \dots; 7.0\}$$

2. Es müssen bei der Enumeration nur 8 Kandidaten probiert werden.
Matlab:

```
r=[ 3 4.1 4.42 4.88 5.3 5.56 6 7 ];
K=40./r +0.06*r.*r*pi;
[ Ks,is ]=min(K);
rs=r(is)
Ks
```

3. Bei der Relaxation erhalten wir die reelwertige Lösung: $r = 4,734$ dm. Die beiden diskreten r in der Nähe sind: $r_3 = 4.42$ dm und $r_4 = 4.88$ dm. Wir testen beide Kandidaten und erhalten als Lösung: $r^* = r_4 = 4.88$ dm

2.7.1

wahr sind 2, 4, 7, 8, 10.

2.7.2

wahr sind 1, 5, 10.



Kapitel 3

Multivariate Optimierung

3.1.1.1

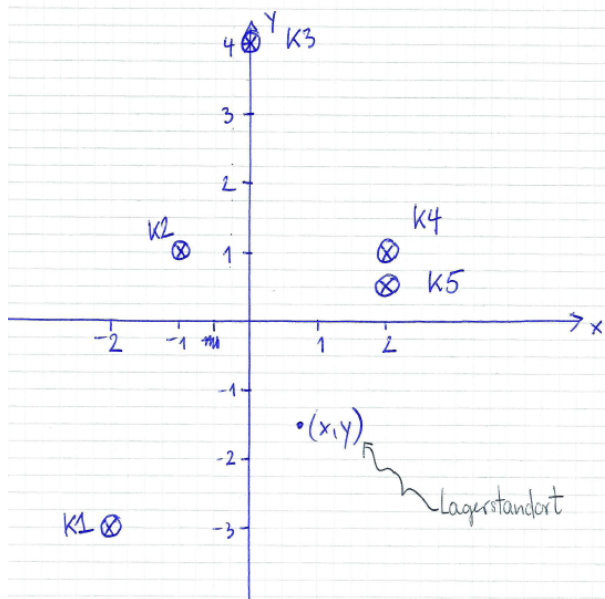
1. reelwertige Optimierungsaufgabe ohne Nebenbedingungen mit zwei Design Variablen.
2. Die Optimierungsaufgabe besitzt unendlich viele Lösungen: $x = 2n, n \in \mathbb{Z}$, $y = 2m, m \in \mathbb{Z}$.
3. Matlab:

```
[ X,Y ] = meshgrid(-3:.1:3, -3:.1:3);  
Z=-cos(pi*X)-cos(pi*Y);  
figure(3);surf(X,Y,Z)
```

3.1.1.2

1. reelwertige Optimierungsaufgabe ohne Nebenbedingungen mit zwei Design Variablen.
2. Wir berechnen die Gradienten und den einen kritischen Punkt: $x = -0.5, y = -0.25$. Wir berechnen die Hesse-Matrix. Diese ist unabhängig für alle (x, y) gemischt konvex-konkav, da sie eine negative Determinante besitzt. Deshalb ist der kritische Punkt ein Sattelpunkt und die Optimierungsaufgabe besitzt keine Lösung.
3. Matlab:

```
[ X,Y ] = meshgrid(-3:.1:3, -3:.1:3);  
$Z=4*X.*Y-X.^2+2*Y-1$;  
figure(4);surf(X,Y,Z)
```



3.1.1.3

- 1.
2. Die Zielfunktion wird aus der Summe der Abstandsquadrate gebildet:
 $f(x, y) = 5x^2 - 2x + 5y^2 - 7y + 40.25$. Somit entsteht eine unrestringierte, multi-variate, reelwertige Optimierungsaufgabe:

$$f^* = \min_{(x,y) \in \mathbb{R}^2} (5x^2 - 2x + 5y^2 - 7y + 40.25).$$

3. Wir berechnen den Gradienten und bestimmen die Werte von (x, y) für die der Gradient verschwindet: $x^* = 0.2; y^* = 0.7, f^* \approx 37.6$. An der Hesse-Matrix erkennen wir, dass die Krümmung konvex ist. Somit handelt es sich um ein Minimum.

3.1.1.4

1. Skizze wie Aufgabe zuvor.
2. Die Zielfunktion wird aus der Summe der Abstandsquadrate gebildet. Somit entsteht eine unrestringierte, multi-variate, reelwertige Optimierungsaufgabe:

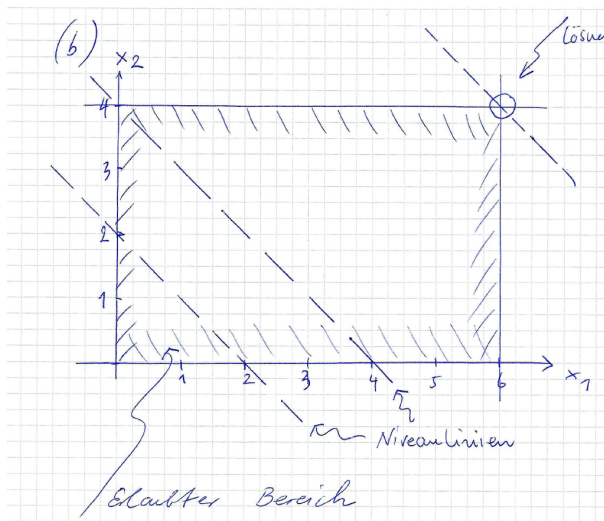
be:

$$\begin{aligned}
 f(x, y) &= \frac{1}{a^2} \sum t_i^2 \cdot s_i^2 \\
 f(x, y) &= 16x^2 + 26x + 16y^2 - 5y + 95.25 \\
 f^* &= \min_{(x, y) \in \mathbb{R}^2} (16x^2 + 26x + 16y^2 - 5y + 95.25).
 \end{aligned}$$

3. Wir berechnen den Gradienten und bestimmen die Werte von (x, y) für die der Gradient verschwindet: $x^* = -\frac{13}{16}; y^* = \frac{5}{32}, f^* \approx 84.297$. An der Hesse-Matrix erkennen wir, dass die Krümmung konvex ist. Somit handelt es sich um ein Minimum.

3.2.1.1

1. Linear Programming, LP.

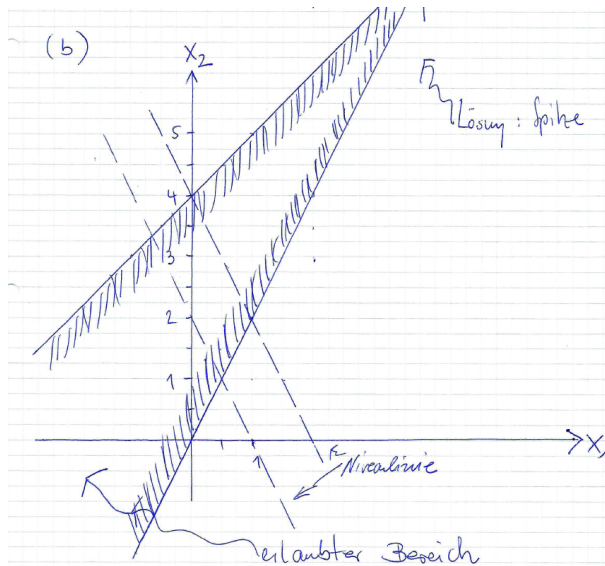


2. Die Lösung ist: $x_1^* = 6; x_2^* = 4$ und $f^* = -10$.

3.2.1.2

1. Linear Programming, LP.

2. Die Lösung ist: $x_1^* = 4; x_2^* = 8$ und $f^* = -8$.



3.2.1.3

1. Das Optimierungsproblem ist für $x \in \mathbb{R}^4$:

$$f^* = \max(1.5x_1 + x_2 + 2x_3 + 1.4x_4), \quad (3.1)$$

$$x_1 + x_2 \leq 1300 \quad (3.2)$$

$$x_1 + x_3 + x_4 \leq 2000 \quad (3.3)$$

$$x_4 \leq 500 \quad (3.4)$$

$$x_1 \geq 0 \quad (3.5)$$

$$x_2 \geq 0 \quad (3.6)$$

$$x_3 \geq 800 \quad (3.7)$$

$$x_4 \geq 0 \quad (3.8)$$

Es handelt sich um ein Linear Programming (LP) Problem.

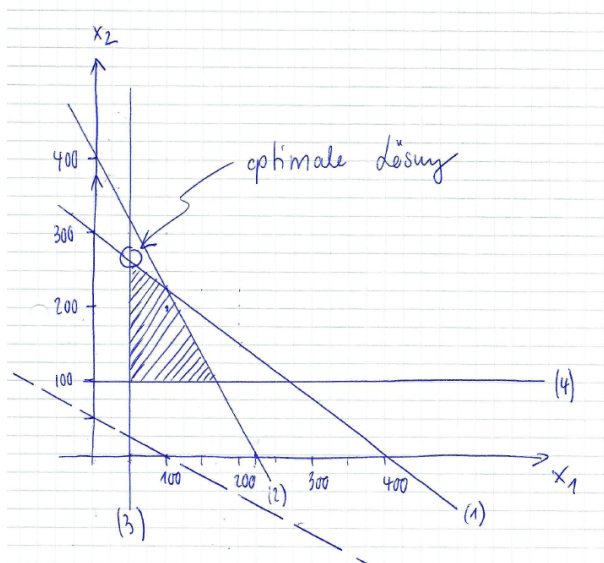
2. ...

3.2.1.4

1. Die Optimierungs-Aufgabe ist für $(x_1, x_2) \in \mathbb{R}^2$:

$$\begin{aligned} G^* &= \max 1000(x_1 + 2x_2), \\ 0.15x_1 + 0.2x_2 &\leq 60 \\ 0.2x_1 + 0.1x_2 &\leq 40 \\ x_1 &\geq 50 \\ x_2 &\geq 100 \end{aligned}$$

Es handelt sich um ein LP.



2. $x_1^* = 50$; $x_2^* = 262.5$; $f^* = 575'000$. Die aktiven Nebenbedingungen sind (1) und (3).

3.2.1.5

1. Die Optimierungsaufgabe ist:

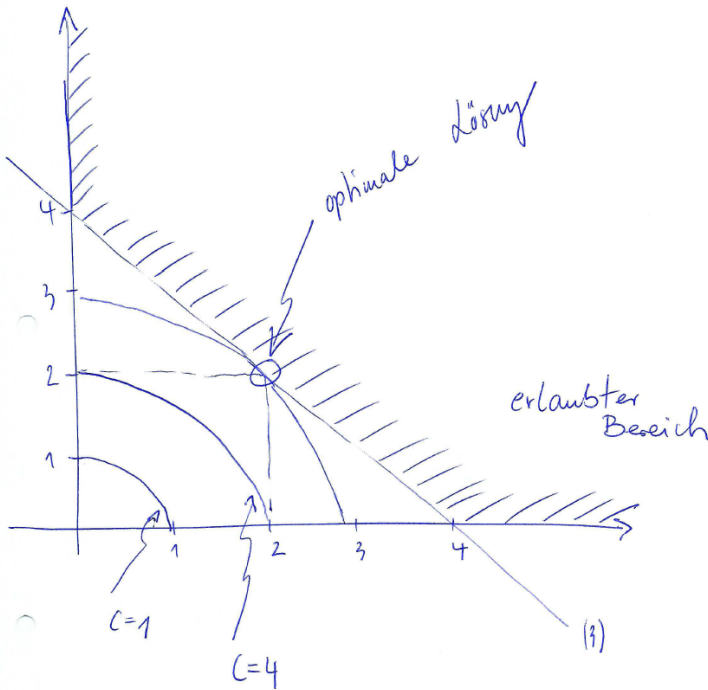
$$\begin{aligned}
 \min(x_1 + x_2 + x_3 + x_4), \\
 -2x_1 + 8x_2 + 10x_4 &\geq 0 \\
 5x_1 + 2x_2 &\geq 40 \\
 3x_1 - 5x_2 + 10x_3 - 2x_4 &\geq 15 \\
 -2x_1 + 8x_2 + 10x_4 &\leq 50 \\
 5x_1 + 2x_2 &\leq 140 \\
 3x_1 - 5x_2 + 10x_3 - 2x_4 &\leq 40 \\
 x_1 &\geq 0, \\
 x_2 &\geq 0, \\
 x_3 &\geq 0, \\
 x_4 &\geq 0.
 \end{aligned}$$

2. Die Optimierungsaufgabe ist:

$$\begin{aligned}
 \min(x_1 + x_2 + x_3 + x_4), \\
 6x_1 + 5x_2 + 10x_3 + 8x_4 &\geq 55 \\
 -2x_1 + 8x_2 + 10x_4 &\leq 50 \\
 5x_1 + 2x_2 &\leq 140 \\
 3x_1 - 5x_2 + 10x_3 - 2x_4 &\leq 40 \\
 x_1 &\geq 0, \\
 x_2 &\geq 0, \\
 x_3 &\geq 0, \\
 x_4 &\geq 0.
 \end{aligned}$$

3.3.1.1

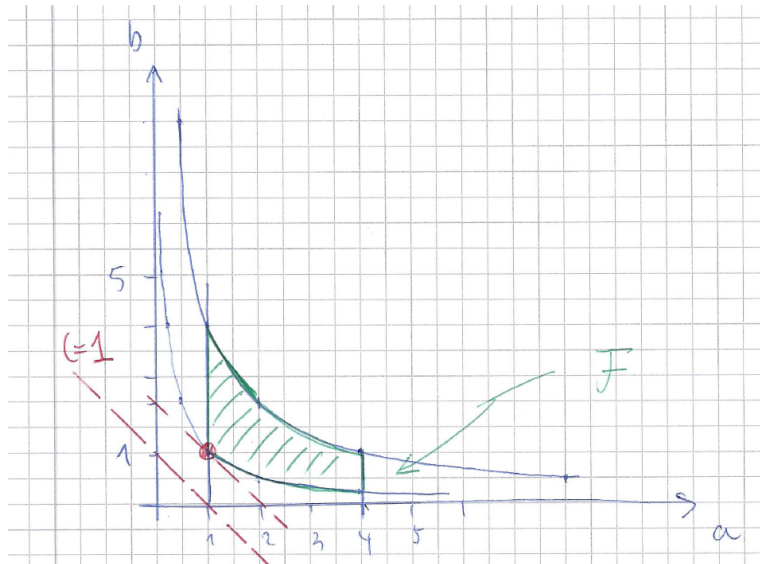
1. Nonlinear Programming, NLP. Speziell: Quadratic Programming, QP.



2. Die Lösung ist: $x_1^* = 2; x_2^* = 2$. Die aktive Nebenbedingung ist (3).

3.3.1.2

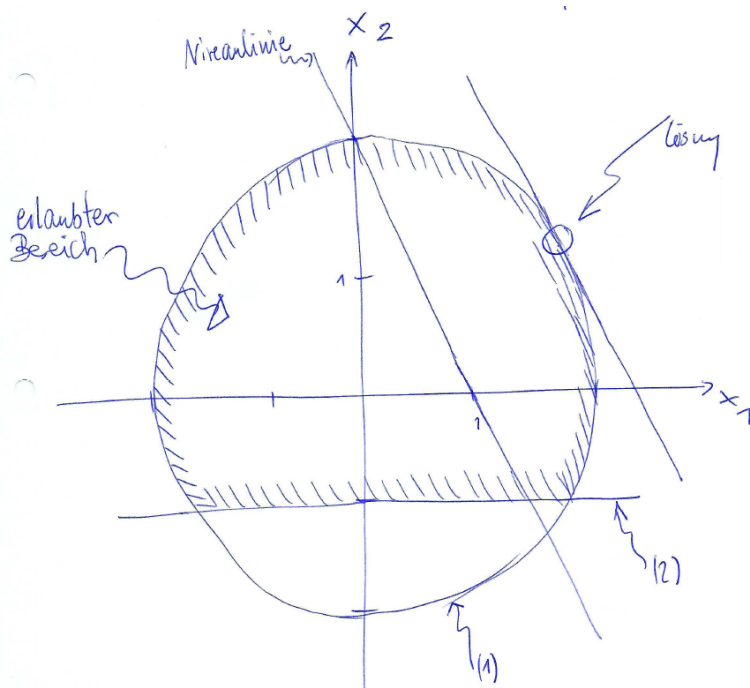
1. Es handelt sich um ein NLP, da: (a) 2 reelwertige Design-Variablen, (b) 1 lineare Zielfunktion und (c) 2 lineare und 2 nichtlineare Ungleichheits-Nebenbedingungen.



2. Die Lösung ist: $a^* = 1; b^* = 1$. Die aktiven Nebenbedingung sind (1) und (4).

3.3.1.3

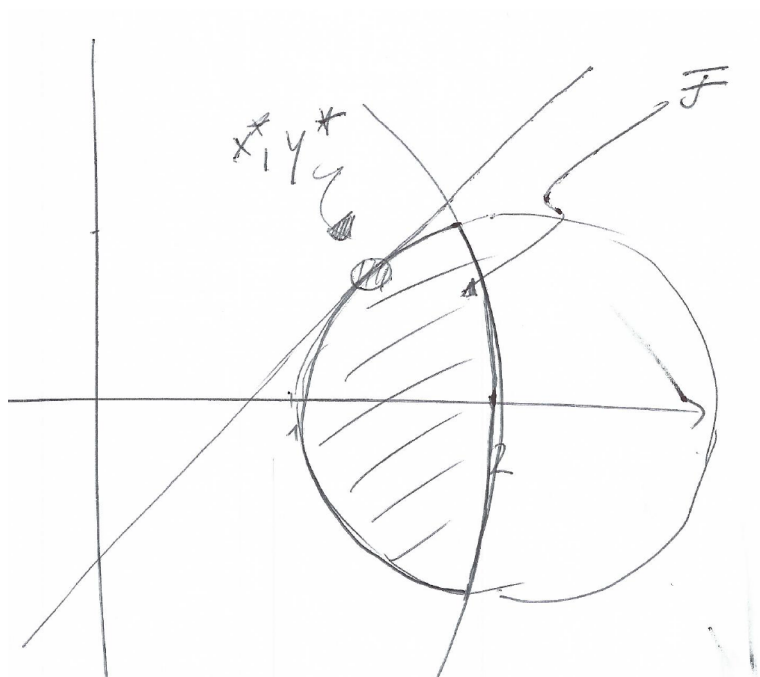
1. Nonlinear Programming.



2. Die Lösung ist: $x_1^* = \frac{4}{5}\sqrt{5}$; $x_2^* = \frac{2}{5}\sqrt{5}$. Die aktive Nebenbedingung ist (1).

3.3.1.4

1. Nonlinear Programming.



2. Die Lösung ist: $x^* = 2 - \frac{\sqrt{2}}{2}$; $y^* = \frac{\sqrt{2}}{2}$; $f^* = \sqrt{2} - 2$. Die aktive Nebenbedingung ist (2).

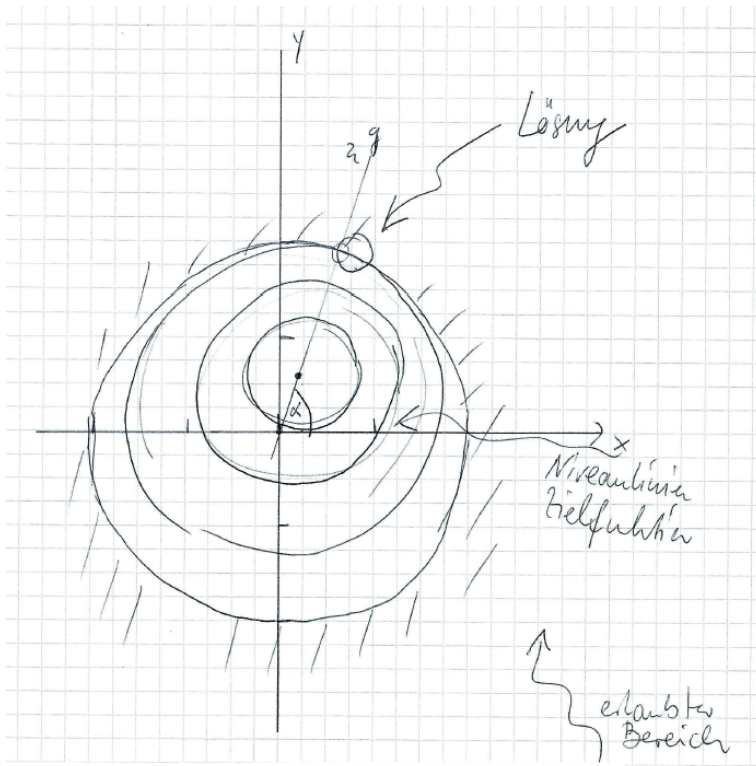
3.3.1.5

1. Skizze s. Aufg. zuvor.
2. Das Optimierungsproblem ist:

$$\min_{(x,y) \in \mathbb{R}^2} (5x^2 - 2x + 5y^2 - 7y + 40.25),$$

$$x^2 + y^2 \geq 4.$$

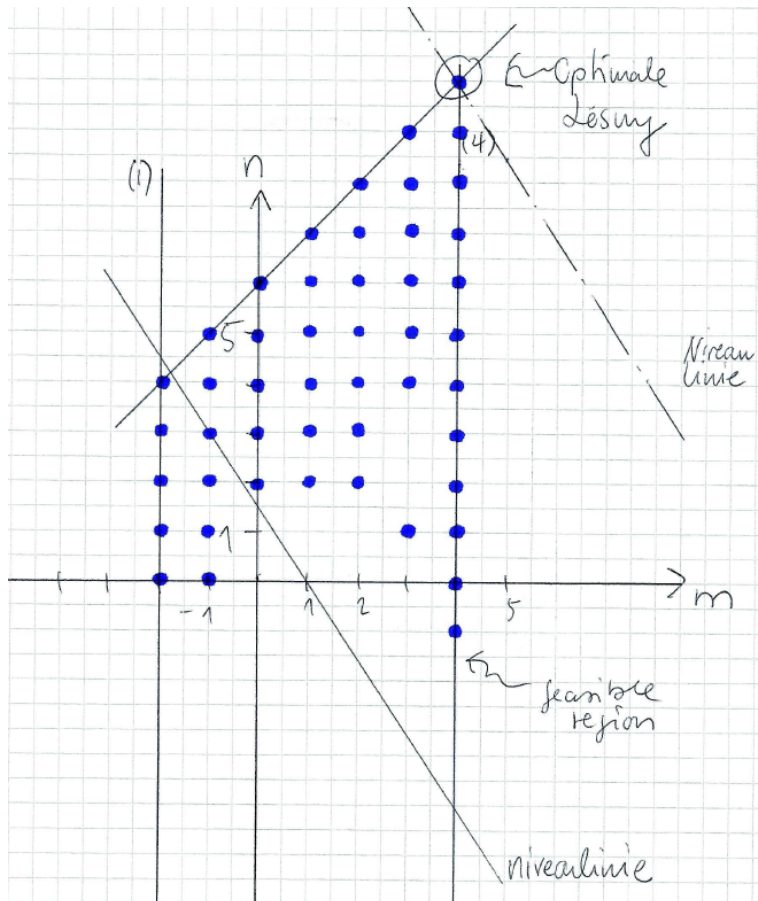
Es handelt sich um ein NLP.



3. Zur Lösung muss der Schnittpunkt der beiden sich berührenden Kreise bestimmt werden. Dazu legen wir eine Gerade vom Ursprung zum Berührungspunkt der beiden Kreise und erhalten für deren Winkel mit der x-Achse: $\tan(\alpha) = \frac{0.7}{0.2} = 3.5$. Damit ist $\alpha = 12925$ in Bogenmass und wir erhalten die Lösung: $x^* \approx 0.5495, y^* \approx 1.923, f^* \approx 45.690$.

3.4.1.1

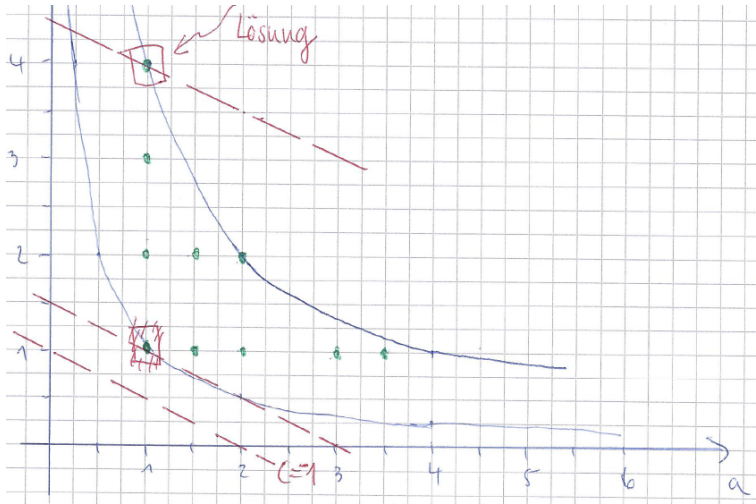
1. Es handelt sich um ein Integer LP.
2. Der erlaubte Bereich besitzt unendlich Elemente. Lösung: $m^* = 4; n^* = 10$.



Nebenbedingung (2) und (3) sind aktiv.

3.4.1.2

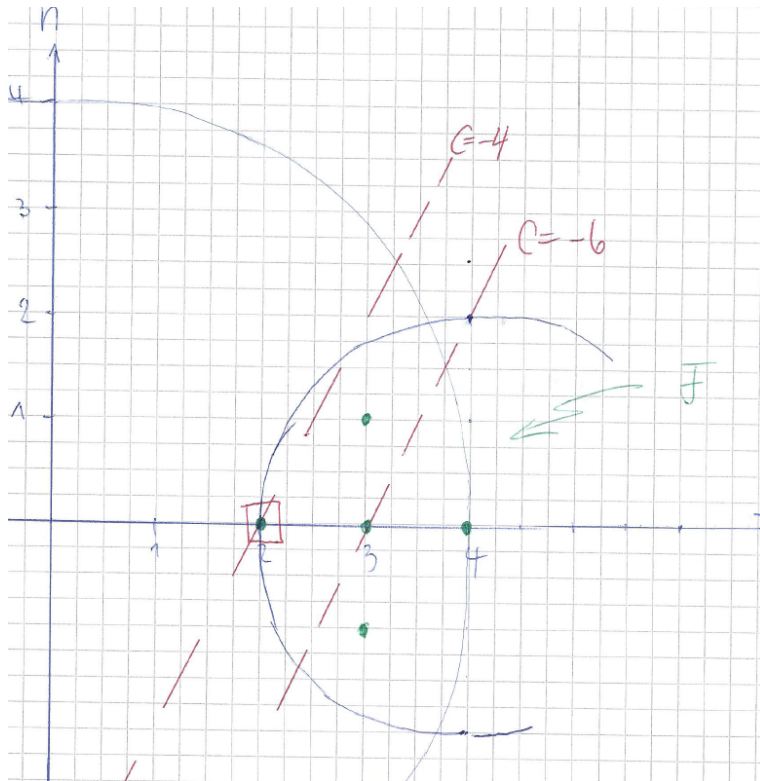
1. Es handelt sich um ein Integer LP.
2. Der erlaubte Bereich besitzt 10 Elemente. Lösung: $a^* = 1; b^* = 4; f^* = -7$.



Nebenbedingung (2) ist aktiv.

3.4.1.3

1. Es handelt sich um ein Integer-NLP.
2. Der erlaubte Bereich besitzt 5 Elemente. Lösung: $m^* = 2; n^* = 0; f^* = -4$.



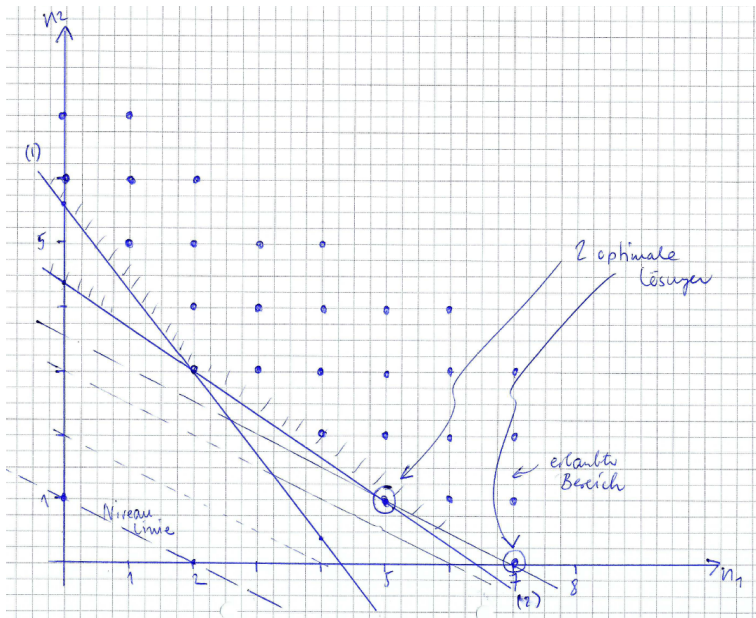
Nebenbedingung (2) ist aktiv.

3.4.1.4

Wir modellieren die Aufgabe als LP mit n_1, n_2 jeweils als Anzahl der gekauften Packungen 1 bzw. 2. Somit entsteht ein Integer-LP mit $(n_1, n_2) \in \mathbb{N}_0^2$:

$$\begin{aligned} f^* &= \min(n_1 + 2n_2), \\ 4n_1 + 3n_2 &\geq 17, \\ 2n_1 + 3n_2 &\geq 13, \end{aligned}$$

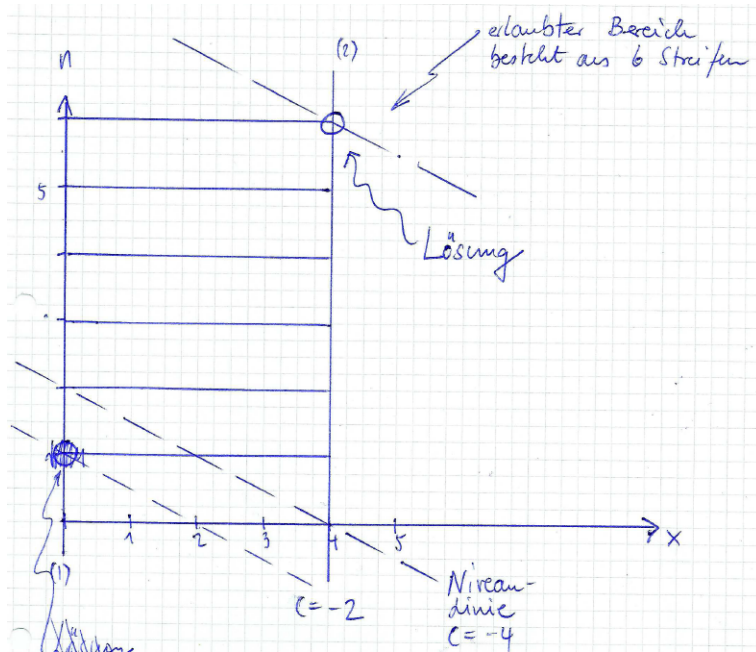
Es gibt zwei gleich gute Lösungen: Entweder fünf Packungen 1 und eine Pa-



ckung 2 oder sieben Packungen 1 und keine Packung 2.

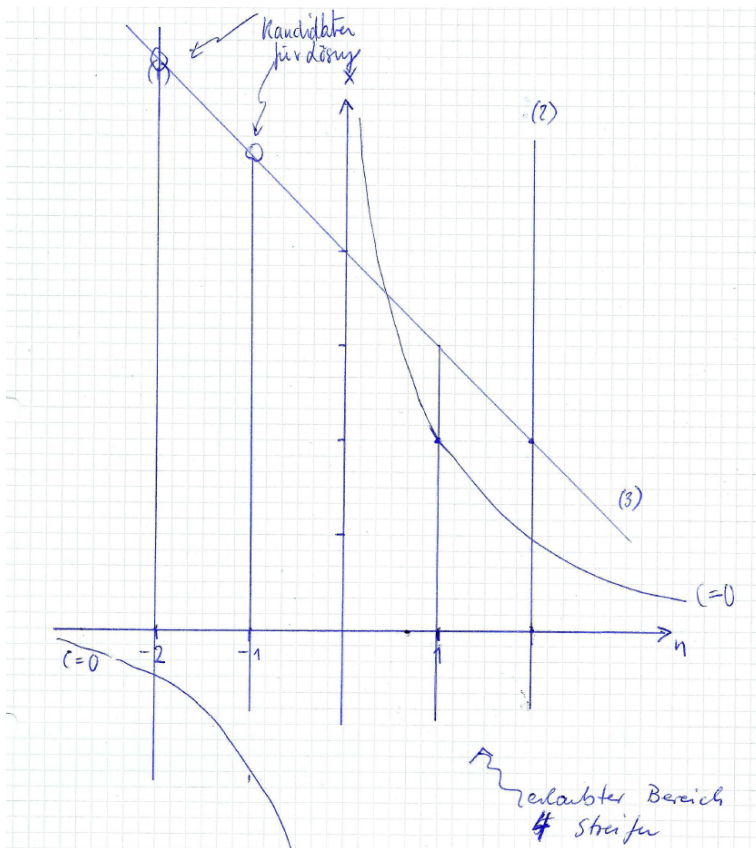
3.5.1.1

1. Es handelt sich um eine mixed-integer Optimierungsaufgabe.
2. Lösung: $n^* = 6$; $x^* = 4$.



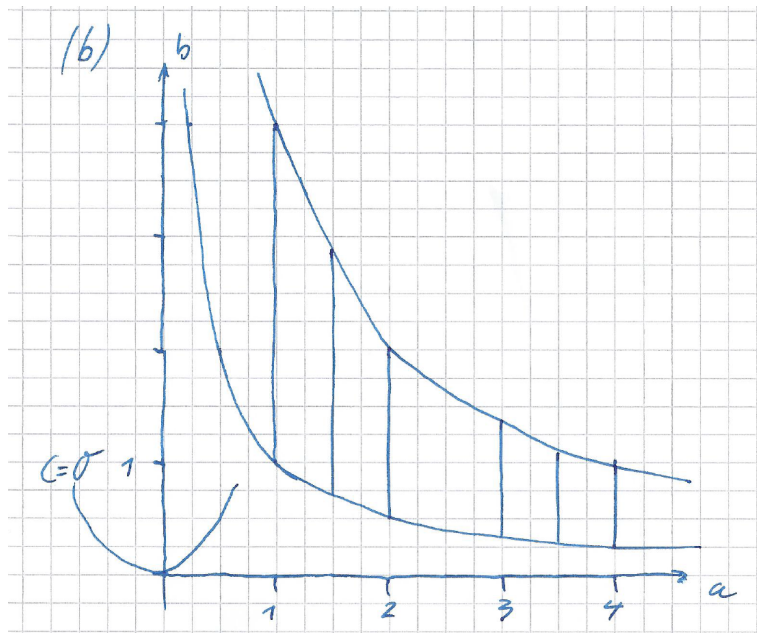
3.5.1.2

1. Es handelt sich um eine mixed-integer Optimierungsaufgabe.
2. Die Optimierungsaufgabe besitzt 2 Lösungen: I) $n_1^* = -1; x_1^* = 5$ und II) $n_2^* = -2; x_2^* = 6$ mit jeweils $f^* = -7$.



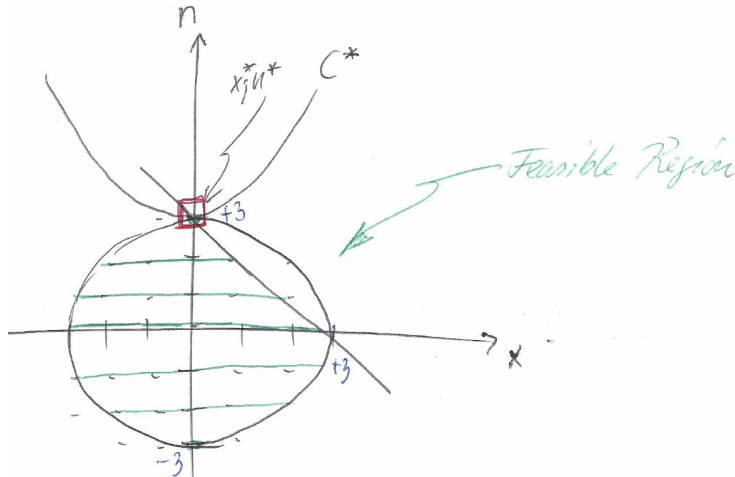
3.5.1.3

1. Es handelt sich um ein MINLP, da: (a) 1 reelwertige und 1 Integer-Design-Variable, (b) 1 lineare Zielfunktion und (c) 2 nichtlineare Ungleichheits-Nebenbedingungen.
2. $a^* = 1; b^* = 4; f^* = 3$. Die aktive Nebenbedingung ist (2).



3.5.1.4

1. Mixed-Integer Nonlinear Programming.
2. Die Feasible Region besteht aus unendlich vielen Elementen. Lösung: $x^* = 0; n^* = 3; f^* = -3$. Die aktive Nebenbedingungen sind (1) und (2).



3.6.1.1

1. Es handelt sich um eine kombinatorische Optimierungsaufgabe.
2. 6 Varianten (davon aber jeweils 3 mit umgekehrter Reiserichtung und identischer Wegstrecke), Es gibt 2 optimale Lösungen: I) ACBDA und II) ADBCA.

3.6.1.2

1. 42 Tage.
2. 41'466 Jahre.
3. 5.6-mal die Lebensdauer des Universums.

3.7.1

wahr sind 2.

3.7.2

wahr sind 1, 4.

3.7.3

wahr sind 3, 5.

3.7.4

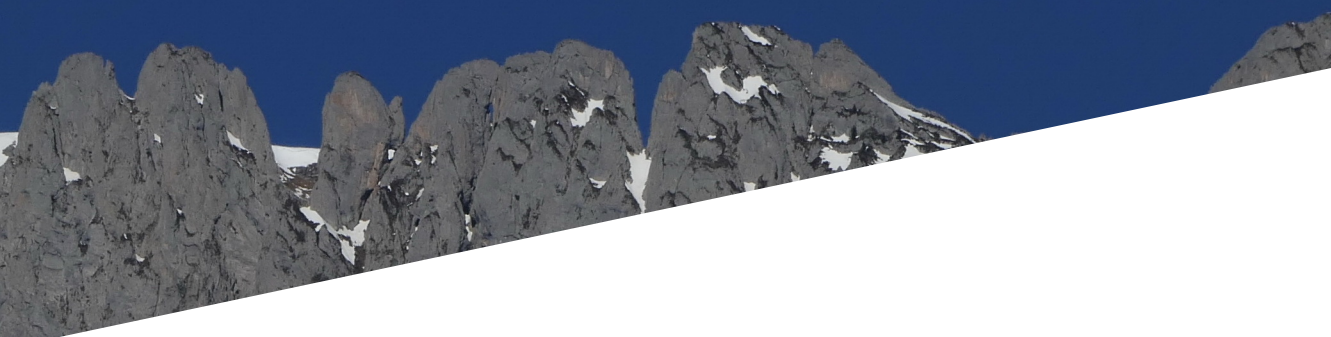
wahr sind 2, 4.

3.7.5

wahr sind 1, 2, 3, 6.

3.7.6

wahr sind 2, 3.



Kapitel 4

Numerische Mathematik, Algorithmen und das Newton-Verfahren

4.7.1

1. Es handelt sich um ein univariates, reellwertiges Optimierungsproblem mit 2 Ungleichheits-Nebenbedingungen.
2. Es sind einige Fallunterscheidungen zu treffen. Lösung in Pseudo-Code:
 - 1: $x^* = -\frac{b}{2a}$.
 - 2: Falls $0 \leq x^* \leq 21$ und $a > 0$: $f^* = a(x^*)^2 + bx^* + c$.
 - 3: Falls $x^* > 21$ und $a > 0$: $x^* = 21, f^* = a(x^*)^2 + bx^* + c$.
 - 4: Falls $x^* < 0$ und $a > 0$: $x^* = 0, f^* = a(x^*)^2 + bx^* + c$.
 - 5: Falls $0 \leq x^* \leq 11.5$ und $a < 0$: $x^* = 21, f^* = a(x^*)^2 + bx^* + c$.
 - 6: Falls $11.5 < x^* \leq 21$ und $a < 0$: $x^* = 0, f^* = a(x^*)^2 + bx^* + c$.
 - 7: Falls $x^* > 21$ und $a < 0$: $x^* = 0, f^* = a(x^*)^2 + bx^* + c$.
 - 8: Falls $x^* < 0$ und $a < 0$: $x^* = 21, f^* = a(x^*)^2 + bx^* + c$.

4.7.2

1. Es handelt sich um ein univariates, reellwertiges Optimierungsproblem mit 2 Ungleichheits-Nebenbedingungen.
2. $x^* \approx 7.405$. Die Zielfunktion besitzt 13 lokale Minima innerhalb der Feasible Region.
3. Eine Möglichkeit könnte sein für ein Programm in Pseudo-Code:
 - 1: Sei $x^* = -10, f^* = f(-10)$ und $j = 1$.

- 2: Wähle eine Zufallszahl x im Bereich $-10 \leq x \leq 22$ mit uniformer Wahrscheinlichkeitsdichte.
- 3: Falls $f(x) < f(x^*)$, dann $x^* = x$ und $j = j + 1$.
- 4: Gehe zu 2, falls $j < J_{max}$.
- 5: Stopp, falls $j = J_{max}$.

4. Matlab

```
f_goal=@(x) sin(4.17*x-7.32)+0.079*x.^2-1.14*x
Jmax=100;
xs=-10;fs=f_goal(xs);
for jj=1:Jmax,
    x=32*rand-10;
    f=f_goal(x);
    if f<fs,
        fs=f; xs=x;
    end
end
display(['Globales Minimum: ' num2str(xs,12)]);
```

5. Bei ca. $J_{max} > 500$ stochastischen Versuchen bleibt der Fehler kleiner als 1 Prozent (in > 9 von 10 Fällen).
6. Es gibt hier zahlreiche Ideen. Eine Möglichkeit ist: Anstatt die Zufallszahl x immer in der ganzen Feasible Region mit gleicher Wahrscheinlichkeit zu bestimmen, könnte die Zufallszahl in der Nähe des bisher gefundenen Minimums gewählt werden.

4.7.3

1. Es handelt sich um ein reellwertiges, univariates, unrestringiertes Optimierungsproblem.
- 2.

$$\begin{aligned} f(x) &= x^2 + \sin(x), \\ f'(x) &= 2x + \cos(x), \\ f''(x) &= 2 - \sin(x), \end{aligned}$$

Damit ist die Newton'sche Rekursionsformel:

$$x_{i+1} = x_i - \frac{2x_i + \cos(x_i)}{2 - \sin(x_i)}$$

3.

$$\begin{aligned}
 x_1 &= x_0 - \frac{2x_0 + \cos(x_0)}{2 - \sin(x_0)} \\
 &= -\frac{\cos(0)}{2 - \sin(0)} \\
 &= -\frac{1}{2}
 \end{aligned}$$

4. Falls die Newton'sche Rekursionsformel auf einen Wert x^* konvergiert, kann nicht garantiert werden, dass es sich bei x^* um die Lösung der Optimierungsaufgabe handelt. Dafür sind zwei Gründe ausschlaggebend: (a) Die Rekursions-Formel kann sowohl auf lokale Minima als auch auf lokale Maxima konvergieren und (b) von einem gefundenen lokalen Minimum kann nicht garantiert werden, dass es die Lösung der Optimierungsaufgabe ist, da es weitere, bessere lokale Minima geben kann.

4.7.4

1. Die lokalen Minima sind: $x_1 \approx -1.7$; $x_2 \approx 8.5$;

2.

$$\begin{aligned}
 f(x) &= x^4 - 10x^3 - 20x^2 + 40x - 1, \\
 f'(x) &= 4x^3 - 30x^2 - 40x + 40, \\
 f''(x) &= 12x^2 - 60x - 40,
 \end{aligned}$$

Damit ist das Modifizierte Newton-Verfahren:

$$x_{i+1} = x_i - \frac{4x_i^3 - 30x_i^2 - 40x_i + 40}{|12x_i^2 - 60x_i - 40|}$$

3. Wir wählen nun geeignete Startwerte in der Nähe der lokalen Minima und erhalten: $x_1 \approx -1.71689688931$; $x_2 \approx 8.5344305471$; $f(x_1) \approx -69.332$; $f(x_2) \approx -2027.377$

Matlab:

```

% Zielfunktion grafisch darstellen
x=(-10:0.01:10);f=x.^4-10*x.^3-20*x.^2+40*x-1;figure(1);plot(x,f)
% lokale Minima mit modifiziertem Newton-Verfahren
xhist=[];fshist=[];xi=-1.7;epsGrad=1e-12;grad=2*epsGrad;N=0;
for ii=1:20,
    xhist =[xhist xi];

```

4 Numerische Mathematik, Algorithmen und das Newton-Verfahren

```
fs=4*xi^3-30*xi^2-40*xi+40;
fss=12*xi^2-60*xi-40;
fshist=[fshist fs];
xipe=xi-fs/abs(fss);
xi=xipe;
end
xstern=xipe;
fstern=xstern.^4-10*xstern.^3-20*xstern.^2+40*xstern-1
display(['lokales Minimum: ' num2str(xstern,12)]);
display(['Wert der Ableitung: ' num2str(4*xi^3-30*xi^2-40*xi+40)]);
display(['Anzahl Iterationen: ' num2str(N)]);
display(['Funktionswert ' num2str(fstern,12)]);
```

4. Die Zielfunktion ist ein Polynom 4. Ordnung. Deshalb kann sie höchstens 2 lokale Minima besitzen. Das lokale Minimum mit dem niedrigeren Funktionswert ist deshalb die Lösung der Optimierungsaufgabe: $x^* \approx -1.71689688931$; $f^* \approx -2027.377$.

4.7.5

1. Lokale Minima sind: $x_1 \approx 1.2$; $x_2 \approx 2.4$; $x_3 \approx 3.7$; $x_4 \approx 4.9$.
2. Lokale Minima sind: $x_1 \approx 1.23485804556$; $f(x_1) \approx -0.889993557896$; $x_2 \approx 2.40329544014$; $f(x_2) \approx -0.886234434319$; $x_3 \approx 3.61384874227$; $f(x_3) \approx -0.257608703498$; $x_4 \approx 4.88998633256$; $f(x_4) \approx -0.0322192610066$;

Matlab:

```
% Zielfunktion im Bereich 1:5 darstellen
x=(1:0.01:5);f=sin(2*x-0.5).*cos(pi*x -1);
figure(1);plot(x,f);
% Modifiziertes Newton-Verfahren
xhist =[];fshist=[];xi=4.9;epsGrad=1e-12;normGrad=2*epsGrad;N=0;
while normGrad > epsGrad,
    xhist =[xhist xi];
    fs=2*cos(2*xi-0.5)*cos(pi*xi -1)-pi*sin(2*xi-0.5)*sin(pi*xi -1);
    fss=-4*sin(2*xi-0.5)*cos(pi*xi -1) - 2*pi*cos(2*xi-0.5)*sin(pi*xi -1)-...
        (2*pi*cos(2*xi-0.5)*sin(pi*xi -1) + pi*pi*sin(2*xi-0.5)*cos(pi*xi -1));
    fshist=[fshist fs];
    normGrad=abs(fs);
    xipe=xi-fs/abs(fss);
    schrittweite=abs(xipe-xi);
    N=N+1;
    xi=xipe;
end
```

```
xstern=xipe;  
fstern=sin(2*xstern-0.5).*cos(pi*xstern -1);  
display(['lokales Minimum: ' num2str(xstern,12)]);  
display(['Wert der Ableitung: ' num2str(abs(fs))]);  
display(['Anzahl Iterationen: ' num2str(N)]);  
display(['Funktionswert ' num2str(fstern,12)]);
```

3. Die Lösung der Optimierungs-Aufgabe ist unbekannt.



Kapitel 5

Das Simplex-Verfahren zur Lösung von LP-Problemen

5.4.1

$x_1^* = 6; x_2^* = 4$. Die aktiven Nebenbedingungen sind (1) und (2). Überprüfen, ob $\text{exitflag}=1$.

Matlab:

```
A=[];b=[];f=[-1;-1];Aeq=[];beq=[];lb=[0;0];ub=[6;4];x0=[];
options=optimoptions('linprog','Algorithm','simplex','Display','iter');
[xs,fs,exitflag,output,lambda] = linprog(f,A,b,Aeq,beq,lb,ub,x0,options);
lambda.ineqlin, xs, fs, exitflag
```

5.4.2

$x_1^* = 4; x_2^* = 8$. Die aktiven Nebenbedingungen sind (1) und (2). Überprüfen, ob $\text{exitflag}=1$.

Matlab:

```
A=[-1 1;2 -1];b=[-4;0];f=[-1;-1/2];Aeq=[];beq=[];lb=[];ub=[];x0=[];
options=optimoptions('linprog','Algorithm','simplex','Display','iter');
[xs,fs,exitflag,output,lambda] = linprog(f,A,b,Aeq,beq,lb,ub,x0,options);
lambda.ineqlin, xs, fs, exitflag
```

5.4.3

$x_1^* = 0; x_2^* \approx 4.592; x_3^* \approx 0.413$. Die aktiven Nebenbedingungen sind (2,3,4). Überprüfen, ob $\text{exitflag}=1$.

Matlab:

5 Das Simplex-Verfahren zur Lösung von LP-Problemen

```
A=[1 0.1 0.3; 0 1 -0.9; 2.4 1 2.2; -1 0 0; 0 -1 0];
b=[6.1;4.22;5.5;0;0];Aeq=[];beq=[];ub=[];lb=[];
f=[-1;-1;-1.2];Aeq=[];beq=[];lb=[];ub=[];x0=[];
options=optimoptions('linprog','Algorithm','simplex','Display','iter');
[x,fval,exitflag,output,lambda] = linprog(f,A,b,Aeq,beq,lb,ub,x0,options)
lambda.ineqlin
```

5.4.4

$x_1^* = 0; x_2^* = 0; x_3^* \approx 3.1818$. Die aktiven Nebenbedingungen sind (1, 4, 5). Überprüfen, ob $\text{exitflag}=1$.

Matlab:

```
A=[2 0.44 3.3; 1 -2.22 0.9; -2 -1 -2.2];
b=[10.5;12.8;22.45];Aeq=[];beq=[];ub=inf(3,1);lb=[0;0;2.45];f=[-1;0.5;-2.8];
[x,fval,exitflag,output,lambda] = linprog(f,A,b,Aeq,beq,lb,ub)
lambda.ineqlin
lambda.lower
```

5.4.5

Es folgendes Optimierungs-Problem:

$$\min(-1.5x_1 - x_2 - 2x_3 - 1.4x_4),$$
$$x_1 + x_2 \leq 1300 \quad (5.1)$$

$$x_1 + x_2 + x_4 \leq 2000 \quad (5.2)$$

$$x_4 \leq 500 \quad (5.3)$$

$$x_1 \geq 0, \quad (5.4)$$

$$x_2 \geq 0, \quad (5.5)$$

$$x_3 \geq 800 \quad (5.6)$$

$$x_4 \geq 0. \quad (5.7)$$

1. Linear Programming, LP.

2. $x_1^* = 0; x_2^* = 1.30; x_3^* \approx 2.0; x_4^* = 0$. Der maximale Gewinn ist 5'300 CHF. Die beiden lower Constraints der Design-Variablen (x_1, x_4) sind aktiv. Zusätzlich sind beide lineare Constraints aktiv.

Matlab:

```
A=[1 1 0 0;1 0 1 1];b=[1300;2000];
Aeq=[];beq=[];ub=inf(4,1);ub(4)=500;lb=[0;0;800;0];f=[-1.5;-1;-2;-1.4];
[x,fval,exitflag,output,lambda] = linprog(f,A,b,Aeq,beq,lb,ub)
lambda.ineqlin,lambda.lower,lambda.upper
```

3. Variante i): $x_1^* = 0$; $x_2^* = 3.0$; $x_3^* \approx 2.0$; $x_4^* = 0$. Der maximale Gewinn ist 7'000 CHF.

Matlab:

```
A=[1 1 0 0;1 0 1 1];
b=[3000;2000];Aeq=[];beq=[];ub=inf(4,1);ub(4)=500;lb=[0;0;800;0];f=[-1.5;-1;-2;-1.4];
[x,fval_i,exitflag,output,lambda] = linprog(f,A,b,Aeq,beq,lb,ub);
```

Variante ii): $x_1^* = 0$; $x_2^* = 1.30$; $x_3^* \approx 3.4$; $x_4^* = 0$. Der maximale Gewinn ist 8'100 CHF.

Matlab:

```
A=[1 1 0 0;1 0 1 1];
b=[1300;3400];Aeq=[];beq=[];ub=inf(4,1);ub(4)=500;lb=[0;0;800;0];f=[-1.5;-1;-2;-1.4];
[x,fval_ii,exitflag,output,lambda] = linprog(f,A,b,Aeq,beq,lb,ub);
```

5.4.6

1. Die Optimierungsaufgabe ist:

$$\min(x_1 + x_2 + x_3 + x_4),$$

$$-2x_1 + 8x_2 + 10x_4 \geq 0 \quad (5.1)$$

$$5x_1 + 2x_2 \geq 40 \quad (5.2)$$

$$3x_1 - 5x_2 + 10x_3 - 2x_4 \geq 15 \quad (5.3)$$

$$-2x_1 + 8x_2 + 10x_4 \leq 50 \quad (5.4)$$

$$5x_1 + 2x_2 \leq 140 \quad (5.5)$$

$$3x_1 - 5x_2 + 10x_3 - 2x_4 \leq 40 \quad (5.6)$$

$$x_1 \geq 0, \quad (5.7)$$

$$x_2 \geq 0, \quad (5.8)$$

$$x_3 \geq 0, \quad (5.9)$$

$$x_4 \geq 0. \quad (5.10)$$

Ein Beispiel für ein Matlab-Skript ist:

```
S_S=50;S_VS=60; S_L=10;
total_S=100;total_VS=200;total_L=50;
A=[ 2 -8 0 -10;...
    -5 -2 0 0;...
```

5 Das Simplex-Verfahren zur Lösung von LP-Problemen

```

-3  5 -10  2;...
-2  8   0 10;...
 5  2   0  0;...
 3 -5  10 -2];
b=[(S_S-total_S/2) (S_VS-total_VS/2) (S_L-total_L/2) ...
   (total_S-S_S) (total_VS-S_VS) (total_L-S_L)];
c=[1 1 1 1];
lb=zeros(4,1);ub=[Inf; Inf; Inf; Inf];
[xsa,fvala,exitflaga,outputa,lambdaa] = linprog(c,A,b,[],[],lb,ub);
xsa, lambdaa.ineqlin

```

Die Lösung ist: $x^* = (7.4775, 1.3063, 0.0000, 0.4505)$, die einzusetzenden Gesamtwahlkampfgelder sind: 9'234 SFr. Die Lagrange-Multiplikatoren zeigen, dass in jeder Gruppe genau die Hälfte der Stimmen erreicht wird. Die Lösung erscheint sinnvoll.

2. Die Optimierungsaufgabe ist:

$$\begin{aligned}
 \min(x_1 + x_2 + x_3 + x_4), \\
 6x_1 + 5x_2 + 10x_3 + 8x_4 &\geq 55 \\
 -2x_1 + 8x_2 + 10x_4 &\leq 50 \\
 5x_1 + 2x_2 &\leq 140 \\
 3x_1 - 5x_2 + 10x_3 - 2x_4 &\leq 40 \\
 x_1 &\geq 0, \\
 x_2 &\geq 0, \\
 x_3 &\geq 0, \\
 x_4 &\geq 0.
 \end{aligned}$$

Ein Beispiel für ein Matlab-Skript ist:

```

A3=[];A2=[];
for ii=1:4, A3(ii)=sum(A(1:3,ii));end
A2=[A3;A(4:6,:)];
b2=[sum(b(1:3)) b(4:6)];
lb=zeros(4,1);ub=[Inf; Inf; Inf; Inf];
[xsb,fvalb,exitflagb,outputb,lambdab] = linprog(c,A2,b2,[],[],lb,ub);
xsb, lambdab.ineqlin

```

Die Lösung ist: $x^* = (0, 0, 4.3, 1.5)$, die einzusetzenden Gesamtwahlkampfgelder sind: 5'800 SFr. Die Lagrange-Multiplikatoren zeigen, dass: (a) im gesamten die Hälfte der Stimmen erreicht wird und (b) der Stimmenanteil auf dem Land 100% beträgt. Diese Lösung erscheint nicht sinnvoll. Um eine sinnvolle

Lösung zu erreichen müsste Berater B z.B. noch eine maximale Anzahl von Stimmen vorgeben, die der Kandidat in jeder Gruppe überhaupt erreichen kann.



Kapitel 6

Das SQP-Verfahren zur Lösung von NLP-Problemen

6.3.1

$x_1^* = 2; x_2^* = 2$. Die aktive Nebenbedingung ist (3).

Skript:

```
x0=[1;1]';A=[-1 -1];b=[-4];Aeq=[];beq=[];lb=[0;0]';ub=[];
options = optimoptions('fmincon','Algorithm','sqp','Display','iter',...
    'SpecifyConstraintGradient',true,'SpecifyObjectiveGradient',true,...
    'TolX',1e-6,'TolFun',1e-6);
[xs,fs,exitflag] = fmincon(@goalfunc,x0,A,b,Aeq,beq,lb,ub,[],options);
xs,exitflag
```

Zielfunktion:

```
function [ f,grad ] = goalfunc(x)
f=x(1)*x(1)+x(2)*x(2);
grad(1)=2*x(1);
grad(2)=2*x(2);
end
```

6.3.2

$x_1^* \approx 1.788854; x_2^* \approx 0.8944278$. Die aktive Nebenbedingung ist (1).

Matlab:

Skript:

```
x0=[1;1]';A=[];b=[];Aeq=[];beq=[];lb=[-1;-1]';ub=[];
options = optimoptions('fmincon','Algorithm','sqp','Display','iter',...
    'SpecifyConstraintGradient',true,'SpecifyObjectiveGradient',true,...
```

```
'TolX',1e-6,'TolFun',1e-6);
[xs,fs,exitflag] = fmincon(@goalfunc,x0,A,b,Aeq,beq,lb,ub,@nonlincon,options);
xs,exitflag
```

Zielfunktion:

```
function [ f,grad ] = goalfunc(x)
f=-x(1)-x(2)/2;
grad(1)=-1;
grad(2)=-0.5;
end
```

Nebenbedingungen:

```
function [c,ceq,cgrad,ceqgrad] = nonlincon(x)
c=x(1)*x(1)+x(2)*x(2)-4;
ceq=[];
cgrad(1)=2*x(1);
cgrad(2)=2*x(2);
cgrad=cgrad';
ceqgrad=[];
end
```

6.3.3

Das Optimierungsproblem ist:

$$U^* = \max(199x + 299y),$$

$$s. 4x^2 + 2.9y^2 - 1000 \leq 0 \quad (6.1)$$

$$x \geq 0 \quad (6.2)$$

$$y \geq 0 \quad (6.3)$$

$$(6.4)$$

Die Lösung ist: $x^*12.0526; y^* \approx 14.9869$. Der maximale Umsatz ist 6'880 SFr.

Skript:

```
z0=[1;1]';A=[];b=[];Aeq=[];beq=[];lb=[0;0]';ub=[];
options = optimoptions('fmincon','Algorithm','sqp','Display','iter',...
    'SpecifyConstraintGradient',true,'SpecifyObjectiveGradient',true,...
    'TolX',1e-6,'TolFun',1e-6);
[xs,fs,exitflag] = fmincon(@goalfunc,z0,A,b,Aeq,beq,lb,ub,@nonlincon,options);
xs,exitflag
```

Zielfunktion:

```
function [ f,grad ] = goalfunc(z)
```

```

x=z(1);
y=z(2);
f=-199*x-299*y;
grad(1)=-199;
grad(2)=-299;
end

```

Nebenbedingungen:

```

function [c,ceq,cgrad,ceqgrad] = nonlincon(z)
x=z(1);y=z(2);
c=2.4*x*x+2.9*y*y-1000;ceq=[];
cgrad(1)=4.8*x;cgrad(2)=5.8*y;cgrad=cgrad';ceqgrad=[];
end

```

6.3.4

Die optimale Lösung ist: $x^* \approx 12.3049$; $y^* \approx 20.0000$; $z^* \approx 142.8768$. Der maximale Umsatz ist 25'646,- SFr. Matlab:

Skriptfile:

```

z0=[1;1;1]';A=[];b=[];Aeq=[];beq=[];lb=[8;20;0];ub=[];
options = optimoptions('fmincon','Algorithm','sqp',...
    'GradObj','on','Display','iter',...
    'GradConstr','on','TolX',1e-6,'TolFun',1e-6);
[xs,fs,exitflag] = fmincon(@goalfunc,z0,A,b,Aeq,beq,lb,ub,@nonlincon,options);
xs,exitflag

```

Zielfunktion:

```

function [ f,grad ] = goalfunc(s)
x=s(1);
y=s(2);
z=s(3);
f=-249*x-279*y-119*z;
grad(1)=-249;
grad(2)=-279;
grad(3)=-119;
end

```

Nebenbedingungen:

```

function [c,ceq,cgrad,ceqgrad] = nonlincon(s)
x=s(1);y=s(2);z=s(3);
c=2.124*x*x+2.773*y*y+24.98*z-5000;
ceq=[];
cgrad=zeros(3,1);

```

```
cgrad(1)=4.248*x;
cgrad(2)=5.546*y;
cgrad(3)=24.98;
ceqgrad=[];
end
```

6.3.5

Das globale Maximum ist: $u^* = 2000$; $v^* = 400$; $w^* = 0$; $x^* \approx 148'041$. Der maximale Umsatz ist 1,79 Mio SFr.

Daneben gibt es ein weiteres lokales Optimum: $u^* = 2000$; $v^* = 400$; $w^* \approx 59'040$; $x^* =$, mit einem Umsatz von 1,32 Mio SFr. Wir verwenden Zufalls-Anfangswerte für die Suche mit SQP-Algorithmus, um die beiden lokalen Minima zu finden. Matlab:

Skriptfile:

```
s0=10000*rand(4,1);A=[];b=[];Aeq=[];beq=[];
options = optimoptions('fmincon','Algorithm','sqp','Display','iter',...
    'SpecifyConstraintGradient',true,'SpecifyObjectiveGradient',true,...
    'TolX',1e-6,'TolFun',1e-6);
lb=[2000;400;0;0];ub=[];
[xs0,fs0,exitflag] = fmincon(@goalfunc_0,s0,A,b,Aeq,beq,lb,ub,@nonlincon_0,options);
[xs0' fs0]
```

Zielfunktion:

```
function [ f,grad ] = goalfunc_0(s)
u=s(1);v=s(2);w=s(3);x=s(4);
f=-12.9*u-9.9*v-21.9*w-11.9*x;
grad(1)=-12.9;grad(2)=-9.9;
grad(3)=-21.9;grad(4)=-11.9;
end
```

Nebenbedingungen:

```
function [c,ceq,cgrad,ceqgrad] = nonlincon(s)
u=s(1);v=s(2);w=s(3);x=s(4);
c(1)=12.1*u+3800*(sqrt(v+1)-1)+2.6*w+400*(sqrt(x+1)-1)-250000;
cgrad=zeros(4,1);
cgrad(1,1)=12.1;cgrad(2,1)=3800/sqrt(v+1)/2;
cgrad(3,1)=2.6;cgrad(4,1)=400/sqrt(x+1)/2;
ceq=[];ceqgrad=[];
end
```



Kapitel 7

Relaxation und Enumeration zur Lösung von Integer- und mixed-integer Problemen

7.3.1

1. Es handelt sich um ein Mixed-Integer LP.
2. In der Enumeration sind jetzt 3 LP zu lösen. Der beste Wert ergibt sich bei $n^* = 3, x_1^* = 6; x_2^* = 0$ mit $f^* = 12$.

Matlab:

```
A=[1 1;-1 -1;-1 1];  
b=1*[2;2;-1;-1];Aeq=[];beq=[];ub=[];lb=[];f=[-2;-1];x0=[];  
options=optimoptions('linprog','Algorithm','simplex','Display','iter');  
[x1,f1,exitflag1,output,lambda] = linprog(f,A,b,Aeq,beq,lb,ub,x0,options);  
b=2*[2;2;-1;-1];  
[x2,f2,exitflag2,output,lambda] = linprog(f,A,b,Aeq,beq,lb,ub,x0,options);  
b=3*[2;2;-1;-1];  
[x3,f3,exitflag3,output,lambda] = linprog(f,A,b,Aeq,beq,lb,ub,x0,options);
```

Es müssen keine Tests auf weitere lokale Lösungen durchgeführt werden, da es sich jeweils um ein LP handelt.

3. In der Relaxation ist ein LP zu lösen. Der beste Wert ergibt sich bei $n^* = 3, x_1^* = 6; x_2^* = 0$ mit $f^* = 12$.

Matlab:

```
A=[1 1 -2;1 -1 -2;-1 -1 1;-1 1 1];  
b=[0;0;0;0];Aeq=[];beq=[];ub=[Inf; Inf; 3];lb=[-Inf; -Inf; 1];f=[-2;-1;0];z0=[];
```

```
options=optimoptions('linprog','Algorithm','simplex','Display','iter');
[zs,fs,exitflag1,output,lambda] = linprog(f,A,b,Aeq,beq,lb,ub,z0,options);
```

Es müssen keine Tests auf weitere lokale Lösungen durchgeführt werden, da es sich jeweils um ein LP handelt.

7.3.2

1. In der Enumeration lösen wir 6 NLP. Die optimalen Werte sind: $x_1^* \approx 5.7306196$; $x_2^* \approx 5.40000$; $z^* = 12$ mit $f^* = -96.3061963$.

Matlab:

Skriptfile:

```
clear;global z; x0=[1;1]';A=[];b=[];Aeq=[];beq=[];lb=[];ub=[];
options = optimoptions('fmincon','Algorithm','sqp','GradObj','on','Display','iter');
z=1; [xs1,fs1,exitflag] = fmincon(@goalfunc,x0,A,b,Aeq,beq,lb,ub,@nonlincon,options);
z=2; [xs2,fs2,exitflag] = fmincon(@goalfunc,x0,A,b,Aeq,beq,lb,ub,@nonlincon,options);
z=4; [xs3,fs3,exitflag] = fmincon(@goalfunc,x0,A,b,Aeq,beq,lb,ub,@nonlincon,options);
z=6; [xs4,fs4,exitflag] = fmincon(@goalfunc,x0,A,b,Aeq,beq,lb,ub,@nonlincon,options);
z=8; [xs5,fs5,exitflag] = fmincon(@goalfunc,x0,A,b,Aeq,beq,lb,ub,@nonlincon,options);
z=12;[xs6,fs6,exitflag] = fmincon(@goalfunc,x0,A,b,Aeq,beq,lb,ub,@nonlincon,options);
```

Zielfunktion:

```
function [ f,grad ] = goalfunc(x)
global z
f=-10*x(1)-5*x(2)-z;
grad(1)=-10;
grad(2)=-5;
end
```

Nebenbedingungen:

```
function [c,ceq,cgrad,ceqgrad] = nonlincon(x)
global z
c(1)=x(1)*x(1)+x(2)*x(2)-z-50;
c(2)=-x(1)-1;
c(3)=-x(2)+3+z/5;
ceq=[];
cgrad(1,1)=2*x(1);
```

```

cgrad(1,2)=2*x(2);
cgrad(2,1)=-1;
cgrad(2,2)=0;
cgrad(3,1)=0;
cgrad(3,2)=-1;
cgrad=cgrad;
ceqgrad=[];
end

```

Wir können nicht sicher sein, ob diese gefundene Lösung korrekt ist, da jeweils die NLP weitere (bisher nicht aufgefundene) lokal optimale Lösungen besitzen können. Deshalb führen wir für jedes NLP stichprobenartig Tests mit Zufalls-Startwerten x_0 durch.

2. Mit der Relaxation ist ein NLP zu lösen. Ergebnis wie bei (a). Matlab: Skriptfile:

```

clear;x0=[1;1;1]';A=[];b=[];Aeq=[];beq=[];lb=[-Inf; -Inf;1];ub=[Inf;Inf;12];
options = optimoptions('fmincon','Algorithm','sqp','GradObj','on','Display','iter');
[xs,fs,exitflag] = fmincon(@goalfuncB,x0,A,b,Aeq,beq,lb,ub,@nonlinconB,options);

```

Zielfunktion:

```

function [ f,grad ] = goalfuncB(x)
f=-10*x(1)-5*x(2)-x(3);
grad(1)=-10;
grad(2)=-5;
grad(3)=-1;
end

```

Nebenbedingungen:

```

function [c,ceq,cgrad,ceqgrad] = nonlinconB(x)
c(1)=x(1)*x(1)+x(2)*x(2)-x(3)-50;
c(2)=-x(1)-1;
c(3)=-x(2)+3+x(3)/5;
ceq=[];
cgrad(1,1)=2*x(1);
cgrad(1,2)=2*x(2);
cgrad(1,3)=-1;
cgrad(2,1)=-1;
cgrad(2,2)=0;
cgrad(2,3)=0;
cgrad(3,1)=0;
cgrad(3,2)=-1;
cgrad(3,3)=0.2;

```

```

cgrad=cgrad;
ceqgrad=[];
end

```

Wir können nicht sicher sein, ob diese gefundene Lösung korrekt ist, da jeweils die NLP weitere (bisher nicht aufgefundene) lokal optimale Lösungen besitzen können. Deshalb führen wir für jedes NLP stichprobenartig Tests mit Zufalls-Startwerten x_0 durch.

7.3.3

1. Modellierung mittels einer MI-Optimierungsaufgabe mit $(x, y, z) \in \mathbb{R}^3$ und $n \in \{0; 1\}$:

$$\begin{aligned}
 \max(249x + 279y + 119z - 1000n), \\
 x &\geq 0 \\
 y &\geq 20(1 - n) \\
 y &\leq 200 - 180n \\
 z &\geq 0 \\
 2.124x^2 + 2.773y^2 + 24.98z &\leq 5000
 \end{aligned}$$

Dabei entspricht $n = 0$ dem Fall ohne Vertragsstrafe und $n = 1$ entspricht dem Fall mit Vertragsstrafe.

2. Lösung bzgl der Integer-Variablen mittels Enumeration; Lösung bzgl der reellwertigen Variablen mittels SQP. $n = 0$: $x_0^* \approx 12.3049$; $y_0^* \approx 20.0000$; $z_0^* \approx 142.8768$, mit dem maximalen Umsatz 25'646,- SFr. $n = 1$: $x_1^* \approx 12.3044$; $y_1^* \approx 10.5601$; $z_1^* \approx 174.9077$, mit dem maximalen Umsatz 25'824,- SFr. Somit ist die optimale Lösung des MI-Problems: $n^* = 1$; $x^* \approx 12.3044$; $y^* \approx 10.5601$; $z^* \approx 174.9077$. Matlab:

Skriptfile:

```

z0=[1;1;1]';A=[];b=[];Aeq=[];beq=[];
options = optimoptions('fmincon','Algorithm','sqp',...
    'GradObj','on','Display','iter',...
    'GradConstr','on','TolX',1e-6,'TolFun',1e-6);
lb=[8;20;0];ub=[];
[xs0,fs0,exitflag] = fmincon(@goalfunc_0,z0,A,b,Aeq,beq,lb,ub,@nonlincon,options);
lb=[8;0;0];ub=[Inf;20;Inf];
[xs1,fs1,exitflag] = fmincon(@goalfunc_0,z0,A,b,Aeq,beq,lb,ub,@nonlincon,options);

```

Zielfunktion für $n = 0$:

```
function [ f,grad ] = goalfunc_0(s)
x=s(1);y=s(2);z=s(3);
f=-249*x-279*y-119*z;
grad(1)=-249;grad(2)=-279;grad(3)=-119;
end
```

Zielfunktion für $n = 1$:

```
function [ f,grad ] = goalfunc_1(s)
x=s(1);y=s(2);z=s(3);
f=-249*x-279*y-119*z+1000;
grad(1)=-249;grad(2)=-279;grad(3)=-119;
end
```

Nebenbedingungen:

```
function [c,ceq,cgrad,ceqgrad] = nonlincon(s)
x=s(1);y=s(2);z=s(3);
c=2.124*x*x+2.773*y*y+24.98*z-5000;
ceq=[];
cgrad=zeros(3,1);
cgrad(1)=4.248*x;cgrad(2)=5.546*y;cgrad(3)=24.98;
ceqgrad=[];
end
```

Wir können nicht sicher sein, ob diese gefundene Lösung korrekt ist, da jeweils die NLP weitere (bisher nicht aufgefundene) lokal optimale Lösungen besitzen können. Deshalb führen wir für jedes NLP stichprobenartig Tests mit Zufalls-Startwerten x_0 durch.

3. Lösung mittels Relaxation: Skriptfile:

```
z0=[1;1;1;0]';A=[0 -1 0 -20;0 1 0 180];b=[-20;200];Aeq=[];beq=[];
options = optimoptions('fmincon','Algorithm','sqp',...
    'GradObj','on','Display','iter',...
    'GradConstr','on','TolX',1e-6,'TolFun',1e-6);
lb=[8;0;0;0];ub=[Inf;Inf;Inf;1];
[xs2,fs2,exitflag] = fmincon(@goalfunc_2,z0,A,b,Aeq,beq,lb,ub,@nonlincon_2,options);
```

Zielfunktion:

```
function [ f,grad ] = goalfunc_2(s)
x=s(1);y=s(2);z=s(3);n=s(4);
f=-249*x-279*y-119*z+1000*n;
```

```
grad(1)=-249;grad(2)=-279;
grad(3)=-119;grad(4)=1000;
end
```

Nebenbedingungen:

```
function [c,ceq,cgrad,ceqgrad] = nonlincon_2(s)
x=s(1);y=s(2);z=s(3);n=s(4);
c(1)=2.124*x*x+2.773*y*y+24.98*z-5000;
ceq=[];
cgrad=zeros(4,1);
cgrad(1)=4.248*x;cgrad(2)=5.546*y;
cgrad(3)=24.98;cgrad(4)=0;
ceqgrad=[];
end
```

Wir können nicht sicher sein, ob diese gefundene Lösung korrekt ist, da jeweils die NLP weitere (bisher nicht aufgefundene) lokal optimale Lösungen besitzen können. Deshalb führen wir für jedes NLP stichprobenartig Tests mit Zufalls-Startwerten x_0 durch.

7.3.4

Approximative Lösung mit der Relaxationsmethode. Optimaler Gewinn von 5'610'339 CHF ergibt sich bei dem Bau von 4 Fabriken. Skriptfile:

```
z0=[10;10;10]';A=[];b=[];Aeq=[];beq=[];lb=[0;0;0]';ub=[100;100;100];
options =optimoptions('fmincon','Algorithm','sqp','GradObj','on',...
'Display','iter','GradConstr','on','TolX',1e-6,'TolFun',1e-6);
[xs,fs,exitflag] =fmincon(@goalfunc,z0,A,b,Aeq,beq,lb,ub,@nonlincon,options);
xs,exitflag
```

Zielfunktion:

```
function [ f,grad ] = goalfunc(z)
x=z(1);y=z(2);n=z(3);
f=-0.12*x-0.16*y+0.1*n*n+0.1*n;
grad(1)=-0.12;grad(2)=-0.16;grad(3)=0.2*n+0.1;
end
```

Constraints:

```
function [c,ceq,cgrad,ceqgrad] = nonlincon(z)
x=z(1);y=z(2);n=z(3);
c=2.4*x*x+2.9*y*y-1000*n;ceq=[];
cgrad(1)=4.8*x;cgrad(2)=5.8*y;cgrad(3)=-1000;cgrad=cgrad';ceqgrad=[];
end
```



Kapitel 8

Ausblick

8.1.1

wahr sind 1,2,4, 6, 9, 12.

8.1.2

wahr sind 1, 2, 6, 9, 10, 12.

8.1.3

wahr ist 2, 8.

8.1.4

wahr sind 3, 9.

8.1.5

wahr sind 4, 7.

8.1.6

richtig sind 6, 10.

8 Ausblick

8.1.7

richtig ist 3.

8.1.8

richtig ist 3.



<http://www.springer.com/978-3-658-26609-7>

Optimierung für Wirtschaftsingenieure

Bünner, M.

2019, XX, 156 S. 5 Abb., Softcover

ISBN: 978-3-658-26609-7