

Modellierung und Optimierung mit OPL

3 Techniken der binären Modellierung

Andreas Popp



Die Big-M-Methode

OPL: Modellieren von Zeitperioden

Disjunktive Nebenbedingungen

3.4 Stückweise definierte Funktionen

Treppenfunktionen

Stückweise lineare Funktionen

OPL: Der piecewise-Befehl

3.1 Abbildung logischer Ausdrücke

Die Big-M-Methode

OPL: Modellieren von
Zeitperioden

Disjunktive
Nebenbedingungen

3.4 Stückweise definierte Funktionen

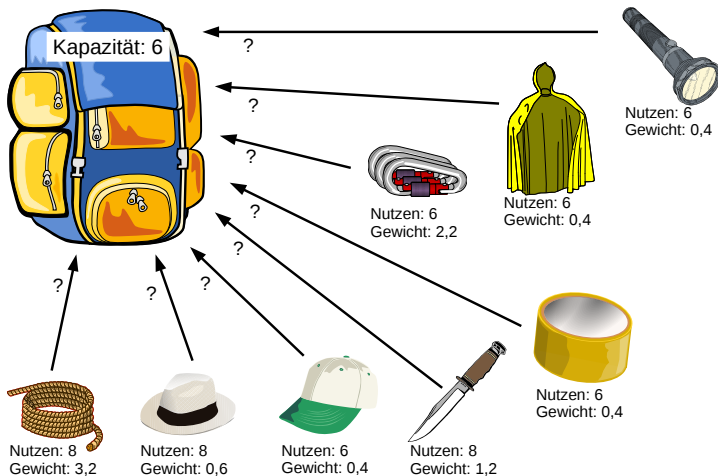
Treppenfunktionen

Stückweise lineare
Funktionen

OPL: Der piecewise-Befehl

3.1 Abbildung logischer Ausdrücke

Beispiel: Adventure Inc.



3 Techniken der
binären
Modellierung

CC-BY-SA
A. Popp

3.1 Abbildung
logischer
Ausdrücke

3.2 Entscheidungs-
abhängige
Nebenbedingungen

Die Big-M-Methode

OPL: Modellieren von
Zeiträumen

Disjunktive
Nebenbedingungen

3.3 OPL:
Kompakte
Schreibweise

3.4 Stückweise
definierte
Funktionen

Treppenfunktionen

Stückweise lineare
Funktionen

OPL: Der piecewise-Befehl

Modell: Rucksackproblem

3 Techniken der
binären
Modellierung

CC-BY-SA
A. Popp

Indexmengen:

I Menge der Gegenstände

Parameter:

w_i Gewicht von Gegenstand $i \in I$

u_i Nutzen von Gegenstand $i \in I$

c Kapazität des Rucksacks

Entscheidungsvariablen:

x_i Binäre Entscheidungsvariable; zeigt an ob
Gegenstand $i \in I$ eingepackt wird

Modellbeschreibung:

$$\begin{aligned} \max \quad & \sum_{i \in I} u_i \cdot x_i \\ \text{s.t.} \quad & \sum_{i \in I} w_i \cdot x_i \leq c \\ & x_i \in \{0,1\} \quad \forall i \in I \end{aligned} \quad (\text{I})$$

3.1 Abbildung
logischer
Ausdrücke

3.2 Entscheidungs-
abhängige
Nebenbedingungen

Die Big-M-Methode

OPL: Modellieren von
Zeitperioden

Disjunktive
Nebenbedingungen

3.3 OPL:
Kompakte
Schreibweise

3.4 Stückweise
definierte
Funktionen

Treppenfunktionen

Stückweise lineare
Funktionen

OPL: Der piecewise-Befehl

- Die logische **Negation**
- ∧ Das logische **Und**
- ∨ Das logische **Oder**
- ⊕ Das logische **exklusive Oder** („Entweder-Oder“)
- ⇒ Die logische **Implikation**
- ⇔ Die logische **Äquivalenz**

Wahrheitstafel in numerischer Beschreibung

A	B	$\neg A$	$\neg B$	$A \wedge B$	$A \vee B$	$A \underline{\vee} B$	$A \Rightarrow B$	$A \Leftrightarrow B$
1	1	0	0	1	1	0	1	1
1	0	0	1	0	1	1	0	0
0	1	1	0	0	1	1	1	0
0	0	1	1	0	0	0	1	1

3.1 Abbildung logischer Ausdrücke

Die Big-M-Methode

OPL: Modellieren von Zeitperioden

Disjunktive
Nebenbedingungen

3.4 Stückweise definierte Funktionen

Treppenfunktionen

Stückweise lineare
Funktionen

OPL: Der piecewise-Befehl

Logische Verknüpfungen in binären Optimierungsmodellen

Beispiel: Seien x_1 und x_2 binäre Entscheidungsvariablen eines Rucksackproblems, die angeben, ob die Gegenstände I_1 und I_2 eingepackt werden.

$\neg h_1$: Man möchte wissen ob h_1 nicht eingepackt ist.

► $1 - x_1$

$I_1 \wedge I_2$: Sowohl I_1 als auch I_2 müssen eingepackt werden.

► $x_1 + x_2 = 2$

$I_1 \vee I_2$: Mindestens einer der Gegenstände muss eingepackt werden.

► $x_1 + x_2 > 1$

$\neg(h_1 \wedge h_2)$: Höchstens einer der Gegenstände darf eingepackt werden.

► $x_1 + x_2 < 1$

3.2 Entscheidungsabhängige Nebenbedingungen

3.2 Entscheidungsabhängige Nebenbedingungen

Die Big-M-Methode

OPL: Modellieren von Zeitperioden

Disjunktive
Nebenbedingungen

3.4 Stückweise definierte Funktionen

Treppenfunktionen

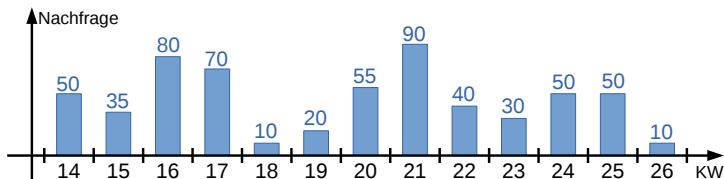
Stückweise lineare
Funktionen

OPL: Der piecewise-Befehl

Beispiel: Lewig Wakuxi

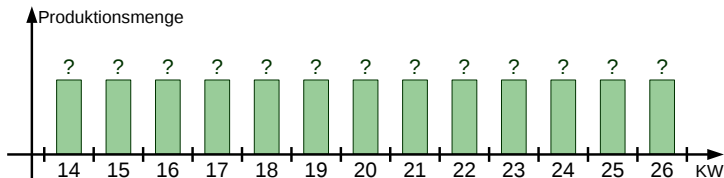
3 Techniken der
binären
Modellierung

CC-BY-SA
A. Popp



Rüstkosten pro Periode: 25.000€

Lagerkosten von einer Periode auf die nächste pro Einheit: 100€



3.1 Abbildung
logischer
Ausdrücke

3.2 Entscheidungs-
abhängige
Nebenbedingungen

Die Big-M-Methode

OPL: Modellieren von
Zeitperioden

Disjunktive
Nebenbedingungen

3.3 OPL:
Kompakte
Schreibweise

3.4 Stückweise
definierte
Funktionen

Treppenfunktionen

Stückweise lineare
Funktionen

OPL: Der piecewise-Befehl

CC-BY-SA
A. Popp

Sei \bar{x} der Vektor der Entscheidungsvariablen und f eine lineare Funktion. Die Nebenbedingung

$$f(\bar{\mathbf{x}}) \leq b \quad \text{bzw.} \quad f(\bar{\mathbf{x}}) \geq b$$

soll nur zwingend sein, wenn eine Entscheidung getroffen wurde, die dadurch repräsentiert wird, dass die Binärvariable y den Wert 0 annimmt.

Entscheidungsabhängige Nebenbedingungen

Sei M eine ausreichend große Zahl.

$$f(\bar{\mathbf{x}}) \leq b \quad \rightarrow \quad f(\bar{\mathbf{x}}) \leq b + M \cdot y$$

$$f(\bar{\mathbf{x}}) \geq b \quad \rightarrow \quad f(\bar{\mathbf{x}}) \geq b - M \cdot y$$

Die Big-M-Methode

OPL: Modellieren von Zeitperioden

Disjunktive
Nebenbedingungen

3.4 Stückweise definierte Funktionen

Treppenfunktionen

Stückweise lineare
Funktionen

OPL: Der piecewise-Befehl

Problem bei der Implementierung von Zeitperioden

3 Techniken der binären Modellierung

CC-BY-SA
A. Popp

Nebenbedingung aus Beispiel „Lewig Wakuxi“

$$i_t = i_{t-1} + x_t - d_t \quad \forall t \in T \quad (\text{I})$$

Implementierungsversuch 1

```
{string} T = {"KW14", "KW15", "KW16", "KW17"};
dvar float+ i[T];
forall(t in T) i[t] == i[t-1] + x[t] - d[t];
❌ Operator für string - int nicht verfügbar.
```

Die Big-M-Methode

OPL: Modellieren von Zeitperioden

Disjunktive
Nebenbedingungen

3.4 Stückweise definierte Funktionen

Treppenfunktionen

Stückweise lineare
Funktionen

OPL: Der piecewise-Befehl

Nebenbedingung aus Beispiel „Lewig Wakuxi“

$$i_t = i_{t-1} + x_t - d_t \quad \forall t \in T \quad (\text{I})$$

```
{int} T = {14, 15, 16, 17};
```

```
dvar float+ i[T];
```

```
forall(t in T) i[t] == i[t-1] + x[t] - d[t];
```

- Der Index für den Array "i" liegt außerhalb des gültigen Bereichs: 13.

Die Big-M-Methode

OPL: Modellieren von Zeitperioden

Disjunktive
Nebenbedingungen

3.4 Stückweise definierte Funktionen

Treppenfunktionen

Stückweise lineare
Funktionen

OPL: Der piecewise-Befehl

Problem bei der Implementierung von Zeitperioden

3 Techniken der binären Modellierung

CC-BY-SA
A. Popp

Die Big-M-Methode

OPL: Modellieren von Zeitperioden

Disjunktive
Nebenbedingungen

3.4 Stückweise definierte Funktionen

Treppenfunktionen

Stückweise lineare
Funktionen

OPL: Der piecewise-Befehl

Nebenbedingung aus Beispiel „Lewig Wakuxi“

$$i_t = i_{t-1} + x_t - d_t \quad \forall t \in T \quad (\text{I})$$

Implementierungsversuch 3

```
{int} T = {14, 15, 16, 17};
{int} T0 = {13, 14, 15, 16, 17};
dvar float+ i[T0];
  forall(t in T) i[t] == i[t-1] + x[t] - d[t];
```

Problem bei der Implementierung von Zeitperioden

3 Techniken der binären Modellierung

CC-BY-SA
A. Popp

Nebenbedingung aus Beispiel „Lewig Wakuxi“

$$i_t = i_{t-1} + x_t - d_t \quad \forall t \in T \quad (\text{I})$$

Implementierungsversuch 4

```
int Tmin = 14;
int Tmax = 17;
range T = Tmin..Tmax;
dvar float+ i[Tmin-1..Tmax];
forall(t in T) i[t] == i[t-1] + x[t] - d[t];
```

Die Big-M-Methode

OPL: Modellieren von Zeitperioden

Disjunktive Nebenbedingungen

3.4 Stückweise definierte Funktionen

Treppenfunktionen

Stückweise lineare
Funktionen

OPL: Der piecewise-Befehl

Ein Modell habe die beiden folgenden Nebenbedingungen:

$$f(\bar{\mathbf{x}}) \leq b$$

$$g(\bar{\mathbf{x}}) \leq d$$

Es reicht eine der beiden Bedingungen zu erfüllen.

Disjunktive Nebenbedingungen

Sei M eine ausreichend große Zahl und y eine binäre Hilfsvariable.

$$f(\bar{\mathbf{x}}) \leq b + M \cdot y$$

$$g(\bar{\mathbf{x}}) \leq d + M \cdot (1 - y)$$

\geq -Nebenbedingungen analog

Die Big-M-Methode

OPL: Modellieren von Zeitperioden

Disjunktive Nebenbedingungen

3.4 Stückweise definierte Funktionen

Treppenfunktionen

Stückweise lineare
Funktionen

OPL: Der piecewise-Befehl

Zielfunktion des Wagner-Whitin-Problems:

```
// Zielfunktion  
minimize sum(t in T)(s[t]*y[t] + h[t]* i[t]);
```

Zerlegen in Entscheidungsausdrücke:

```
// Entscheidungsausdrücke  
dexpr float setupCost = sum(t in T)(s[t]*y[t]);  
dexpr float inventoryCost = sum(t in T)(h[t]*i[t]);  
  
// Zielfunktion  
minimize setupCost + inventoryCost;
```

3.1 Abbildung
logischer
Ausdrücke

3.2 Entscheidungs-
abhängige
Nebenbedingungen

Die Big-M-Methode

OPL: Modellieren von
Zeitperioden

Disjunktive
Nebenbedingungen

3.3 OPL:
Kompakte
Schreibweise

3.4 Stückweise
definierte
Funktionen

Treppenfunktionen

Stückweise lineare
Funktionen

OPL: Der piecewise-Befehl

Arrays von Entscheidungsausdrücken

3 Techniken der
binären
Modellierung

CC-BY-SA
A. Popp

Zielfunktion des Wagner-Whitin-Problems:

```
// Zielfunktion
minimize sum(t in T)(s[t]*y[t] + h[t]* i[t]);
```

Zerlegen in Entscheidungsausdrücke:

```
//Entscheidungsausdrücke
dexpr float periodCost[t in T]
= s[t]*y[t] + h[t]*i[t];

// Zielfunktion
minimize sum (t in T)(periodCost[t]);
```

3.1 Abbildung
logischer
Ausdrücke

3.2 Entscheidungs-
abhängige
Nebenbedingungen

Die Big-M-Methode

OPL: Modellieren von
Zeitperioden

Disjunktive
Nebenbedingungen

3.3 OPL:
Kompakte
Schreibweise

3.4 Stückweise
definierte
Funktionen

Treppenfunktionen

Stückweise lineare
Funktionen

OPL: Der piecewise-Befehl

3.4 Stückweise definierte Funktionen

Die Big-M-Methode

OPL: Modellieren von Zeitperioden

Disjunktive
Nebenbedingungen

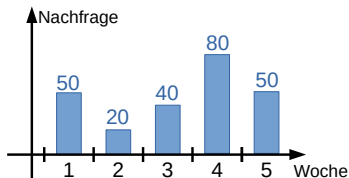
3.4 Stückweise definierte Funktionen

Treppenfunktionen

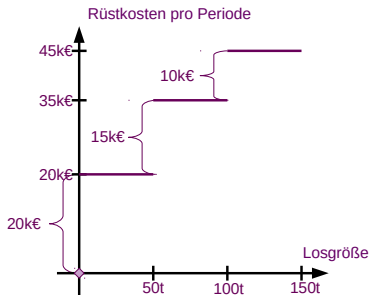
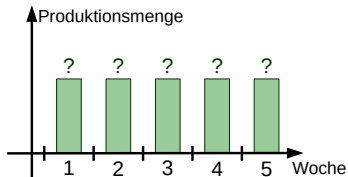
Stückweise lineare
Funktionen

OPL: Der piecewise-Befehl

Beispiel: Lewig Xanxi



Rüstkosten
Lagerkosten: 100€



3.1 Abbildung logischer Ausdrücke

3.2 Entscheidungs- abhängige Nebenbedingungen

Die Big-M-Methode
OPL: Modellieren von
Zeitperioden
Disjunktive
Nebenbedingungen

3.3 OPL: Kompakte Schreibweise

3.4 Stückweise definierte Funktionen

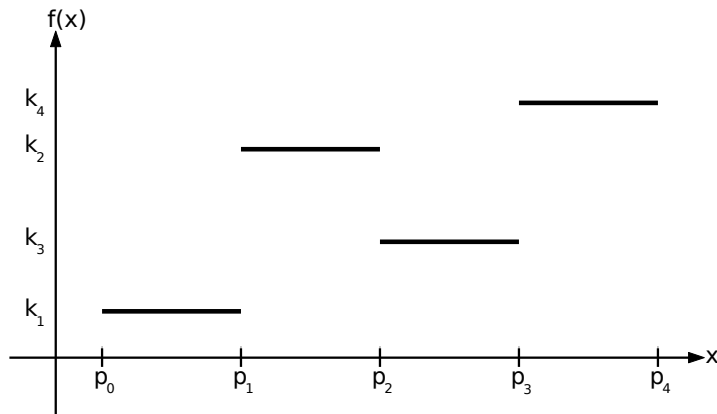
Treppenfunktionen

Stückweise lineare
Funktionen

OPL: Der piecewise-Befehl

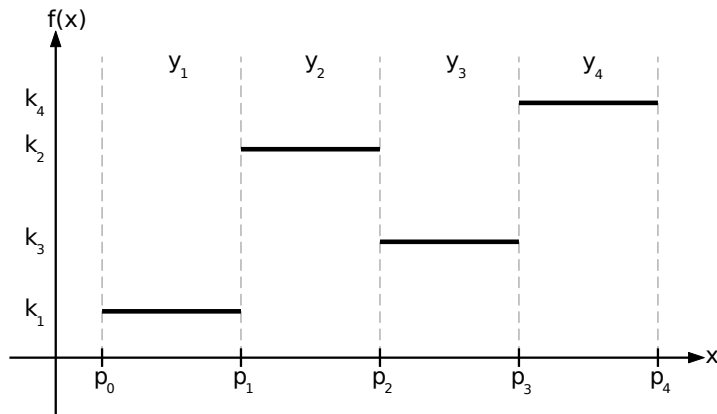
Treppenfunktionen

Sei x eine kontinuierliche Entscheidungsvariable:



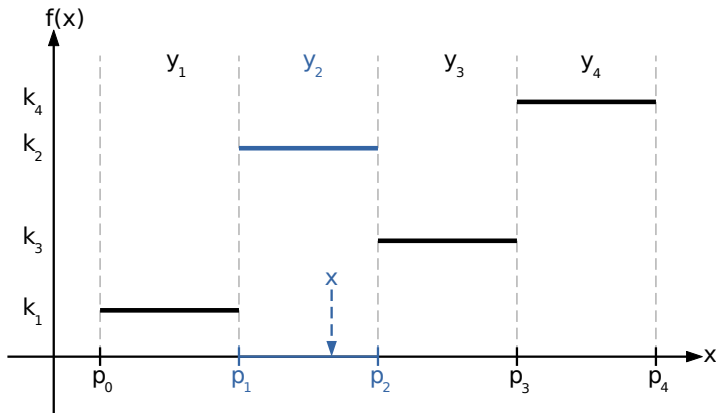
Treppenfunktionen

Sei x eine kontinuierliche Entscheidungsvariable:



Sei x eine kontinuierliche Entscheidungsvariable:

CC-BY-SA
A. Popp



► z.B.: $x = \frac{1}{3} \cdot p_1 + \frac{2}{3} \cdot p_2$

Die Big-M-Methode

OPL: Modellieren von Zeitperioden

Disjunktive Nebenbedingungen

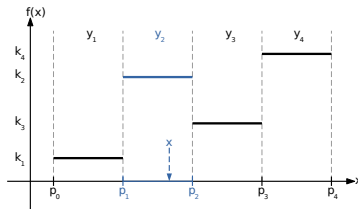
3.4 Stückweise definierte Funktionen

Treppenfunktionen

Stückweise lineare
Funktionen

OPL: Der piecewise-Befehl

Entscheidungsvariablen als Konvexkombination der Stützstellen



$$x = \sum_{n=0}^N z_n \cdot p_n$$

$$\sum_{n=0}^N z_n = 1$$

$$0 \leq z_n \leq 1 \quad \forall n \in \{1, \dots, N\}$$

3 Techniken der
binären
Modellierung

CC-BY-SA
A. Popp

3.1 Abbildung
logischer
Ausdrücke

3.2 Entscheidungs-
abhängige
Nebenbedingungen

Die Big-M-Methode

OPL: Modellieren von
Zeitperioden

Disjunktive
Nebenbedingungen

3.3 OPL:
Kompakte
Schreibweise

3.4 Stückweise
definierte
Funktionen

Treppenfunktionen

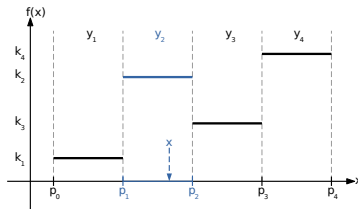
Stückweise lineare
Funktionen

OPL: Der piecewise-Befehl

Auswahl des korrekten Intervalls

3 Techniken der
binären
Modellierung

CC-BY-SA
A. Popp



$$\sum_{n=1}^N y_n = 1$$

$$z_0 \leq y_1$$

$$z_n \leq y_n + y_{n+1} \quad \forall n \in \{1, \dots, N-1\}$$

$$z_N \leq y_N$$

3.1 Abbildung
logischer
Ausdrücke

3.2 Entscheidungs-
abhängige
Nebenbedingungen

Die Big-M-Methode

OPL: Modellieren von
Zeitperioden

Disjunktive
Nebenbedingungen

3.3 OPL:
Kompakte
Schreibweise

3.4 Stückweise
definierte
Funktionen

Treppenfunktionen

Stückweise lineare
Funktionen

OPL: Der `piecewise`-Befehl

Vollständige Modellierung

$$f(x) = \sum_{n=1}^N y_n \cdot k_n$$

$$x = \sum_{n=0}^N z_n \cdot p_n$$

$$\sum_{n=0}^N z_n = 1$$

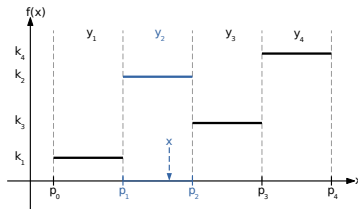
$$0 \leq z_n \leq 1 \quad \forall n \in \{1, \dots, N\}$$

$$\sum_{n=1}^N y_n = 1$$

$$z_0 \leq y_1$$

$$z_n \leq y_n + y_{n+1} \quad \forall n \in \{1, \dots, N-1\}$$

$$z_N \leq y_N$$



3 Techniken der
binären
Modellierung

CC-BY-SA
A. Popp

3.1 Abbildung
logischer
Ausdrücke

3.2 Entscheidungs-
abhängige
Nebenbedingungen

Die Big-M-Methode

OPL: Modellieren von
Zeitperioden

Disjunktive
Nebenbedingungen

3.3 OPL:
Kompakte
Schreibweise

3.4 Stückweise
definierte
Funktionen

Treppenfunktionen

Stückweise lineare
Funktionen

OPL: Der piecewise-Befehl

Ausgangssituation ist ein Produktionsproblem.

Kapazitäten

	Maschine A	Maschine B	Maschine C
I_1	2,2	1,6	2,8
I_2	1,2	1,9	2,3
	72	48	60

Verkaufserlöse

$$f_1(x_1) = 2000 \cdot \sqrt{x_1}$$

$$f_2(x_2) = 1800 \cdot \sqrt{x_2}$$

Die Big-M-Methode

OPL: Modellieren von Zeitperioden

Disjunktive
Nebenbedingungen

3.4 Stückweise definierte Funktionen

Treppenfunktionen

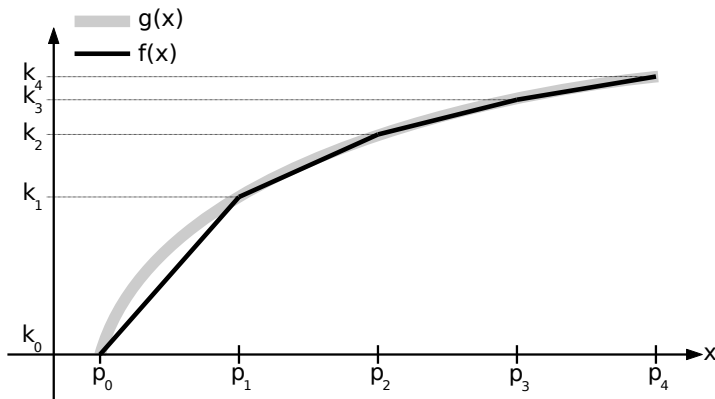
Stückweise lineare Funktionen

OPL: Der piecewise-Befehl

Stückweise lineare Funktionen

3 Techniken der
binären
Modellierung

CC-BY-SA
A. Popp



3.1 Abbildung
logischer
Ausdrücke

3.2 Entscheidungs-
abhängige
Nebenbedingungen

Die Big-M-Methode
OPL: Modellieren von
Zeitperioden
Disjunktive
Nebenbedingungen

3.3 OPL:
Kompakte
Schreibweise

3.4 Stückweise
definierte
Funktionen

Treppenfunktionen

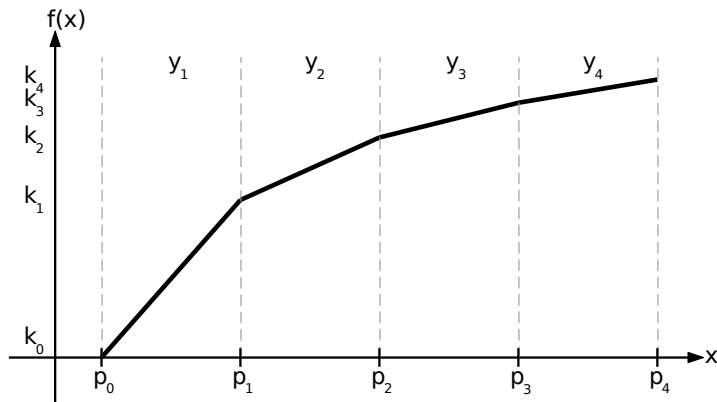
Stückweise lineare
Funktionen

OPL: Der `piecewise`-Befehl

Stückweise lineare Funktionen

3 Techniken der
binären
Modellierung

CC-BY-SA
A. Popp



3.1 Abbildung
logischer
Ausdrücke

3.2 Entscheidungs-
abhängige
Nebenbedingungen

Die Big-M-Methode
OPL: Modellieren von
Zeitperioden
Disjunktive
Nebenbedingungen

3.3 OPL:
Kompakte
Schreibweise

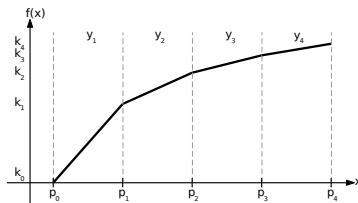
3.4 Stückweise
definierte
Funktionen

Treppenfunktionen
Stückweise lineare
Funktionen
OPL: Der piecewise-Befehl

Funktionswerte als Konvexkombination

3 Techniken der
binären
Modellierung

CC-BY-SA
A. Popp



$$x = \sum_{n=0}^N z_n \cdot p_n$$

$$f(x) = \sum_{n=0}^N z_n \cdot f(p_n)$$

$$\sum_{n=0}^N z_n = 1$$

$$0 \leq z_n \leq 1 \quad \forall n \in \{1, \dots, N\}$$

3.1 Abbildung
logischer
Ausdrücke

3.2 Entscheidungs-
abhängige
Nebenbedingungen

Die Big-M-Methode

OPL: Modellieren von
Zeitperioden

Disjunktive
Nebenbedingungen

3.3 OPL:
Kompakte
Schreibweise

3.4 Stückweise
definierte
Funktionen

Treppenfunktionen

Stückweise lineare
Funktionen

OPL: Der piecewise-Befehl

Vollständige Modellierung

$$f(x) = \sum_{n=0}^N z_n \cdot k_n$$

$$x = \sum_{n=0}^N z_n \cdot p_n$$

$$\sum_{n=0}^N z_n = 1$$

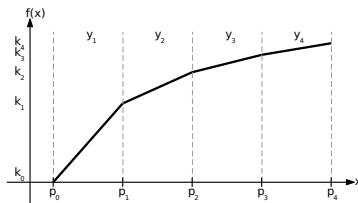
$$0 \leq z_n \leq 1 \quad \forall n \in \{1, \dots, N\}$$

$$\sum_{n=1}^N y_n = 1$$

$$z_0 \leq y_1$$

$$z_n \leq y_n + y_{n+1} \quad \forall n \in \{1, \dots, N-1\}$$

$$z_N \leq y_N$$



3 Techniken der
binären
Modellierung

CC-BY-SA
A. Popp

3.1 Abbildung
logischer
Ausdrücke

3.2 Entscheidungs-
abhängige
Nebenbedingungen

Die Big-M-Methode

OPL: Modellieren von
Zeitperioden

Disjunktive
Nebenbedingungen

3.3 OPL:
Kompakte
Schreibweise

3.4 Stückweise
definierte
Funktionen

Treppenfunktionen

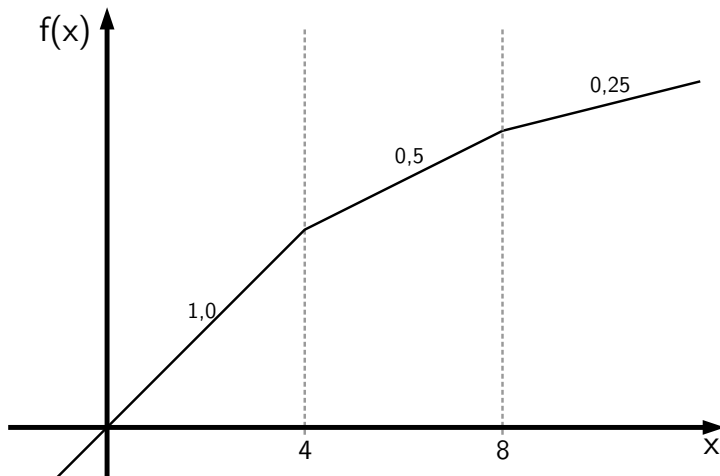
Stückweise lineare
Funktionen

OPL: Der piecewise-Befehl

Stückweise lineare Funktionen nach Steigung

3 Techniken der
binären
Modellierung

CC-BY-SA
A. Popp



3.1 Abbildung
logischer
Ausdrücke

3.2 Entscheidungs-
abhängige
Nebenbedingungen

Die Big-M-Methode

OPL: Modellieren von
Zeitperioden

Disjunktive
Nebenbedingungen

3.3 OPL:
Kompakte
Schreibweise

3.4 Stückweise
definierte
Funktionen

Treppenfunktionen

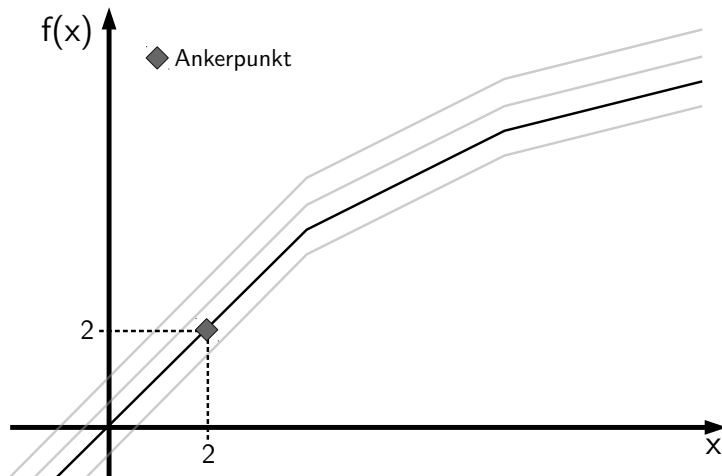
Stückweise lineare
Funktionen

OPL: Der `piecewise`-Befehl

Verankerung von stückweise linearen Funktionen

3 Techniken der
binären
Modellierung

CC-BY-SA
A. Popp



3.1 Abbildung
logischer
Ausdrücke

3.2 Entscheidungs-
abhängige
Nebenbedingungen

Die Big-M-Methode
OPL: Modellieren von
Zeiträumen
Disjunktive
Nebenbedingungen

3.3 OPL:
Kompakte
Schreibweise

3.4 Stückweise
definierte
Funktionen

Treppenfunktionen
Stückweise lineare
Funktionen

OPL: Der `piecewise`-Befehl

Array p der Stützstellen und Array s der Steigungen:

```

piecewise(i in 1..N){
  s[i] -> p[i];
  s[N+1]
} (Ankerpunkt) x;

```

Beispiel aus obiger Abbildung

```
int N = 2;
float p[1..N] = [4, 8];
float s[1..N+1] = [1.0, 0.5, 0.25];
dvar float+ x;
```

```
piecewise(i in 1..N){
  s[i] -> p[i];
  s[N+1]
} (2, 2) x;
```

Die Big-M-Methode

OPL: Modellieren von Zeitperioden

Disjunktive
Nebenbedingungen

3.4 Stückweise definierte Funktionen

Treppenfunktionen

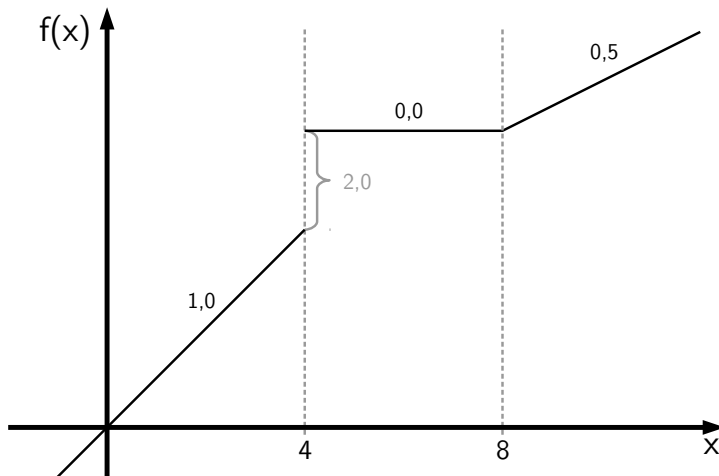
Stückweise lineare
Funktionen

OPL: Der piecewise-Befehl

Treppenfunktionen und allgemeine Unstetigkeiten

3 Techniken der
binären
Modellierung

CC-BY-SA
A. Popp



3.1 Abbildung
logischer
Ausdrücke

3.2 Entscheidungs-
abhängige
Nebenbedingungen

Die Big-M-Methode
OPL: Modellieren von
Zeitperioden
Disjunktive
Nebenbedingungen

3.3 OPL:
Kompakte
Schreibweise

3.4 Stückweise
definierte
Funktionen

Treppenfunktionen

Stückweise lineare
Funktionen

OPL: Der `piecewise`-Befehl

Zweiter Steigungswert an einem Punkt im `piecewise`-Befehl wird zu Sprungwert.

Beispiel aus obiger Abbildung

```
int N = 3;
float p[1..N] = [4, 4, 8];
float s[1..N+1] = [1.0, 2.0, 0.0, 0.5];
dvar float+ x;

piecewise(i in 1..N){
    s[i] -> p[i];
    s[N+1]
} x;
```

3.1 Abbildung
logischer
Ausdrücke

3.2 Entscheidungs-
abhängige
Nebenbedingungen

Die Big-M-Methode

OPL: Modellieren von
Zeitperioden

Disjunktive
Nebenbedingungen

3.3 OPL:
Kompakte
Schreibweise

3.4 Stückweise
definierte
Funktionen

Treppenfunktionen

Stückweise lineare
Funktionen

OPL: Der `piecewise`-Befehl