

Optimierung für Nichtmathematiker Übung 2

Modellierung von Optimierungsproblemen:

1. Lineares Optimierungsproblem:

Ein Zweitaktmotor wird mit einem Öl-Benzin-Gemisch betrieben. Dabei ist zu beachten, dass dieses Gemisch mindestens 4 Prozent Öl enthält, aber auch mindestens 85 Prozent Benzin. Das Gemisch soll möglichst kostengünstig sein. Formuliere eine passende Optimierungsaufgabe, wenn man 100 Liter Kraftstoff braucht!

2. Ganzzahliges Optimierungsproblem:

In einer Tischlerei werden unter anderem drei Sorten Tische produziert. Die Lieferung einer gewissen Mindestanzahl von Tischen wurde bereits vertraglich vereinbart. Es ist ein Modell zu erstellen, das einen maximalen Gewinn realisiert und die angegebenen Zeit- bzw. Materialkapazitäten einhält!

in gewissen Einheiten	Tisch 1	Tisch 2	Tisch 3	Kap.
Gewinn je Stück	3	1	2	
Zeitaufwand je Stück	2	1	1	40
Materialaufwand je Stück	4	2	3	100
vereinbarte Menge	3	2	2	

Umformen von Linearen Optimierungsproblemen (kanonische Form/Normalform):

3. Bringe folgendes lineare Optimierungsproblem in die kanonische Form und in die Normalform

$$\begin{aligned} \max \quad & x_1 + x_2 \\ \text{s.t.} \quad & x_1 - x_2 \geq 3 \\ & x_1 + 2x_2 = 1 \\ & -1 \leq x_2 \leq 1 \\ & x_1, x_2 \in \mathbb{R}. \end{aligned}$$

4. Bringe folgendes lineare Optimierungsproblem in die Normalform

$$\begin{aligned} \min \quad & -2x_1 - 3x_2 - 4x_3 \\ \text{s.t.} \quad & x_1 + x_2 + x_3 \leq 4 \\ & 3x_2 + x_3 \leq 6 \\ & x_1 \leq 2, x_3 \leq 3 \\ & x_1, x_2, x_3 \in \mathbb{R}. \end{aligned}$$

Einführung in die Modellierungssprache AMPL:

Wir betrachten zunächst das Mozartproblem aus der Vorlesung:

$$\begin{aligned} \max \quad & 3x_K + 2x_T \\ \text{s.t.} \quad & 2x_K + x_T \leq 10 \\ & x_K + x_T \leq 6 \\ & x_K + 2x_T \leq 9 \\ & x_K \geq 0, x_T \geq 0 \end{aligned}$$

Einige AMPL-Befehle am Beispiel:

```
var xK;  
var xT;
```

```
maximize Profit: 3*xK + 2*xT;
```

```
subject to Marzipan: 2*xK + 1*xT <= 10;  
subject to Nougat: 1*xK + 1*xT <= 6;
```

```
subject to Edelherb: 1*xK + 2*xT <= 9;
```

```
subject to Schranke_xK: 0 <= xK;
```

```
subject to Schranke_xT: 0 <= xT;
```

- Dateien, die Modelle enthalten, werden als *.mod abgespeichert.
- Alle Zeilen müssen mit einem Semikolon enden.
- Alle vergebenen Namen dürfen maximal einmal vorkommen.
- Kommentare mit #: # Das ist ein Kommentar.
- Vergleiche: =, <=, >=
- Definition von Variablen: var variablenname;
- Zielfunktion: minimize/maximize zielfunktionsname: ... ;
- Nebenbedingung: subject to nebenbedingungsname: ... ;
- Aufruf/Lösen mit AMPL in der Kommandozeile:

```
    ampl
    model modelname.mod;
    solve;
```

Die obige Darstellungsweise wird leicht unübersichtlich. Außerdem ist es aufwändig, wenn man das Modell z. B. um weitere Produktvarianten erweitern will. Hierbei wäre es günstiger, wenn man die Zielfunktion und die Nebenbedingungen einmal allgemein aufschreibt und zusätzlich die konkreten Werte über Dateien einlesen kann. AMPL bietet diese Möglichkeit an:

- Modelldatei: moztart2.mod

```
set P; # Produkte

param a1 { i in P }; # fuer Marzipan
param a2 { i in P }; # fuer Nougat
param a3 { i in P }; # fuer Edelherb
param c { i in P }; # Zielfunktion
param b { i in 1..3 }; # rechte Seiten

var x { i in P }; # Variablen

maximize Profit: sum { i in P } c[i] * x[i];

subject to Marzipan: sum { i in P } a1[i] * x[i] <= b[1];
subject to Nougat: sum { i in P } a2[i] * x[i] <= b[2];
subject to Edelherb: sum { i in P } a3[i] * x[i] <= b[3];

subject to Schranken { i in P }: 0 <= x[i];
```

- Datendatei: moztart2.dat

```
set P := Kugel Taler;
param: a1 a2 a3 c:=
Kugel 2 1 1 3
Taler 1 1 2 2;
param: b:=
1 10
2 6
3 9 ;
```

Neue Befehle:

- Definition von (Index-)Mengen: `set setname;`
- Parameter: `param paramname { i in 1..number };`
oder `param paramname { i in setname };`
- auch Definition von mehreren Variablen über Indexmengen möglich
- Zugriff auf Elemente: `x[1]` entspricht x_1
- Summe: `sum { i in setname } ... ;`

Lösen der Probleme:

- Aufruf/Lösen mit AMPL in der Kommandozeile:

```
ampl
model modelname.mod;
data dataname.dat;
solve;
```

Aufgabe: Ändern Sie die Datei `mozart2.dat` so ab, dass es neben den Kugeln und Talern auch noch Stangen gibt, die einen Gewinn von 6 bringen und für die jeweils drei Einheiten Marzipan und je eine Einheit Nougat und Edelherb benötigt werden.