# Modellierung und Optimierung mit OPL

3 Techniken der binären Modellierung

#### Andreas Popp



Dieser Foliensatz ist lizenziert unter einer Creative Commons Namensnennung - Weitergabe unter gleichen Bedingungen 4.0 International Lizenz. 3 Techniken der binären Modellierung

> CC-BY-SA A. Popp

- 3.1 Abbildung logischer Ausdrücke
  - 2 Entscheidungsbhängige ebenbedingungen
- OPL: Modellieren von Zeitperioden Disjunktive
- Disjunktive Nebenbedingunge
- Kompakte Schreibweise
- 3.4 Stückweis definierte Funktionen
- Freppenfunktionen Stückweise lineare Funktionen
- OPL: Der piecewise-Befehl

### Inhalt

## 3.1 Abbildung logischer Ausdrücke

## 3.2 Entscheidungsabhängige 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 Techniken der binären Modellierung

> CC-BY-SA A. Popp

3.1 Abbildung ogischer Ausdrücke

.2 Entscheidungsbhängige lebenbedingungen

OPL: Modellieren voi Zeitperioden Disiunktive

lebenbedingungen

Kompakte Schreibweise

3.4 Stuckweis definierte Funktionen

Treppenfunktionen Stückweise lineare Funktionen

#### abhängige Nebenbedingungen Die Big-M-Methode

OPL: Modellieren vo Zeitperioden Disjunktive

Nebenbedingung

#### 3.3 OPL: Kompakte Schreibweise

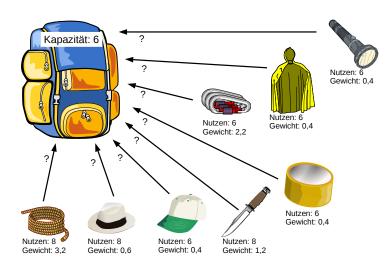
#### 3.4 Stückweise definierte Funktionen

Treppenfunktionen Stückweise lineare

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 Entscheidungsbhängige Vebenbedingungen
  - OPL: Modellieren voi Zeitperioden Disjunktive
- 3.3 OPL: Kompakte Schreibweise
- 3.4 Stückwei definierte Funktionen
- Treppenfunktionen Stückweise lineare Funktionen
- OPL: Der piecewise-Befeh

## Modell: Rucksackproblem

#### 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:

$$\max \sum_{i \in I} u_i \cdot x_i$$

$$s.t. \sum_{i \in I} w_i \cdot x_i \le c$$

$$x_i \in \{0,1\} \qquad \forall i \in I$$

$$(I)$$

3 Techniken der binären Modellierung

> CC-BY-SA A. Popp

3.1 Abbildung logischer Ausdrücke

3.2 Entscheidungsabhängige

OPL: Modellieren von Zeitperioden

Disjunktive Nebenbedingunger

3.3 OPL: Kompakte

3.4 Stückweise definierte

Treppenfunktionen Stückweise lineare Funktionen

# Logische Verknüpfungen

- 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

| Α | В | $\neg A$ | $\neg B$ | $A \wedge B$ | $A \lor B$ | $A \vee B$ | $A \Rightarrow B$ | $A \Leftrightarrow B$ |
|---|---|----------|----------|--------------|------------|------------|-------------------|-----------------------|
| 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 Techniken der binären Modellierung

> CC-BY-SA A. Popp

3.1 Abbildung logischer Ausdrücke

# 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.

 $\neg l_1$ : Man möchte wissen ob  $l_1$  nicht eingepackt ist.

▶ 
$$1 - x_1$$

 $\emph{I}_1 \wedge \emph{I}_2$ : Sowohl  $\emph{I}_1$  als auch  $\emph{I}_2$  müssen eingepackt werden.

$$x_1 + x_2 = 2$$

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

► 
$$x_1 + x_2 \ge 1$$

 $\neg (\mathit{I}_1 \wedge \mathit{I}_2)$ : Höchstens einer der Gegenstände darf eingepackt werden

$$x_1 + x_2 < 1$$

3 Techniken der binären Modellierung

> CC-BY-SA A. Popp

3.1 Abbildung logischer Ausdrücke

> 2 Entscheidungshängige ebenbedingungen

DPL: Modellieren vo leitperioden Disjunktive

3.3 OPL: Kompakte

3.4 Stückweise definierte

Treppenfunktionen Stückweise lineare Funktionen

# 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.

 $\neg(I_1 \lor I_2)$ : Keiner der Gegenstände darf eingepackt werden.

$$x_1 + x_2 = 0$$

 $l_1 \vee l_2$ : Genau einer der Gegenstände muss eingepackt werden.

$$x_1 + x_2 = 1$$

 $I_1 \Rightarrow I_2$ : Wenn  $I_1$  eingepackt ist, muss auch  $I_2$  eingepackt werden.

▶ 
$$x_1 \le x_2$$

 $l_1 \Leftrightarrow l_2$ : Die Entscheidung für beide Gegenstände ist identisch.

3 Techniken der binären Modellierung

> CC-BY-SA A. Popp

3.1 Abbildung logischer Ausdrücke

> 2 Entscheidungshängige ebenbedingungen

le Big-W-Methode
PL: Modellieren vor
eitperioden
isjunktive
ebenbedingungen

3.3 OPL: Kompakte Schreibweise

3.4 Stückweise definierte Funktionen

Treppenfunktionen Stückweise lineare Funktionen

# 3.2 Entscheidungsabhängige Nebenbedingungen

3 Techniken der binären Modellierung

> CC-BY-SA A. Popp

3.1 Abbildung logischer Ausdrücke

#### 3.2 Entscheidungsabhängige Nebenbedingungen

OPL: Modellieren von Zeitperioden Disjunktive

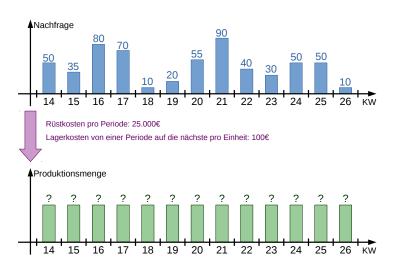
#### 3.3 OPL: Kompakte Schreibweise

3.4 Stückweise definierte

Treppenfunktionen

OPI : Dor minescripe Refebi

# Beispiel: Lewig Wakuxi



3 Techniken der binären Modellierung

> CC-BY-SA A. Popp

#### 3.2 Entscheidungsabhängige Nebenbedingungen

# Modell: Wagner-Whitin-Problem

#### Indexmengen:

T Menge der Planungsperioden  $\{t_{min}, \dots, t_{max}\}$ 

#### Parameter:

 $d_t$  Nachfrage in Periode  $t \in T$ 

 $s_t$  Rüstkostensatz in Periode  $t \in T$ 

 $h_t$  Lagerkostensatz in Periode  $t \in T$ 

 $i_{t_{min}-1}$  Anfangsbestand

M Eine große Zahl

#### Entscheidungsvariablen:

 $x_t$  Produktionsmenge in Periode  $t \in T$ 

 $i_t$  Bestand am Ende von Periode  $t \in T$ 

 $y_t$  Produktionsentscheidung in Periode  $t \in T$ 

#### Modellbeschreibung:

$$\min \quad \sum_{t \in T} s_t \cdot y_t + h_t \cdot i_t$$

s.t. 
$$i_t = i_{t-1} + x_t - d_t$$
  $\forall t \in T$  (I)  
 $x_t \leq M \cdot y_t$   $\forall t \in T$  (II)

$$x_t, i_t \geq 0; \ y_t \in \{0,1\} \qquad \forall t \in T$$

3 Techniken der binären Modellierung

> CC-BY-SA A. Popp

.1 Abbildung ogischer Ausdrücke

3.2 Entscheidungsabhängige Nebenbedingungen

OPL: Modellieren v Zeitperioden Disjunktive

3.3 OPL:

Kompakte Schreibweise

> 6.4 Stückwei lefinierte Funktionen

Treppenfunktionen Stückweise lineare Funktionen

# Die Big-M-Methode

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

$$f(\overline{\mathbf{x}}) \le b$$
 bzw.  $f(\overline{\mathbf{x}}) \ge b$ 

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.

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

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

3 Techniken der binären Modellierung

> CC-BY-SA A. Popp

- 3.1 Abbildung logischer Ausdrücke
- 3.2 Entscheidungsabhängige Nebenbedingungen

#### Die Big-M-Methode

OPL: Modellieren vor Zeitperioden Disjunktive Nebenbedingungen

#### 3.3 OPL: Kompakte

Kompakte Schreibweise

.4 Stückweis efinierte unktionen

reppenfunktionen stückweise lineare unktionen

# Problem bei der Implementierung von Zeitperioden

# Nebenbedingung aus Beispiel "Lewig Wakuxi"

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

3 Techniken der binären Modellierung

> CC-BY-SA A. Popp

3.1 Abbildung ogischer Ausdrücke

.2 Entscheidungsbhängige lebenbedingungen

OPL: Modellieren von

Zeitperioden Disjunktive

ebenbedingungen

3.3 OPL: Kompakte

Schreibweise

I.4 Stückweis lefinierte <sup>F</sup>unktionen

Treppenfunktionen Stückweise lineare Funktionen

# Problem bei der Implementierung von Zeitperioden

## Nebenbedingung aus Beispiel "Lewig Wakuxi"

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

#### Implementierungsversuch 2

```
{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.

3 Techniken der binären Modellierung

> CC-BY-SA A. Popp

OPI · Modellieren von Zeitnerioden

# Problem bei der Implementierung von Zeitperioden

# Nebenbedingung aus Beispiel "Lewig Wakuxi"

$$i_t = i_{t-1} + x_t - d_t \quad \forall t \in T$$
 (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];
```

3 Techniken der binären Modellierung

> CC-BY-SA A. Popp

- 3.1 Abbildung ogischer Ausdrücke
- .2 Entscheidungsbhängige Iebenbedingungen
- OPL: Modellieren von Zeitperioden

Disjunktive

ebenbedingungen

- 3.3 OPL: Kompakte Schreibweise
- Schreibweise
- 3.4 Stuckweise definierte Funktionen
- Treppenfunktionen Stückweise lineare Funktionen
- OPL: Der piecewise-Befehl

# Problem bei der Implementierung von Zeitperioden

## Nebenbedingung aus Beispiel "Lewig Wakuxi"

$$i_t = i_{t-1} + x_t - d_t \quad \forall t \in T$$
 (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];
```

3 Techniken der binären Modellierung

> CC-BY-SA A. Popp

- 3.1 Abbildung ogischer Ausdrücke
- .2 Entscheidungsbhängige Iebenbedingungen
- OPL: Modellieren von

Zeitperioden Disjunktive

Vebenbedingungen

3.3 OPL: Kompakte

- Schreibweise 3.4 Stückweis
- definierte Funktionen
- Treppenfunktionen Stückweise lineare Funktionen
- OPL: Der piecewise-Befehl

# Disjunktive Nebenbedingungen I

Ein Modell habe die beiden folgenden Nebenbedingungen:

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

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

Es reicht eine der beiden Bedingungen zu erfüllen.

## Disjunktive Nebenbedingungen

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

$$f(\overline{\mathbf{x}}) \le b + M \cdot y$$
  
 $g(\overline{\mathbf{x}}) \le d + M \cdot (1 - y)$ 

≥-Nebenbedingungen analog

3 Techniken der binären Modellierung

> CC-BY-SA A. Popp

8.1 Abbildung ogischer Ausdrücke

2 Entscheidungshängige ebenbedingungen

OPL: Modellieren vo Zeitperioden

Disjunktive Nebenbedingungen

3.3 OPL: Kompakte

Kompakte Schreibweise

> definierte Funktionen

Treppenfunktioner Stückweise lineare

Stückweise linear Funktionen

# Disjunktive Nebenbedingungen II

Ein Modell habe die folgende Nebenbedingungen:

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

Diese muss nur erfüllt werden, wenn gilt:

$$f(\overline{\mathbf{x}}) > b$$

### Disjunktive Nebenbedingungen

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

$$f(\overline{\mathbf{x}}) \le b + M \cdot y$$
  
 $g(\overline{\mathbf{x}}) \le d + M \cdot (1 - y)$ 

≥-Nebenbedingungen analog

3 Techniken der binären Modellierung

> CC-BY-SA A. Popp

3.1 Abbildung ogischer
Ausdrücke

2 Entscheidungshängige ebenbedingungen

OPL: Modellieren vo Zeitperioden

Disjunktive Nebenbedingungen

3.3 OPL: Kompakte

3.4 Stückweise definierte

Treppenfunktionen Stückweise lineare

# 3.3 OPL: Kompakte Schreibweise

3 Techniken der binären Modellierung

> CC-BY-SA A. Popp

3.1 Abbildung logischer Ausdrücke

3.2 Entscheidungsabhängige Nebenbedingungen

OPL: Modellieren vi Zeitperioden

3.3 OPL: Kompakte Schreibweise

3.4 Stückweise definierte Funktionen

Treppenfunktionen

# Entscheidungsausdrücke

#### 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 Techniken der binären Modellierung

> CC-BY-SA A. Popp

3.1 Abbildung ogischer Ausdrücke

2 Entscheidungsbhängige ebenbedingungen

OPL: Modellieren von Zeitperioden

Disjunktive Nebenbedingunge

#### 3.3 OPL: Kompakte Schreibweise

3.4 Stückweise definierte

Treppenfunktionen

tückweise lineare unktionen

# Arrays von Entscheidungsausdrücken

#### 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 Techniken der binären Modellierung

#### CC-BY-SA A. Popp

- 3.1 Abbildung logischer Ausdrücke
- .2 Entscheidungsbhängige lebenbedingungen
- Die Big-M-Methode

  OPL: Modellieren vor
  Zeitperioden

  Disjunktive
- Nebenbedingunger
- 3.3 OPL: Kompakte Schreibweise
  - 3.4 Stückweise definierte Funktionen
  - Treppenfunktionen Stückweise lineare
  - OPL: Der piecewise-Befehl

# 3.4 Stückweise definierte Funktionen

3 Techniken der binären Modellierung

> CC-BY-SA A. Popp

3.1 Abbildung logischer Ausdrücke

.2 Entscheidungsbhängige Iebenbedingungen

OPL: Modellieren v Zeitperioden

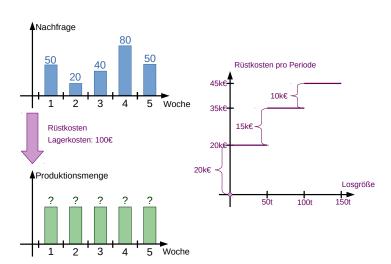
Disjunktive Nebenbedingung

3.3 OPL: Kompakte Schreibweise

#### 3.4 Stückweise definierte Funktionen

Treppenfunktioner Stückweise lineare Funktionen

# Beispiel: Lewig Xanxi



3 Techniken der binären Modellierung

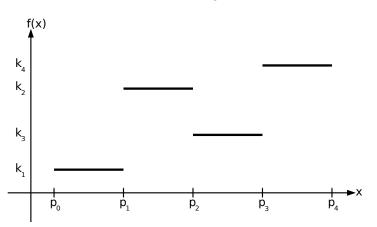
> CC-BY-SA A. Popp

- 3.1 Abbildung logischer Ausdrücke
- 3.2 Entscheidungsibhängige Nebenbedingungen
- OPL: Modellieren v Zeitperioden
- Disjunktive Vebenbedingungen
- 3.3 OPL: Kompakte Schreibweise
  - 3.4 Stückweise Iefinierte Funktionen

Treppenfunktionen Stückweise lineare Eunktionen

## Treppenfunktionen

Sei x eine kontinuierliche Entscheidungsvariable:



3 Techniken der binären Modellierung

> CC-BY-SA A. Popp

3.1 Abbildung logischer Ausdrücke

> 3.2 Entscheidungsibhängige Nebenbedingungen

OPL: Modellieren vo Zeitperioden Disjunktive

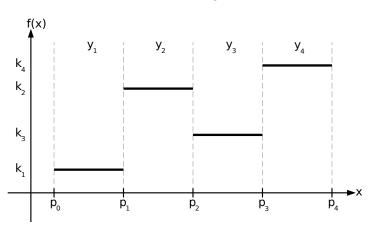
3.3 OPL: Kompakte

3.4 Stückweise definierte Eunktionen

Treppenfunktionen Stückweise lineare Funktionen

## Treppenfunktionen

Sei x eine kontinuierliche Entscheidungsvariable:



3 Techniken der binären Modellierung

> CC-BY-SA A. Popp

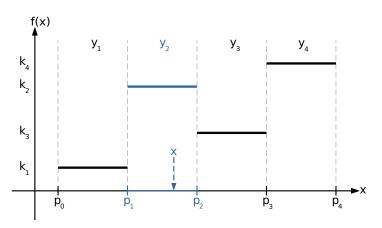
- 3.1 Abbildung logischer Ausdrücke
  - .2 Entscheidungsbhängige lebenbedingungen
- OPL: Modellieren w Zeitperioden
- Disjunktive Nebenbedingungen
- Kompakte Schreibweise
- 3.4 Stückweise definierte Funktionen

Treppenfunktionen Stückweise lineare Funktionen

OPI : Der niecewise-Befeh

## Treppenfunktionen

Sei x eine kontinuierliche Entscheidungsvariable:



3 Techniken der binären Modellierung

> CC-BY-SA A. Popp

3.1 Abbildung logischer Ausdrücke

2 Entscheidungsbhängige ebenbedingungen

OPL: Modellieren vo Zeitperioden Disjunktive

Nebenbedingungen

3.3 OPL:

Kompakte Schreibweise

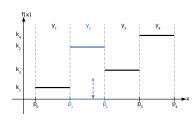
definierte Funktionen

Treppenfunktionen Stückweise lineare Funktionen

OPL: Der piecewise-Befeh

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

# Entscheidungsvariablen als Konvexkombination der Stützstellen



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

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

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

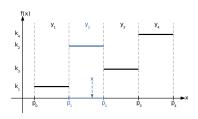
3 Techniken der binären Modellierung

> CC-BY-SA A. Popp

- 3.1 Abbildung logischer Ausdrücke
- 3.2 Entscheidungsabhängige Nebenbedingungen
- OPL: Modellieren vo Zeitperioden
- Disjunktive Nebenbedingungen
- Kompakte Schreibweise
- 3.4 Stückweise definierte Funktionen

Treppenfunktionen Stückweise lineare Funktionen

### Auswahl des korrekten Intervalls



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

$$z_0 \le y_1$$

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

$$z_N \le y_N$$

3 Techniken der binären Modellierung

> CC-BY-SA A. Popp

3.1 Abbildung logischer Ausdrücke

3.2 Entscheidungsabhängige Nebenbedingungen

OPL: Modellieren vo Zeitperioden Disjunktive

Nebenbedingungen

Kompakte Schreibweise

definierte
Funktionen

Treppenfunktionen Stückweise lineare Funktionen

# Vollständige Modellierung

 $z_0 \leq y_1$ 

 $z_N < y_N$ 

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

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

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

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

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

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

3 Techniken der binären Modellierung

CC-BY-SA A. Popp

3.1 Abbildung logischer Ausdrücke

3.2 Entscheidungsabhängige Nebenbedingungen

DPL: Modellieren von Zeitperioden Disjunktive Nebenbedingungen

Kompakte Schreibweise

definierte Funktionen

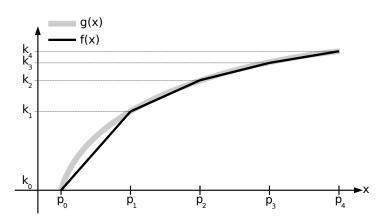
Treppenfunktionen Stückweise lineare Funktionen

OPL: Der piecewise-Befeh

У,

 $y_4$ 

### Stückweise lineare Funktionen



3 Techniken der binären Modellierung

> CC-BY-SA A. Popp

3.1 Abbildung logischer Ausdrücke

.2 Entscheidungsbhängige lebenbedingungen

OPL: Modellieren vo Zeitperioden

Disjunktive Nebenbedingunger

3.3 OPL: Kompakte

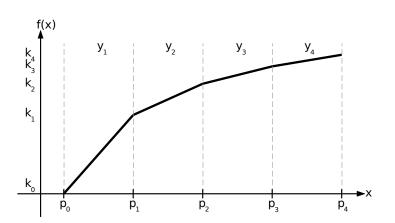
3.4 Stückweis definierte

Funktionen
Treppenfunktionen

Stückweise lineare

Funktionen

## Stückweise lineare Funktionen



3 Techniken der binären Modellierung

> CC-BY-SA A. Popp

3.1 Abbildung logischer Ausdrücke

2 Entscheidungsbhängige ebenbedingungen

OPL: Modellieren vo Zeitperioden

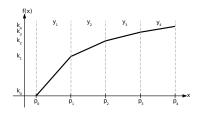
3.3 OPL: Kompakte

Schreibweise

3.4 Stuckweise definierte Funktionen

Stückweise lineare

## Funktionswerte als Konvexkombination



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

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

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

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

3 Techniken der binären Modellierung

> CC-BY-SA A. Popp

3.1 Abbildung logischer Ausdrücke

> 3.2 Entscheidungs-Ibhängige Vebenbedingungen

PL: Modellieren von eitperioden Pisjunktive lebenbedingungen

8.3 OPL: Kompakte Schreibweise

3.4 Stückweise definierte Funktionen

Stückweise lineare

# Vollständige Modellierung

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

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

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

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

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

$$z_0 \le y_1$$

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

 $z_N < y_N$ 

3 Techniken der binären Modellierung

CC-BY-SA A. Popp

3.1 Abbildung logischer Ausdrücke

3.2 Entscheidungsabhängige Nebenbedingungen

Die Big-M-Methode

DPL: Modellieren von
Zeitperioden

Disjunktive

Kompakte Schreibweise

definierte Funktionen

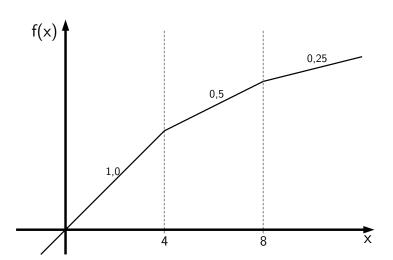
Stückweise lineare Funktionen

OPL: Der piecewise-Befeh

У,

У<sub>3</sub>

# Stückweise lineare Funktionen nach Steigung



3 Techniken der binären Modellierung

> CC-BY-SA A. Popp

3.1 Abbildung logischer Ausdrücke

5.2 EntscheidungsbhängigeVebenbedingungen

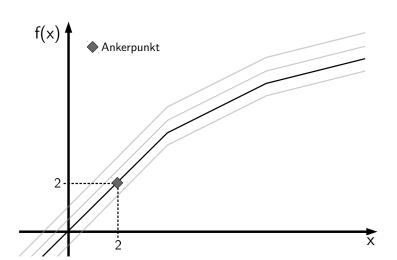
DPL: Modellieren von Leitperioden Disjunktive Jebenbedingungen

Kompakte Schreibweise

3.4 Stückweise definierte Funktionen

Treppenfunktionen Stückweise lineare Funktionen

# Verankerung von stückweise linearen Funktionen



3 Techniken der binären Modellierung

> CC-BY-SA A. Popp

3.1 Abbildung logischer Ausdrücke

3.2 Entscheidungsibhängige Nebenbedingungen

DPL: Modellieren vo eitperioden

Disjunktive Nebenbedingunger

Kompakte Schreibweise

3.4 Stückweise definierte Funktionen

Treppenfunktionen Stückweise lineare Funktionen

# Syntax des piecewise-Befehls

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:
```

3 Techniken der binären Modellierung

> CC-BY-SA A. Popp

3.1 Abbildung ogischer Ausdrücke

.2 Entscheidungsbhängige lebenbedingungen

OPL: Modellieren vo Zeitperioden

Disjunktive Nebenbedingungen

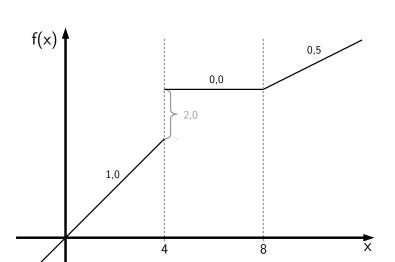
Kompakte

Schreibweise

s.4 Stuckweise lefinierte <sup>-</sup>unktionen

Treppenfunktionen Stückweise lineare Funktionen

# Treppenfunktionen und allgemeine Unstetigkeiten



3 Techniken der binären Modellierung

> CC-BY-SA A. Popp

3.1 Abbildung logischer Ausdrücke

> 2 Entscheidungsbhängige ebenbedingungen

OPL: Modellieren v Zeitperioden Disiunktive

Disjunktive Nebenbedingunger

Kompakte Schreibweise

3.4 Stückweise definierte Funktionen

Treppenfunktionen Stückweise lineare

# Treppenfunktionen und allgemeine Unstetigkeiten

3 Techniken der binären Modellierung

> CC-BY-SA A. Popp

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] \rightarrow p[i];
  s[N+1]
} x:
```