



---

Wintersemester 2017/2018

---

## Lineare und kombinatorische Optimierung

### Übungsblatt 10

**Wir wünschen Ihnen ein frohes und gesundes neues Jahr 2018!**

#### Gruppenübungen

##### Aufgabe 1. (Simplex-Algorithmus)

(0 Punkte)

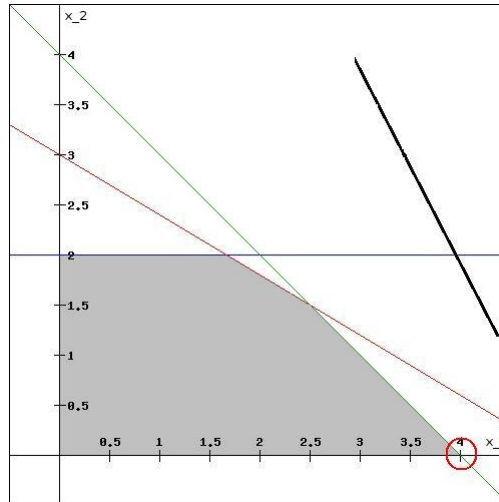
Gegeben sei folgendes lineare Optimierungsproblem:

$$\begin{aligned} \max_x \quad & 2x_1 + x_2 \\ \text{s.t.} \quad & x_2 \leq 2, \\ & 3x_1 + 5x_2 \leq 15, \\ & x_1 + x_2 \leq 4, \\ & x_1, x_2 \geq 0. \end{aligned}$$

1. Bestimmen Sie graphisch eine Optimallösung.
2. Bestimmen Sie eine Optimallösung mithilfe des Simplex-Algorithmus. Starten Sie im Punkt  $(0, 0)$  und benutzen Sie die Erkenntnisse aus dem ersten Aufgabenteil, um die Anzahl der benötigten Iterationen klein zu halten.

#### Lösung:

- (i) Die zulässige Menge sowie eine Niveaulinie der Zielfunktion sind in folgender Skizze dargestellt:



Optimalpunkt ist  $(x_1, x_2) = (4, 0)$ .

(ii) Das Problem in Standardform lautet

$$\begin{aligned} \min \quad & -2x_1 - x_2 \\ \text{s.t.} \quad & x_2 + x_3 = 2 \\ & 3x_1 + 5x_2 + x_4 = 15 \\ & x_1 + x_2 + x_5 = 4 \\ & x_1, x_2, x_3, x_4, x_5 \geq 0. \end{aligned}$$

*It.1:* Wir beginnen mit  $x = (0, 0, 2, 15, 4)$ ,  $B = (3, 4, 5)$  und  $N = (1, 2)$ .

**BTRAN:**

$$\bar{y}^T A_B = c_B^T \Rightarrow \bar{y}^T \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = (0, 0, 0) \Rightarrow \bar{y} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}.$$

**Pricing:**

$$z_N = c_N - A_N^T \bar{y} = \begin{pmatrix} -2 \\ -1 \end{pmatrix} - 0 = \begin{pmatrix} -2 \\ -1 \end{pmatrix}.$$

Da  $z_1 < 0$ , wählen wir  $N_1 = j = 1$  in die Basis.

(Wir könnten hier natürlich auch  $j = 2$  in die Basis wählen. Das würde bedeuten, dass  $x_1 = 0$  bleibt und wir  $x_2$  erhöhen. Aus der Zeichnung sehen wir, dass wir dadurch in Richtung des Punktes  $(0, 2)$  laufen. Dies ist aber die falsche Richtung, da die Optimallösung in  $(4, 0)$  liegt.)

**FTRAN:**

$$A_B w = A_{\cdot j} \Rightarrow \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} w = \begin{pmatrix} 0 \\ 3 \\ 1 \end{pmatrix} \Rightarrow w = \begin{pmatrix} 0 \\ 3 \\ 1 \end{pmatrix}.$$

**Ratio-Test:**

$$\gamma = x_{B_3} / w_3 = \min\{15/3, 4/1\} = 4.$$

Also verlässt  $B_3 = 5$  die Basis.

**Update:**

$$x_B = x_B - \gamma w = \begin{pmatrix} 2 \\ 15 \\ 4 \end{pmatrix} - 4 \begin{pmatrix} 0 \\ 3 \\ 1 \end{pmatrix} = \begin{pmatrix} 2 \\ 3 \\ 0 \end{pmatrix},$$

$$N = (5, 2),$$

$$B = (3, 4, 1),$$

$$x_1 = 4.$$

It.2: Neue Basislösung ist also  $x_B = \begin{pmatrix} 2 \\ 3 \\ 4 \end{pmatrix}$ ,

**BTRAN:**

$$\bar{y}^T A_B = c_B^T \Rightarrow \bar{y}^T \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 3 \\ 0 & 0 & 1 \end{pmatrix} = (0, 0, -2) \Rightarrow \bar{y} = \begin{pmatrix} 0 \\ 0 \\ -2 \end{pmatrix}.$$

**Pricing:**

$$z_N = c_N - A_N^T \bar{y} = \begin{pmatrix} 0 \\ -1 \end{pmatrix} - \begin{pmatrix} 0 & 0 & 1 \\ 1 & 5 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ -2 \end{pmatrix} = \begin{pmatrix} 2 \\ 1 \end{pmatrix}.$$

Da  $z_N \geq 0$ : Lösung optimal.

Damit ist  $x^* = \begin{pmatrix} 4 \\ 0 \\ 2 \\ 3 \\ 0 \end{pmatrix}$  die Optimallösung mit ZF-Wert 8.

**Aufgabe 2.** (Umformulierungen)

(0 Punkte)

1. Betrachten Sie die Funktion  $f(x) = \max\{c^T x + \alpha, d^T x + \beta\}$ . Formulieren das Optimierungsproblem

$$\min\{f(x) : Ax = b, x \geq 0\}$$

als lineares Optimierungsproblem. Dabei seien  $\alpha, \beta \in \mathbb{R}$ ,  $b \in \mathbb{R}^m$ ,  $c, d \in \mathbb{R}^n$ ,  $A \in \mathbb{R}^{m \times n}$ .

2. Zum näherungsweise Lösen überbestimmter Gleichungssysteme  $Ax = b$ ,  $A \in \mathbb{R}^{m \times n}$ ,  $x \in \mathbb{R}^n$ ,  $b \in \mathbb{R}^m$ ,  $m > n$ , wird oft ein Optimierungsproblem formuliert, in dem das Residuum bezüglich einer gegebenen Norm minimiert werden soll:

$$\min_{x \in \mathbb{R}^n} \|Ax - b\|.$$

Formulieren Sie dieses Problem als lineares Programm für

- die Maximumsnorm

$$\|v\|_\infty := \max_{i=1, \dots, m} |v_i|$$

- und die Summennorm

$$\|v\|_1 := \sum_{i=1}^m |v_i|.$$

**Lösung:**

- (A) Das Problem kann als lineares Optimierungsproblem formuliert werden, indem man eine zusätzliche Variable  $y \in \mathbb{R}$  einführt. Bezeichnet man  $y = \max\{c^T x + \alpha, d^T x + \beta\}$ , so besteht das Optimierungsproblem darin,  $y$  zu minimieren. Als Nebenbedingungen erhält man aus der Definition von  $y$ :

$$c^T x + \alpha \leq y \quad \text{und} \quad d^T x + \beta \leq y.$$

Das LP lautet dann:

$$\begin{array}{ll} \min & y \\ \text{s.t.} & y - c^T x \geq \alpha \\ & y - d^T x \geq \beta \\ & Ax = b \\ & x \geq 0 \end{array}$$

- (B) (a) Wie in (A) führen wir eine zusätzliche Variable  $t \in \mathbb{R}$  ein und setzen  $t = \|Ax - b\|_\infty = \max_{i=1 \dots m} |A_i \cdot x - b_i|$ .  
Damit ergibt sich das folgende lineare Programm:

$$\begin{array}{ll} \min & t \\ \text{s.t.} & -t \leq A_i \cdot x - b_i \leq t, i = 1 \dots m \\ & t \geq 0 \end{array}$$

- (b) Zum Minimieren der Summe  $\sum_{i=1}^m |A_i \cdot x - b_i|$  führen wir diesmal einen Vektor  $t \in \mathbb{R}^m$  ein mit  $t_i = |A_i \cdot x - b_i|$ . Damit erhalten wir das lineare Programm

$$\begin{array}{ll} \min & \sum_{i=1}^m t_i \\ \text{s.t.} & -t_i \leq A_i \cdot x - b_i \leq t_i, i = 1 \dots m \\ & t \geq 0 \end{array}$$

### Aufgabe 3. (Modellierung)

(0 Punkte)

Ein Unternehmen stellt zwei Gürteltypen A und B her. A ist von besserer Qualität als B. Der Nettogewinn beträgt bei A 2 Geldeinheiten und bei B 1.5 Geldeinheiten. Der Zeitaufwand für die Produktion eines Gürtels vom Typ A beträgt 2 Zeiteinheiten. Für den Typ B wird 1 Zeiteinheit pro Gürtel benötigt. Täglich stehen maximal 1000 Zeiteinheiten zur Verfügung. Die Lederbelieferung erlaubt eine Produktion von 800 Gürteln pro Tag, egal um welchen Typ es sich handelt. Außerdem stehen pro Tag höchstens 400 Schnallen für den Typ A und 700 Schnallen für den Typ B zur Verfügung.

Wie soll die Produktion aufgeteilt werden, damit ein maximaler Gewinn erzielt wird? Modellieren Sie diese Problemstellung als Optimierungsproblem und lösen Sie es graphisch.

### Lösung:

Sei  $x_1$  die Anzahl der Gürtel vom Typ A,  $x_2$  die Anzahl der Gürtel vom Typ B, die hergestellt werden sollen. Dann lautet das lineare Optimierungsproblem:

$$\begin{array}{ll} \max & 2x_1 + 1.5x_2 \\ \text{s.t.} & 2x_1 + x_2 \leq 1000 \\ & x_1 + x_2 \leq 800 \\ & x_1 \leq 400 \\ & x_2 \leq 700 \\ & x_1, x_2 \geq 0 \\ & x_1, x_2 \in \mathbb{Z} \end{array}$$

Die zulässige Menge ist in Abbildung 1 dargestellt. Die Optimallösung erhält man, indem man eine Niveaulinie der Zielfunktion soweit in Richtung  $(2, 1.5)$  schiebt, dass sie die zulässige Menge gerade noch berührt. Der Schnitt der Niveaulinie mit der zulässigen Menge ist dann die Menge der Optimallösungen.

In diesem Beispiel gibt es einen eindeutigen Kandidaten für die Optimallösung, nämlich den Punkt  $(200, 600)^T$ . Da der Punkt ganzzahlig ist, wird dort die Optimallösung angenommen. Der zugehörige Optimalwert lautet 1300. Es sollten also 200 Stück Gürtel vom Typ A und 600 Stück vom Typ B hergestellt werden, um den maximalen Gewinn von 1300 Geldeinheiten zu realisieren.

## Hausübungen

### Aufgabe 4. (Simplex-Algorithmus)

(4 Punkte)

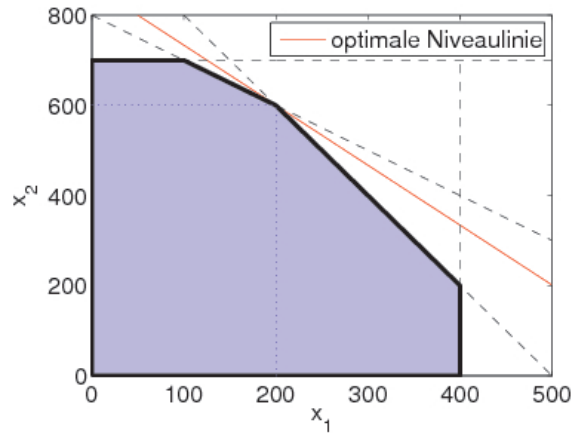


Abbildung 1: Die zulässige Menge und die optimale Niveaulinie.

Betrachten Sie das folgende lineare Optimierungsproblem:

$$\begin{aligned} \max_x \quad & 150x_1 + 450x_2 \\ \text{s.t.} \quad & x_1 \leq 120, \\ & x_2 \leq 70, \\ & x_1 + x_2 \leq 140, \\ & x_1 + 2x_2 \leq 180, \\ & x_1, x_2 \geq 0. \end{aligned}$$

Lösen Sie das Problem mit dem Simplex-Algorithmus für den Startpunkt  $(120, 20)$ . Erstellen Sie eine Skizze der zulässigen Menge, die das LP beschreibt und zeichnen Sie jede Basislösung ein. Interpretieren Sie jeden Basisaustauschschritt anhand der Zeichnung.

**Lösung:**

Zunächst schreiben wir das Programm in Standardform (durch Einführen von Schlupfvariablen):

$$\begin{aligned} \min_x \quad & -150x_1 - 450x_2 \\ \text{s.t.} \quad & x_1 + x_3 = 120 \\ & x_2 + x_4 = 70 \\ & x_1 + x_2 + x_5 = 140 \\ & x_1 + 2x_2 + x_6 = 180 \\ & x_1, x_2, x_3, x_4, x_5, x_6 \geq 0. \end{aligned}$$

*It.1:* Wir starten mit  $x = (120, 20, 0, 50, 0, 20)$ ,  $B = \{1, 2, 4, 6\}$ ,  $N = \{3, 5\}$ . Die Basislösung ist also

$$x_B = \begin{pmatrix} 120 \\ 20 \\ 50 \\ 20 \end{pmatrix}, \text{ mit}$$

$$A_{.B} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 2 & 0 & 1 \end{pmatrix}, A_{.N} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix}, c_B = \begin{pmatrix} -150 \\ -450 \\ 0 \\ 0 \end{pmatrix}, c_N = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

**BTRAN:**

$$A_B^T \bar{y} = c_B \Leftrightarrow \begin{pmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \bar{y} = \begin{pmatrix} -150 \\ -450 \\ 0 \\ 0 \end{pmatrix} \Rightarrow \bar{y} = \begin{pmatrix} 300 \\ 0 \\ -450 \\ 0 \end{pmatrix}.$$

**Pricing:**

$$z_N = \begin{pmatrix} 0 \\ 0 \end{pmatrix} - \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 300 \\ 0 \\ -450 \\ 0 \end{pmatrix} = \begin{pmatrix} -300 \\ 450 \end{pmatrix}$$

Da  $z_1 < 0$ , nehmen wir  $N_1 = j = 3$  in die Basis.

**FTRAN:**

$$A_B w = A_{\cdot j} \Leftrightarrow \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 2 & 0 & 1 \end{pmatrix} w = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \Rightarrow w = \begin{pmatrix} 1 \\ -1 \\ 1 \\ 1 \end{pmatrix}$$

**Ratio-Test:**

$$\gamma = x_{B_4}/w_4 = \min\{120/1, 50/1, 20/1\} = 20$$

Also verlässt  $B_4 = 6$  die Basis.

**Update:**

$$\begin{aligned} x_B &= x_B - \gamma w = \begin{pmatrix} 120 \\ 20 \\ 50 \\ 20 \end{pmatrix} - 20 \begin{pmatrix} 1 \\ -1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 100 \\ 40 \\ 30 \\ 0 \end{pmatrix} \\ N &= \{6, 5\} \\ B &= \{1, 2, 4, 3\} \\ x_3 &= 20 \end{aligned}$$

*It.2:* Die nächste Basislösung ist also  $x_B = \begin{pmatrix} 100 \\ 40 \\ 30 \\ 20 \end{pmatrix}$ , mit

$$A_{\cdot B} = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 2 & 0 & 0 \end{pmatrix}, A_{\cdot N} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 1 \\ 1 & 0 \end{pmatrix}, c_B = \begin{pmatrix} -150 \\ -450 \\ 0 \\ 0 \end{pmatrix}, c_N = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

**BTRAN:**

$$A_B^T \bar{y} = c_B \Leftrightarrow \begin{pmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 2 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \bar{y} = \begin{pmatrix} -150 \\ -450 \\ 0 \\ 0 \end{pmatrix} \Rightarrow \bar{y} = \begin{pmatrix} 0 \\ 0 \\ 150 \\ -300 \end{pmatrix}.$$

**Pricing:**

$$z_N = \begin{pmatrix} 0 \\ 0 \end{pmatrix} - \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 150 \\ -300 \end{pmatrix} = \begin{pmatrix} 300 \\ -150 \end{pmatrix}$$

Da  $z_2 < 0$ :  $N_2 = 5$  kommt in die Basis

**FTRAN:**

$$A_B w = A_{\cdot j} \Leftrightarrow \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 2 & 0 & 0 \end{pmatrix} w = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \Rightarrow w = \begin{pmatrix} 2 \\ -1 \\ 1 \\ -2 \end{pmatrix}$$

**Ratio-Test:**

$$\gamma = x_{B_3}/w_3 = \min\{100/2, 30/1\} = 30$$

Also verlässt  $B_3 = 4$  die Basis.

**Update:**

$$\begin{aligned} x_B &= x_B - \gamma w = x_B = \begin{pmatrix} 100 \\ 40 \\ 30 \\ 20 \end{pmatrix} - 30 \begin{pmatrix} 2 \\ -1 \\ 1 \\ -2 \end{pmatrix} = \begin{pmatrix} 40 \\ 70 \\ 0 \\ 80 \end{pmatrix} \\ N &= \{6, 4\} \\ B &= \{1, 2, 5, 3\} \\ x_5 &= 30 \end{aligned}$$

It.3: Die nächste Basislösung ist also  $x_B = \begin{pmatrix} 40 \\ 70 \\ 30 \\ 80 \end{pmatrix}$ , mit

$$A_{.B} = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 2 & 0 & 0 \end{pmatrix}, A_{.N} = \begin{pmatrix} 0 & 0 \\ 0 & 1 \\ 0 & 0 \\ 1 & 0 \end{pmatrix}, c_B = \begin{pmatrix} -150 \\ -450 \\ 0 \\ 0 \end{pmatrix}, c_N = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

**BTRAN:**

$$A_{.B}^T \bar{y} = c_B \Leftrightarrow \begin{pmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 2 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \bar{y} = \begin{pmatrix} -150 \\ -450 \\ 0 \\ 0 \end{pmatrix} \Rightarrow \bar{y} = \begin{pmatrix} 0 \\ -150 \\ 0 \\ -150 \end{pmatrix}.$$

**Pricing:**

$$z_N = \begin{pmatrix} 0 \\ 0 \end{pmatrix} - \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ -150 \\ 0 \\ -150 \end{pmatrix} = \begin{pmatrix} 150 \\ 150 \end{pmatrix}$$

Da  $z_N \geq 0$ : STOP: Lösung optimal!

Die Optimallsg. lautet also

$$x^* = \begin{pmatrix} 40 \\ 70 \\ 80 \\ 0 \\ 30 \\ 0 \end{pmatrix}$$

Der Algorithmus nimmt die in der Abbildung 2 zu sehende Eckenwanderung vor. Wenn eine der Schlupfvariablen 0 wird bedeutet dies, dass die zugehörige Ungleichung aktiv wird. Wenn eine Schlupfvariable in die Basis kommt, und dann nicht mehr 0 ist, bedeutet dies, dass die zugehörige Ungleichung inaktiv wird.

**Aufgabe 5.** (Unbeschränktheit von Linearen Programmen)

(2+2 Punkte)

Es seien  $c \in \mathbb{R}^n, A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m$ . Beweisen oder widerlegen Sie die beiden folgenden Aussagen:

1. Ist ein lineares Problem der Form  $\max\{c^\top x : Ax \leq b, x \geq 0\}$  unbeschränkt, dann gibt es einen Index  $k$ , so dass das Problem  $\max\{c_k x_k : Ax \leq b, x \geq 0\}$  unbeschränkt ist.
2. Gibt es einen Index  $k$ , so dass das Problem  $\max\{c_k x_k : Ax \leq b, x \geq 0\}$  unbeschränkt ist, dann ist auch  $\max\{c^\top x : Ax \leq b, x \geq 0\}$  unbeschränkt.

**Lösung:**

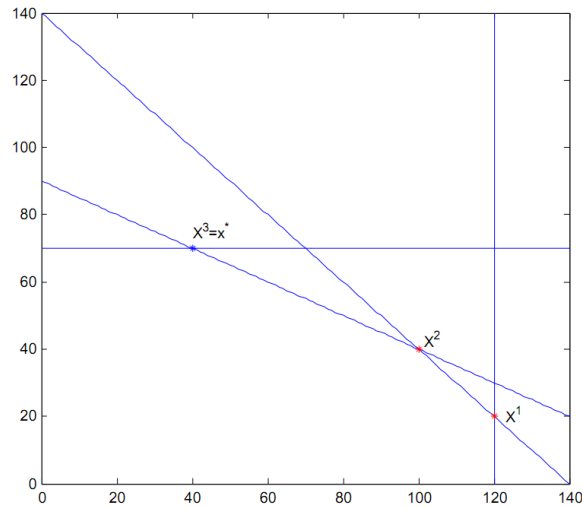


Abbildung 2: Eckenwanderung

1. gilt. Beweis indirekt: Angenommen, für alle Indizes  $k$  ist  $\max\{c_k x_k : Ax \leq b, x \geq 0\}$  beschränkt. Dann gibt es  $M_k \in \mathbb{R}$  mit

$$c_k x_k \leq M_k \quad \forall x \geq 0, Ax \leq b.$$

Dann folgt:

$$c^T x = \sum_k c_k x_k \leq \sum_k M_k < \infty \quad \forall x \geq 0, Ax \leq b.$$

Dann gilt aber auch

$$\max\{c^T x : Ax \leq b, x \geq 0\} < \infty.$$

2. gilt nicht. Ein Gegenbeispiel ist:

$$\max\{x_1 - x_2 : x_1 - x_2 = 0, x_1 \geq 0, x_2 \geq 0\}.$$

#### Aufgabe 6. (Modellierung)

(4 Punkte)

Eine Firma hat sich auf die Fertigung von zwei speziellen Computertypen spezialisiert: Computer mit Ein-Prozessor-System (1 CPU) und Computer mit Zwei-Prozessor-System (2 CPUs). Pro Woche können von den Ein-Prozessor-Systemen maximal 120 Stück hergestellt werden, von den Zwei-Prozessor-Systemen maximal 70 Stück. Insgesamt können pro Woche nur 140 Computer hergestellt werden und es stehen pro Woche höchstens 180 CPUs zur Verfügung.

1. In welcher Weise muss produziert werden, damit der Gesamtgewinn maximal ist, wenn ein Ein-Prozessorsystem 150 Euro Gewinn einbringt und ein Zwei-Prozessor-System 450 Euro Gewinn einbringt? Stellen Sie das lineare Programm auf und lösen Sie es graphisch.
2. Im folgenden Monat sinkt der Gewinn für Zwei-Prozessor-Systeme von 450 auf ebenfalls 150 Euro. Wie ändert sich der maximale Gesamtgewinn des Betriebes?

#### Lösung:

1. Sei  $x_1$  die Anzahl der 1-CPU-Systeme,  $x_2$  die Anzahl der 2-CPU-Systeme. Es müssen folgende Nebenbedingungen erfüllt sein:

$$x_1 \leq 120$$

$$x_2 \leq 70$$

$$x_1 + x_2 \leq 140$$

$$x_1 + 2x_2 \leq 180$$

$$x_1, x_2 \geq 0$$

$$x_1, x_2 \in \mathbb{Z}$$



Dabei maximieren wir die Zielfunktion  $f_a(x) = 150x_1 + 450x_2$ . Um das Problem graphisch zu lösen, zeichnen wir das Polyeder, das durch die Nebenbedingungen beschrieben wird. Wir schieben nun die Niveaulinie (gestrichelte Linie) soweit in Richtung des Zielfunktionsgradienten  $c_1 = (150, 450)^T$  (Pfeil mit Beschriftung 'a)'), bis diese den Zulässigkeitsbereich verlässt. Der Punkt  $x^* = (40, 70)$  ist somit optimal für das Problem ohne Ganzzahligkeitsbedingung. Da er auch noch ganzzahlig ist, ist er die Optimallösung. Der maximale Gewinn ist hier also  $40 \cdot 150 + 70 \cdot 450$  Euro.

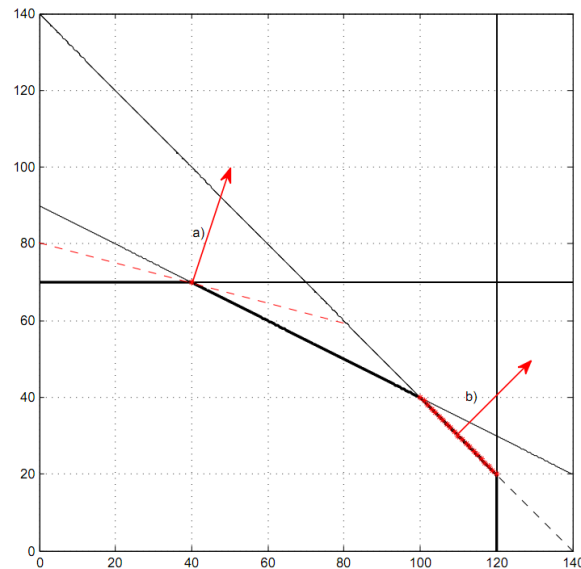


Abbildung 3: Das Polyeder mit den Zielfunktions-Gradienten aus a) und b).

- Der Zielfunktionsgradient ist nun  $c_2 = (150, 150)^T$ . Er steht senkrecht auf der Nebenbedingung  $x_1 + x_2 \leq 140$  (Pfeil mit Beschriftung 'b)'). Alle Punkte mit  $100 \leq x_1 \leq 120$  und  $x_1 + x_2 = 140$  wären also optimal für das Problem ohne Ganzzahligkeit. Damit sind für unser Problem alle ganzzahligen Punkte davon optimal, also alle  $(x_1, x_2)$  mit  $x_1 \in \{100, 101, \dots, 120\}$  und  $x_2 = 140 - x_1$ , also  $(x_1, x_2) \in \{(100, 40), (101, 39), \dots, (120, 20)\}$ . Der maximale Gewinn ist hier  $(x_1 + x_2) \cdot 150 = 140 \cdot 150$  Euro.

#### Aufgabe 7. (Programmieraufgabe: Min-Cost-Flow-Problem als LP lösen)

(4 Punkte)

Laden Sie die Datei `ProgAufgabe10.py` aus Studon herunter und fügen Sie sie in ihr PyCharm ein.

In dieser Aufgabe implementieren Sie eine Methode, die ein übergebenes Min-Cost-Flow-Problem in Gurobi modelliert und löst.

Implementieren Sie hierfür die Methode `min_cost_flow(G)`. Der Parameter `G` ist ein gerichteter networkx-Graph, auf dem das Min-Cost-Flow-Problem gelöst werden soll. Der Graph hat das Knotenattribut `demand`, das den Bedarf im jeweiligen Knoten darstellt (negativer Bedarf ist Angebot), das Kantenattribut `capacity`, das die Kapazität der Kante angibt, und das Kantenattribut `cost`, das die Kosten für das Transportieren entlang der Kante angibt. Die LP-Formulierung dieses Min-Cost-Flow-Problems kennen Sie bereits aus der Vorlesung bzw. Übung.

Nutzen Sie nun die aus der letzten Programmieraufgabe bekannten Methoden, um dieses LP in `gurobipy` zu formulieren und zu lösen, d.h. nutzen Sie `for`-Schleifen, um für die Kanten und Knoten die jeweiligen Variablen und Nebenbedingungen zu spezifizieren.

TIPPS:

- Verwenden Sie Dictionaries, um die erzeugten Gurobivariablen abzuspeichern. Auf diese Weise können Sie die Variablen den jeweiligen Kanten zuordnen.
- Ist `list` eine Liste von Gurobi-Variablen, so erzeugt `grb.quicksum(list)` die Summe dieser Variablen. Dies benötigt man z.B. um für die Flusserspartheitsbedingung die Summe der eingehenden und die Summe der ausgehenden Flussvariablen zu berechnen.

**Lösung:**

Die Musterlösung der Programmieraufgabe finden Sie im StudOn in der Datei `ProgAufgabe10_Loesung.py`.

Gruppe 1: Abgabe der Hausaufgaben am 16.01.2018 in der Übung.  
Gruppe 2+3: Abgabe der Hausaufgaben am 17.01.2018 in der jeweiligen Übung.