



Technische
Universität
Braunschweig

**Decision
Support**

Institut für Wirtschaftsinformatik



Operations Research

Vorlesung 6

Lineare Programmierung: Ganzzahlige Programmierung & Branch and Bound

Wiederholung

- Die drei Schritte im Operations Research
 - Problem, Modell, Lösung
- Typische Problemszenarien
 - Z.B. Transportproblem, Energieflussproblem, Auswahlproblem
- Primaler und dualer Simplex-Algorithmus
- Sensitivitätsanalyse
- Dualität

Heutige Fragestellungen

- Wie kann man Probleme modellieren, in denen die Entscheidungen ganzzahlig sein müssen?
 - Z.B. Anzahl von Personen gesucht
 - Z.B. Auftrag annehmen oder ablehnen (ja oder nein)
- Wie kann man solche Probleme (effizient) lösen?
- Sind die Probleme komplexer als lineare Probleme?

Überblick

1. Lineare Optimierungsprobleme mit ganzzahligen Variablen
2. Das Branch and Bound Verfahren
3. Lösung des Rucksackproblems mit Branch and Bound
4. Komplexitätstheorie

Überblick

1. Lineare Optimierungsprobleme mit ganzzahligen Variablen
2. Das Branch and Bound Verfahren
3. Lösung des Rucksackproblems mit Branch and Bound
4. Komplexitätstheorie

Probleme mit ganzzahligen Variablen

- Die Nachfrage der Großhändler nach Traktoren wird aus unterschiedlichen Produktionsstandorten befriedigt. Die Anzahl der Traktoren $x_{ij} \in \mathbb{N}$ wird von Produktionsstandort i nach Großhändler j verbracht.
- In einer innerbetrieblichen Stellenbesetzung werden Bewerber offenen Stellen zugeordnet. Bewerber i wird auf Stelle j eingestellt: $x_{ij} \in \{0, 1\}$
- Die Menge x_i des eines Gutes wird an Standort y_i produziert. Die Menge $x_i \in \mathbb{R}^+$ kann nur in Standort i produziert werden, wenn dort eine Produktionsfreigabe $y_i \in \{0,1\}$ besteht.

Ganzzahlige Optimierung

Definition:

$$\begin{array}{ll} \max & \sum_{j=1}^n c_j x_j = c^T x \\ \text{u.d.N.} & Ax \leq b \\ & x \in \mathbb{N}_0^n \end{array} \quad \left. \vphantom{\begin{array}{l} \max \\ \text{u.d.N.} \end{array}} \right\} \text{ ganzzahliges lineares Programm}$$

$$\begin{array}{ll} \text{gilt:} & x_1, \dots, x_n \in \{0, 1\} \end{array} \quad \left. \vphantom{\begin{array}{l} \text{gilt:} \end{array}} \right\} \text{ binäres lineares Programm}$$

$$\begin{array}{ll} \text{gilt:} & x_1, \dots, x_k \in \mathbb{N}_0 \\ & x_{k+1}, \dots, x_n \in \mathbb{R}^* \end{array} \quad \left. \vphantom{\begin{array}{l} \text{gilt:} \end{array}} \right\} \begin{array}{l} \text{gemischt ganzzahliges lineares Programm} \\ \text{Allg. MIP = mixed integer programming} \end{array}$$

Überblick

1. Lineare Optimierungsprobleme mit ganzzahligen Variablen
- 2. Das Branch and Bound Verfahren**
3. Anwendungsbeispiele zum Branch and Bound Verfahren
4. Komplexitätstheorie

Branch and Bound

Branch and Bound beschreibt ein Rahmenwerk, in dem drei algorithmische Komponenten beschrieben werden. Diese werden im folgenden spezifiziert:

a) Relaxation: Ableitung eines einfacher zu lösenden Problems

- I.d.R. relaxieren der Ganzzahligkeitsbedingung der Variablen \rightarrow LP

b) Separation: Generierung von sukzessiv zu lösenden Problemen

- Selektives Einfügen von Ganzzahligkeitsbedingungen für Variablen

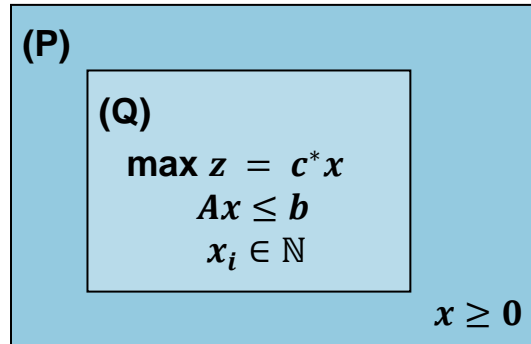
c) Auslotung: Reihenfolge der Abarbeitung von Folgeproblemen

- Entscheidung über Abbruch und ggf. Fortführung des Verfahrens

Branch and Bound: Relaxation

(a) Relaxation

Ersetzung des ganzzahligen Problems (Q) durch ein nicht-ganzzahliges Problem (P)



Besitzt (P) keine Lösung, so gibt es auch keine Lösung für (Q). Das Problem kann nicht gelöst werden.

Besitzt (P) eine ganzzahlige optimale Lösung x^* , so ist dies die optimale Lösung des Problems (Q). Der ZFW z bildet ggf. eine untere Schranke für (Q).

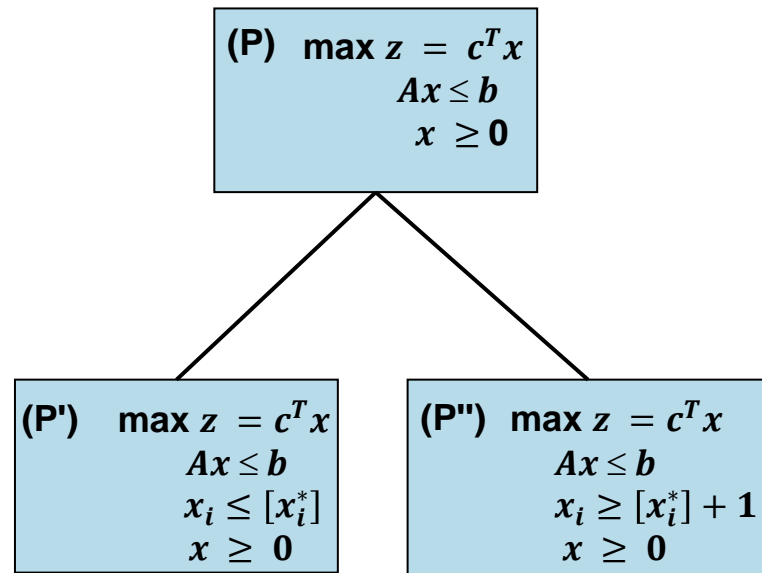
Besitzt (P) eine nicht-ganzzahlige optimale Lösung x^* , so ist dessen ZFW z eine obere Schranke für (Q). Weiter mit Separation → nächste Folie.

Branch and Bound: Separation

(b) Separation

Aufspaltung des Problems (P) in zwei Teilprobleme für eine nicht-ganzzahlige Variable x_i^* in der Lösung x^* von (P).

- Teilprobleme (P') und (P'') werden durch Hinzufügen von Restriktionen gebildet.
- Generell kann jede zusätzliche Restriktion den ZFW nur verschlechtern.
- Untere Schranke werden durch beste bekannte ganzzahlige Lösung bestimmt.
- Obere Schranken bestimmen sich aus ZFW von nicht-ganzzahligen Lösungen.
- Separation nur wenn obere Schranke eines Folgeproblems > untere Schranke.



$a \in \mathbb{R} \Rightarrow \lfloor a \rfloor$: größte ganze Zahl kleiner oder gleich a

Branch and Bound: Auslotung

(c) Auslotung

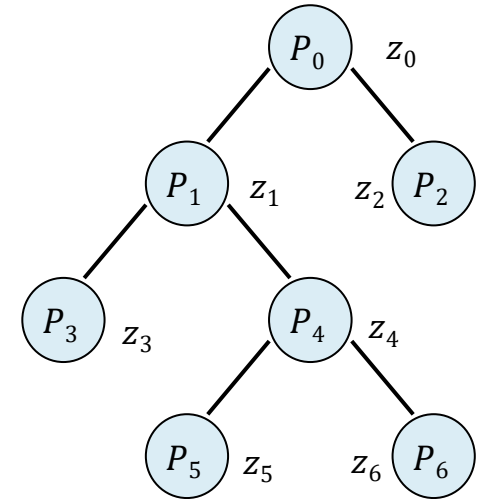
Es liege der folgende Entscheidungsbaum mit den oberen Schranken z_0, \dots, z_6 vor. Dabei sei $z_i = -\infty$, falls (P_i) keine Lösung besitzt.

Entscheidung,

- ob überhaupt eine weitere Separation notwendig ist, und
- ggf. welches Teilproblem (P_2) , (P_3) , (P_5) oder (P_6) im nächsten Schritt bearbeitet werden soll

Spezifiziere (P_{i0}) mit $z_{i0} = \max\{z_2, z_3, z_5, z_6\}$ und verfare wie folgt:

- Ist $z_{i0} = -\infty$, so besitzt (Q) keine zulässige Lösung.
- Ist $z_{i0} > \infty$ und besitzt (P_{i0}) eine optimale Lösung, die ganzzahlig ist, so ist diese Lösung optimal für (Q) .
- Ist $z_{i0} > -\infty$ und besitzt (P_{i0}) eine optimale Lösung, die nicht ganzzahlig ist, so nehme man eine weitere Separation mit (P_{i0}) vor.



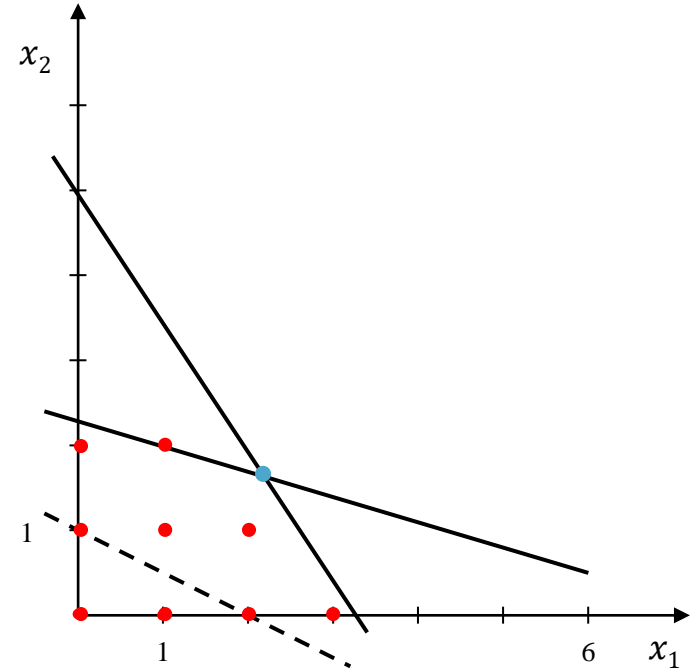
Beispiel: Lineares ganzzahliges Maximierungsproblem

(Q): $\max z = x_1 + 2x_2$
u.d.N. $x_1 + 3x_2 \leq 7$
 $3x_1 + 2x_2 \leq 10$
 $x_1, x_2 \in \mathbb{N}_0$

(P₀): $\max z = x_1 + 2x_2$
u.d.N. $x_1 + 3x_2 \leq 7$
 $3x_1 + 2x_2 \leq 10$
 $x_1, x_2 \geq 0$

LP-Relaxation

Lösung: $x_1 = 16/7 \approx 2,29$, $x_2 = 11/7 \approx 1,57$, $z_0 = 38/7 \approx 5,43$



Beispiel: Lineares ganzzahliges Maximierungsproblem

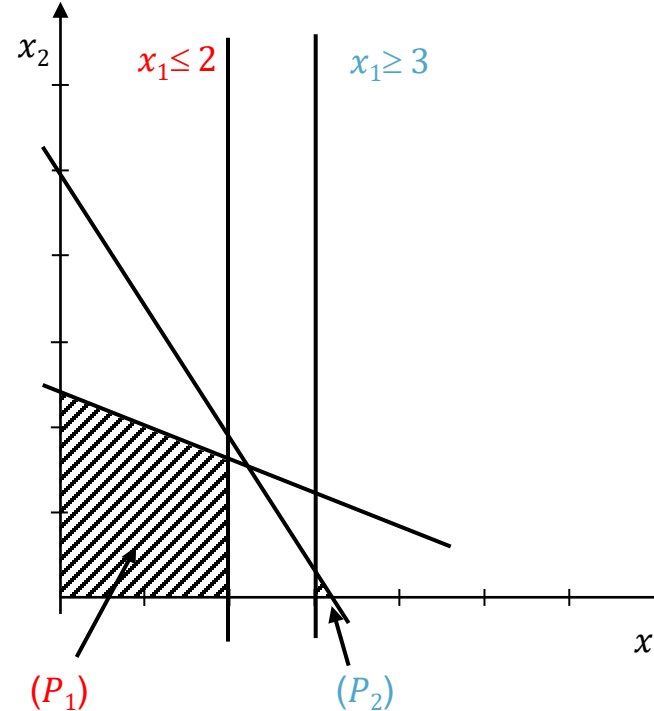
$$P_0$$
$$x = (2,29; 1,57) \quad z^{(0)} = 5,43$$

Beispiel: Lineares ganzzahliges Maximierungsproblem

Da x_1 nicht ganzzahlig ist, wird eine Separation von (P_0) bzgl. x_1 vorgenommen.

$$P_1: x_1^{(1)} \leq [x_1] = 2$$

$$P_2: x_1^{(2)} \geq [x_1] + 1 = 3$$



Beispiel: Lineares ganzzahliges Maximierungsproblem

(P₁): $x_1 \leq 2$

$$(P_1): \max z = x_1 + 2x_2$$

$$\text{u.d.N.} \quad x_1 + 3x_2 \leq 7$$

$$3x_1 + 2x_2 \leq 10$$

$$x_1 \leq 2$$

$$x_1, x_2 \geq 0$$

$$\text{Lösung: } x_1 = 2, x_2 = 5/3, z^{(1)} = 16/3$$

(P₂): $x_1 \geq 3$

$$(P_2): \max z = x_1 + 2x_2$$

$$\text{u.d.N.} \quad x_1 + 3x_2 \leq 7$$

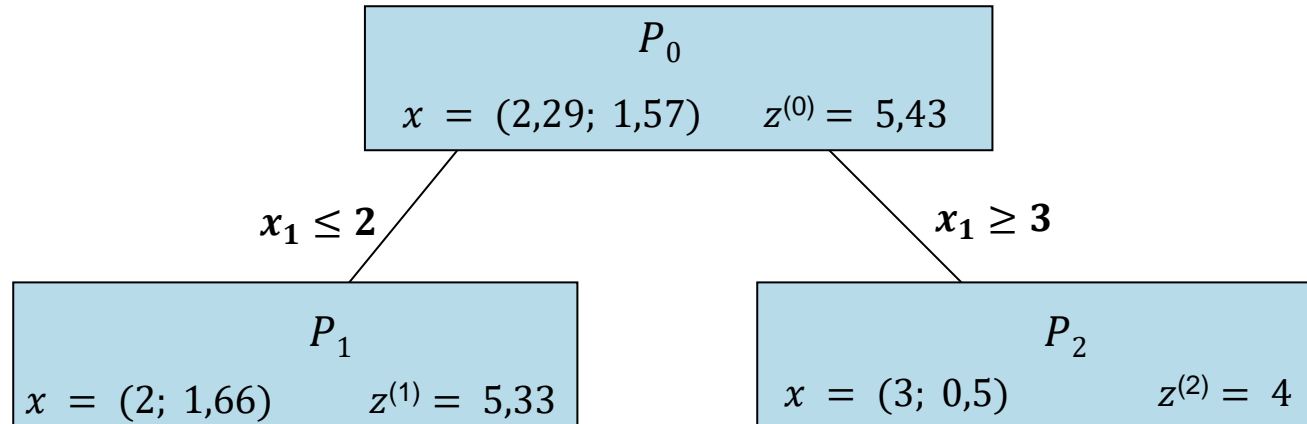
$$3x_1 + 2x_2 \leq 10$$

$$x_1 \geq 3$$

$$x_1, x_2 \geq 0$$

$$\text{Lösung: } x_1 = 3, x_2 = 1/2, z^{(2)} = 4$$

Beispiel: Lineares ganzzahliges Maximierungsproblem

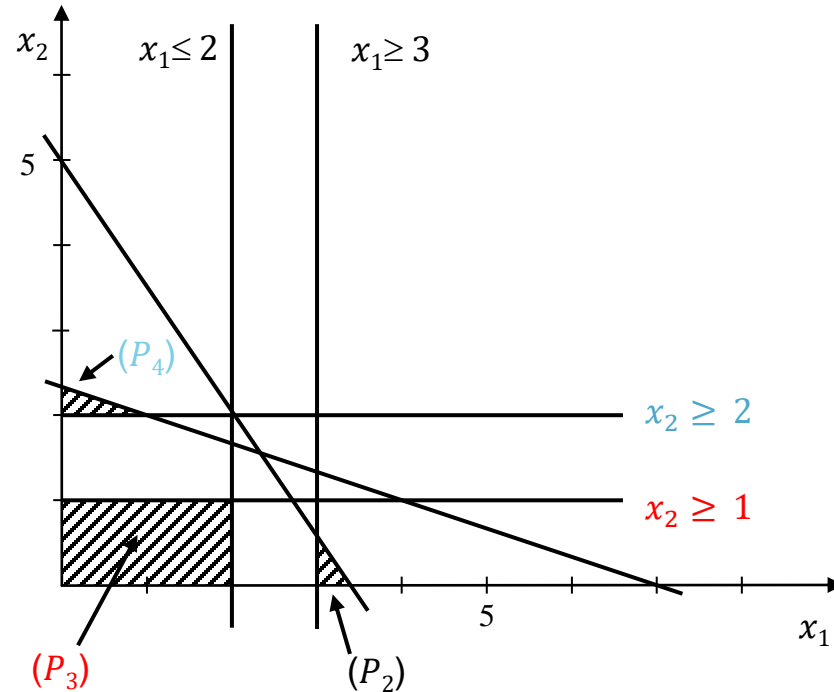


Beispiel: Lineares ganzzahliges Maximierungsproblem

In der Auslotung wird wegen $z^{(1)} > z^{(2)}$ wird auf der nächsten Stufe eine Separation von (P_1) bzgl. x_2 vorgenommen.

$$P_3: x_2^{(3)} \leq [x_2] = 1$$

$$P_4: x_2^{(4)} \geq [x_2] + 1 = 2$$



Beispiel: Lineares ganzzahliges Maximierungsproblem

$$(P_3): x_1 \leq 2, x_2 \leq 1$$

$$(P_3): \max z = x_1 + 2x_2$$

$$\text{u.d.N. } x_1 + 3x_2 \leq 7$$

$$3x_1 + 2x_2 \leq 10$$

$$x_1 \leq 2$$

$$x_2 \leq 1$$

$$x_1, x_2 \geq 0$$

$$\text{Lösung: } x_1 = 2, x_2 = 1, z^{(3)} = 4$$

$$(P_4): x_1 \leq 2, x_2 \geq 2$$

$$(P_4): \max z = x_1 + 2x_2$$

$$\text{u.d.N. } x_1 + 3x_2 \leq 7$$

$$3x_1 + 2x_2 \leq 10$$

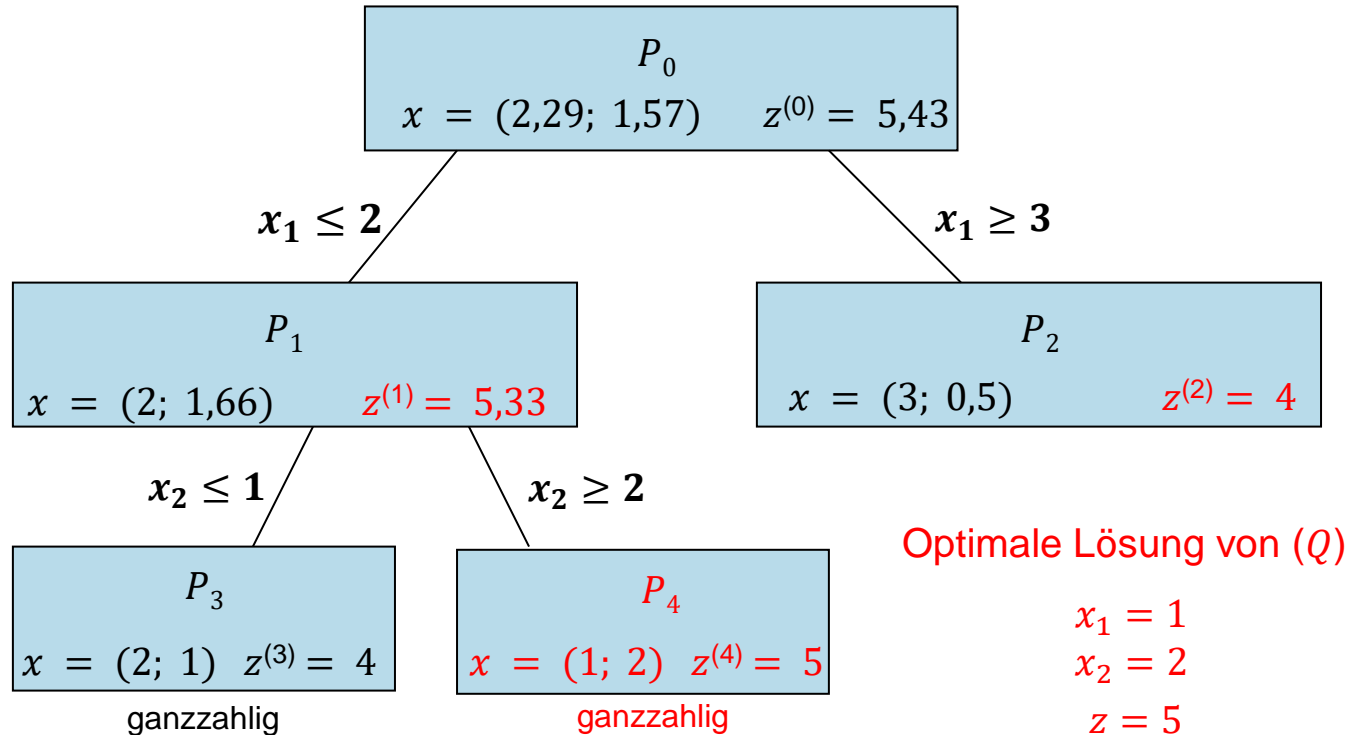
$$x_1 \leq 2$$

$$x_2 \geq 2$$

$$x_1, x_2 \geq 0$$

$$\text{Lösung: } x_1 = 1, x_2 = 2, z^{(4)} = 5$$

Beispiel: Lineares ganzzahliges Maximierungsproblem



Branch and Bound für allgemeine ganzzahlige Probleme

Übersicht des Algorithmus

0. Initialisierung: **Untere Schranke** $\underline{z} = -\infty$

1. Bilde für das relaxierte Ausgangsproblem P_0 die Lösung x^0 mit Wert \bar{z}_0

Wenn x^0 ganzzahlig, so ist x^0 die optimale Lösung: Knoten 0 ist ausgelotet \rightarrow Ende

2. Wähle einen noch nicht ausgeloteten Blattknoten k

Wenn **obere Schranke** $\bar{z}_k \leq \underline{z}$: k ist ausgelotet; gehe zu 3.

Führe eine Separation bezüglich einer der gebrochenen Variablen in x^k durch. Für jeden der zwei entstehenden Knoten i mit Problem P_i :

Bilde die relaxierte Lösung x^i mit Wert \bar{z}_i

Wenn P_i unlösbar ist: i ist ausgelotet: Gehe zu 3.

Wenn x^i ganzzahlig und $\bar{z}_i \geq \underline{z}$: Setze $\underline{z} = \bar{z}_i$ und $\underline{x} = x^i$

Wenn x^i ganzzahlig oder $\bar{z}_i \leq \underline{z}$: i ist ausgelotet: Gehe zu 3.

3. Wenn noch nicht alle Blattknoten ausgelotet sind, gehe zu 2.

Falls $\underline{z} > -\infty$, so ist \underline{x} die optimale Lösung, sonst existiert keine zulässige Lösung

Bemerkungen zum Branch and Bound

- Branch and Bound ist ein Rahmenwerk zur Ausgestaltung von Verfahren zur Lösung von gemischt ganzzahligen linearen Problemen
- Im einfachsten Fall bietet die Relaxierung von ganzzahligen Variablen in kontinuierliche Variablen eine Relaxation → oft nicht ausreichend
- Eine Relaxation soll scharfe Schranken bieten, i.e. eine geringe Abweichung des ZFW zur originalen Problemstellung.
- Die Separation soll diejenigen Variablen separieren, von denen eine größtmögliche Auswirkung auf die Lösung zu erwarten ist.
- Die Auslotung soll diejenigen Teilprobleme zur Abarbeitung selektieren, die eine starke Verbesserung der unteren Schranke bewerkstelligen.

Überblick

1. Lineare Optimierungsprobleme mit ganzzahligen Variablen
2. Das Branch and Bound Verfahren
- 3. Lösung des Rucksackproblems mit Branch and Bound**
4. Komplexitätstheorie

Anwendungen des Branch and Bound

- Standortplanung, Layoutplanung
- Reihenfolgenprobleme, Losgrößenplanung
- Maschinenbelegungsprobleme, Fließbandabstimmungsprobleme
- Briefträgerprobleme, Handlungsreisendenprobleme
- Tourenplanungsprobleme
- Rucksack-Probleme
-

Die Anwendbarkeit von Branch and Bound richtet sich nach der geschickten Ausgestaltung der drei Komponenten Relaxation, Separation und Auslotung. Im Falle „mangelnden Geschicks“ tendiert das Verfahren zur vollständigen Enumeration → extrem lange Laufzeiten

Im Folgenden die „geschickte Ausgestaltung“ für das Rucksackproblem.

Rucksackproblem: Beschreibung

- Ein Einbrecher findet in einem Haushalt acht potenzielle Beutestücke vor, die einen geschätzten Wert von EUR 150, 10, 1000, 900, 600, 150, 400 und 100 haben und 2, 10, 20, 20, 30, 30, 40 bzw. 60 Kilogramm wiegen.
- Zum Abtransport seiner Beute steht ihm ein Rucksack zur Verfügung, in dem ein Maximalgewicht von 102 kg untergebracht werden kann.
- Logischerweise besteht das Ziel des Einbrechers darin, den Gesamtwert seiner Beute zu maximieren, ohne an den Tatort zurückkehren zu müssen.
- Welche Stücke soll er mitnehmen?

Rucksackproblem: Modellierung

Entscheidungsvariablen:

$$x_i = \begin{cases} 1, & \text{falls Gegenstand } i \text{ gestohlen wird} \\ 0, & \text{sonst} \end{cases} \quad \text{für } i = 1, 2, \dots, 8$$

$$\begin{array}{ll} \text{Max} & z = 150x_1 + 1000x_2 + 900x_3 + 600x_4 + 400x_5 + 150x_6 + 100x_7 + 10x_8 \\ \text{u.d.N.} & 2x_1 + 20x_2 + 20x_3 + 30x_4 + 40x_5 + 30x_6 + 60x_7 + 10x_8 \leq 102 \\ & x_i \in \{0; 1\} \end{array}$$

Relaxation für das Rucksackproblem

- Das Rucksackproblem ist ein einfacher Spezialfall eines ganzzahligen Optimierungsproblems:
- Eine Lösung des relaxierten Problems lässt sich einfach bestimmen:
 1. Sortiere nach dem fallendem relativen Wert der Objekte
 2. Setze so lange die $x_j = 1$, bis die Kapazitätsgrenze erreicht ist
→ so erhält man eine **ganzzahlige** Lösung \underline{x} und eine **untere** Schranke \underline{z}
 3. Falls noch Kapazität vorhanden ist, so fülle auf, indem der nächsten Variable x_{j+1} ein Wert zwischen 0 und 1 zugewiesen wird
 4. → so erhält man eine **nicht-ganzzahlige** Lösung \bar{x} und eine **obere** Schranke \bar{z}

Rucksackproblem: Relaxation

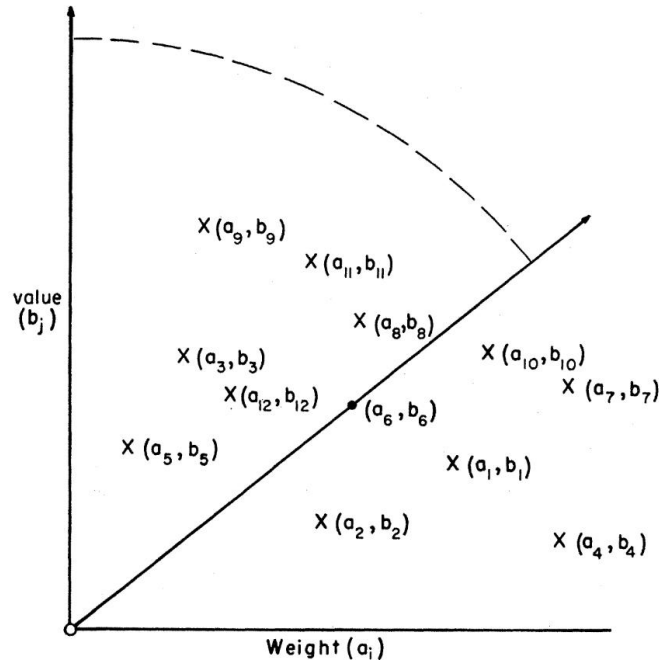


Fig. 3. Graphical solution of the knapsack problem.

Dantzig, G. B. (1957).
Discrete-variable extremum problems.
Operations Research, 5(2)

Rucksackproblem: Relaxation

Sortierung nach fallendem relativen Wert der Beutestücke:

i	1	2	3	4	5	6	7	8
Wert	150	1000	900	600	400	150	100	10
Gewicht	2	20	20	30	40	30	60	10
W/G	75	50	45	20	10	5	$5/3$	1

Lösen des **relaxierten** linearen Problems P_0

$$\max \quad z = 150x_1 + 1000x_2 + 900x_3 + 600x_4 + 400x_5 + 150x_6 + 100x_7 + 10x_8$$

$$\text{u.d.N.} \quad 2x_1 + 20x_2 + 20x_3 + 30x_4 + 40x_5 + 30x_6 + 60x_7 + 10x_8 \leq 102$$

$$x_i \in [0; 1] \quad (\text{aus dem Intervall zwischen 0 und 1})$$

Rucksackproblem: Separation

$$P_0: x_1 = x_2 = x_3 = x_4 = 1$$

$$G = 72, W = 2650 = \underline{z}_0$$

$$x_5 = 3/4$$

$$G = 102, W = 2950 = \bar{z}_0$$

$$x_6 = x_7 = x_8 = 0$$

\Rightarrow Separation bezüglich x_5 : $x_5 = 0 \vee x_5 = 1$

Rucksackproblem: Visualisierung

Beste untere Schranke

$$\underline{z} = \underline{z}_0 = 2650$$

$$\underline{x} = \underline{x}^0$$

$$\begin{aligned} P^0 \\ x^0 &= (1, 1, 1, 1, 3/4, 0, 0, 0) \\ \underline{z}_0 &= 2650 \quad \bar{z}_0 = 2950 \end{aligned}$$

Rucksackproblem: Separation

$$P_0: x_1 = x_2 = x_3 = x_4 = 1$$

$$G = 72, W = 2650 = \underline{z}_0$$

$$x_5 = 3/4$$

$$G = 102, W = 2950 = \bar{z}_0$$

$$x_6 = x_7 = x_8 = 0$$

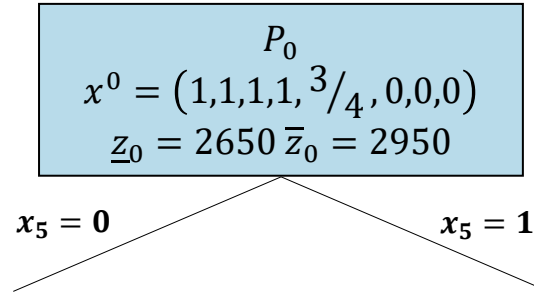
\Rightarrow Separation bezüglich x_5 : $x_5 = 0 \vee x_5 = 1$

Rucksackproblem: Visualisierung

Beste untere Schranke

$$\underline{z} = \underline{z}_1 = 2800$$

$$\underline{x} = \underline{x}^1$$



Rucksackproblem: Separation

$$P_1: \quad x_5 = 0$$

$$x_1 = x_2 = x_3 = x_4 = 1$$

$$x_6 = 1$$

$$x_7 = x_8 = 0$$

$$G = 72, W = 2650$$

$$G = 102, W = 2800 = \underline{z}_1 = \bar{z}_1$$

⇒ Ganzzahlige Lösung

⇒ $\underline{z}_1 = \bar{z}_1$

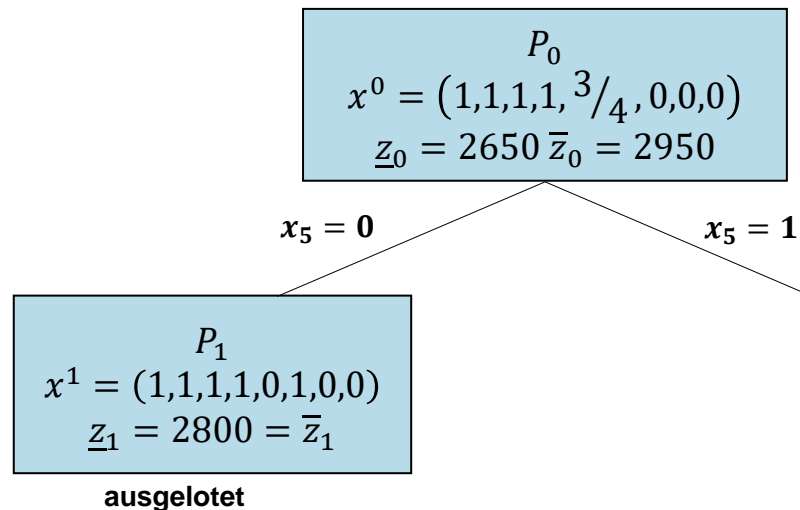
⇒ Zweig ausgelotet

Rucksackproblem: Visualisierung

Beste untere Schranke

$$\underline{z} = \underline{z}_1 = 2800$$

$$\underline{x} = \underline{x}^1$$



Rucksackproblem: Auslotung

$$\begin{array}{ll} P_2: & x_5 = 1 \\ & x_1 = x_2 = x_3 = 1 \\ & x_4 = 2/3 \\ & x_6 = x_7 = x_8 = 0 \\ & \bar{z}_2 = 2850 > 2800 = \underline{z}_1 \end{array} \quad \begin{array}{l} G = 40, W = 400 \\ G = 82, W = 2450 = \underline{z}_2 \\ G = 102, W = 2850 = \bar{z}_2 \end{array}$$

⇒ obere Schranke > größte (globale) untere Schranke

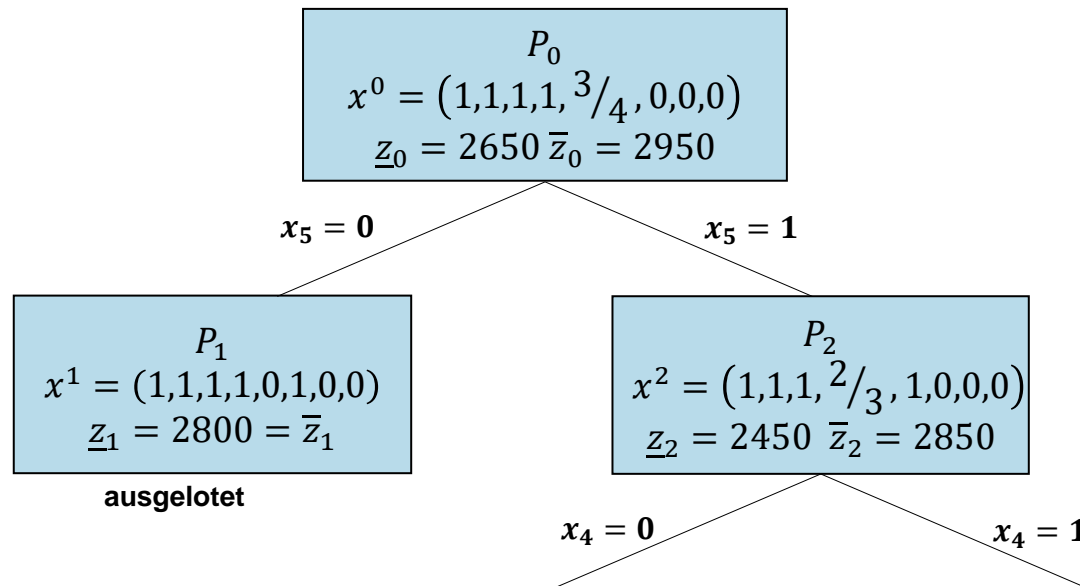
⇒ weitere Separation an P_2 bezüglich x_4 notwendig: $x_4 = 0 \vee x_4 = 1$

Rucksackproblem: Visualisierung

Beste untere Schranke

$$\underline{z} = \underline{z}_1 = 2800$$

$$\underline{x} = \underline{x}^1$$



Rucksackproblem: Auslotung

$$P_3: \quad x_5 = 1 \wedge x_4 = 0$$

$$x_4 = 0, x_5 = 1$$

$$x_1 = x_2 = x_3 = 1$$

$$x_6 = 2/3$$

$$x_7 = x_8 = 0$$

$$G = 40, W = 400$$

$$G = 82, W = 2450 = \underline{z}_3$$

$$G = 102, W = 2550 = \bar{z}_3$$

$$\bar{z}_3 = 2550 < 2800 = \underline{z}_1$$

\Rightarrow obere Schranke < größte (globale) untere Schranke

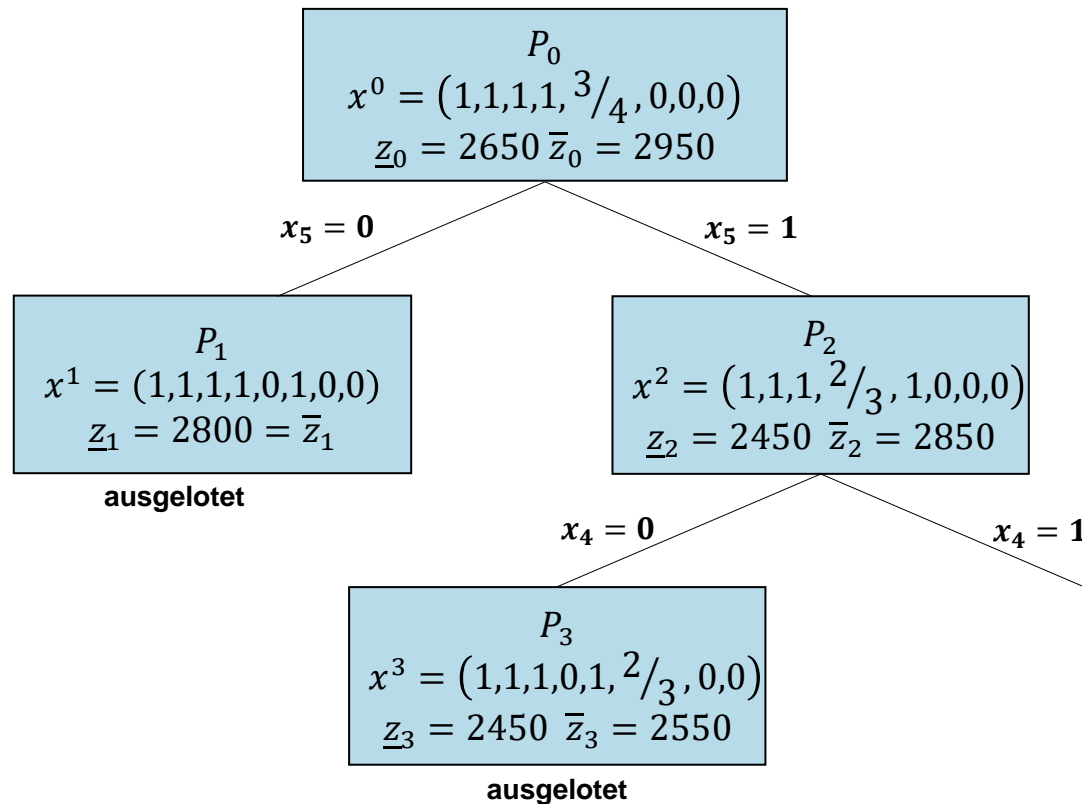
\Rightarrow Zweig ausgelotet

Rucksackproblem: Visualisierung

Beste untere Schranke

$$\underline{z} = \underline{z}_1 = 2800$$

$$\underline{x} = \underline{x}^1$$



Rucksackproblem: Ergebnis

$$P_4: \quad x_5 = 1 \wedge x_4 = 1$$

$$x_4 = x_5 = 1$$

$$x_1 = x_2 = 1$$

$$x_3 = 1/2$$

$$x_6 = x_7 = x_8 = 0$$

$$G = 70, W = 1000$$

$$G = 92, W = 2150 = \underline{z}_4$$

$$G = 102, W = 2600 = \bar{z}_4$$

$$\bar{z}_4 = 2600 < 2800 = \underline{z}_1$$

⇒ obere Schranke < größte (globale) untere Schranke

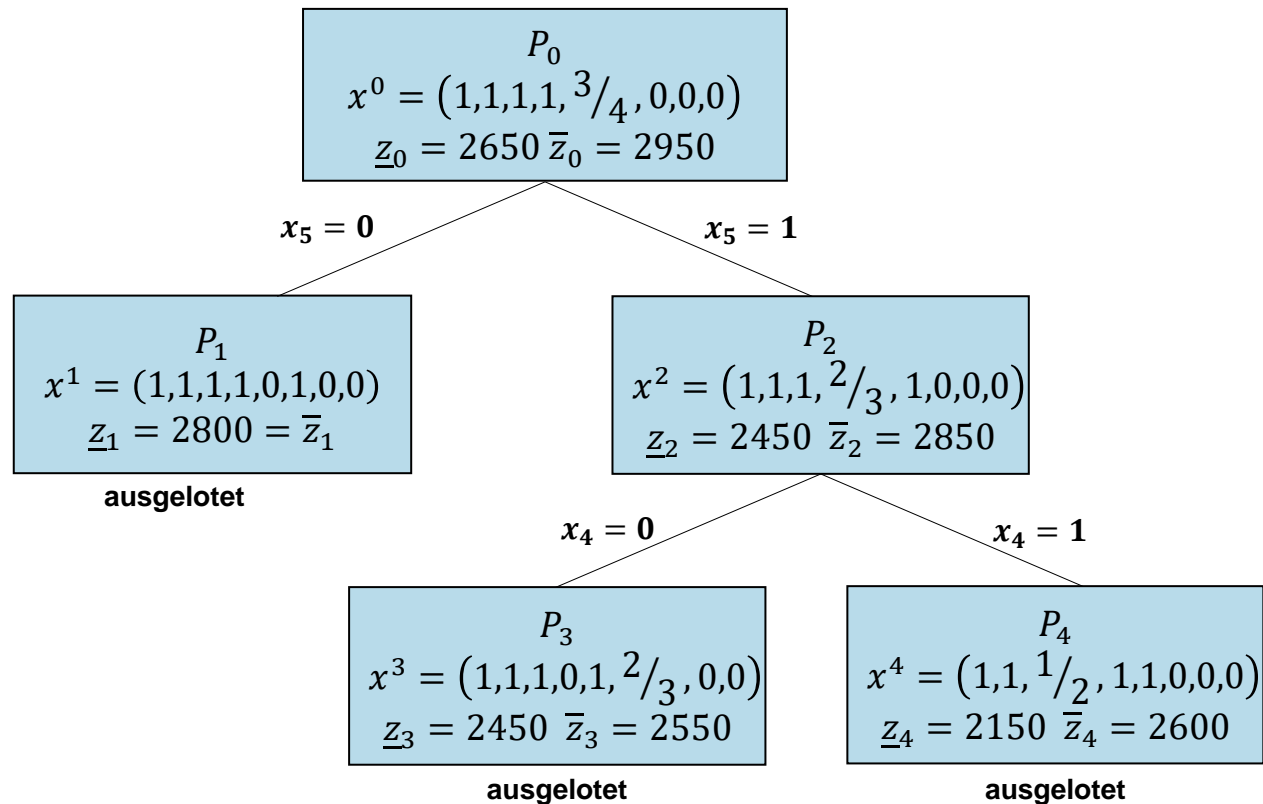
⇒ Zweig ausgelotet

Rucksackproblem: Visualisierung

Beste untere Schranke

$$\underline{z} = \underline{z}_1 = 2800$$

$$\underline{x} = \underline{x}^1$$

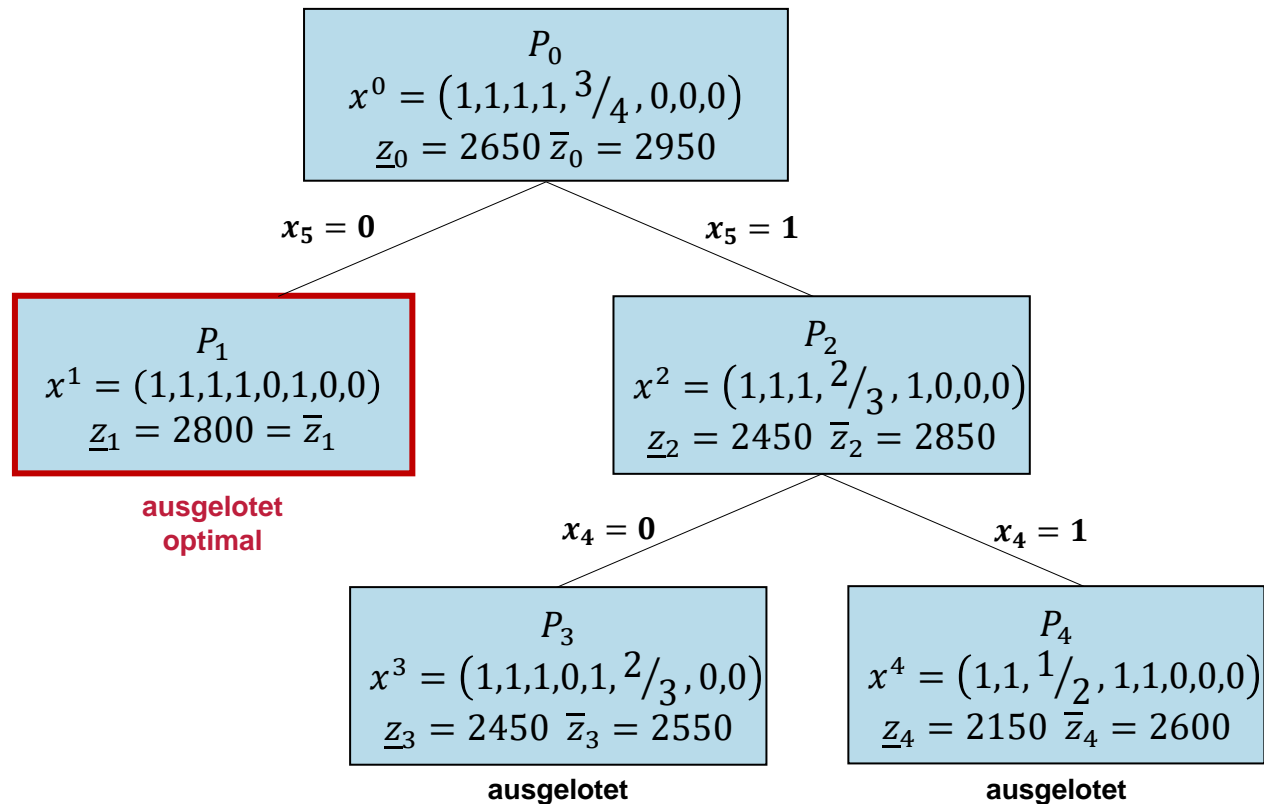


Rucksackproblem: Visualisierung

Beste untere Schranke

$$\underline{z} = \underline{z}_1 = 2800$$

$$\underline{x} = \underline{x}^1$$



Alle Blätter ausgelotet

⇒ optimale Lösung: P_1

$$\Rightarrow x^{*T} = (1,1,1,1,0,1,0,0)$$

$$\Rightarrow z^* = 2800$$

Branch and Bound für Rucksackproblem: Übersicht

1. Bilde für das Ausgangsproblem P_0 die ganzzahlige Lösung \underline{x}^0 mit Wert \underline{z}_0 und die relaxierte Lösung \bar{x}^0 mit Wert \bar{z}_0
 - Wenn \bar{x}^0 ganzzahlig, so ist \bar{x}^0 optimale Lösung: Knoten 0 ist ausgelotet \rightarrow Ende
 - Setze $\underline{z} = \underline{z}_0$ und $\underline{x} = \underline{x}^0$
2. Wähle den nicht ausgeloteten Blattknoten k mit der größten oberen Schranke \bar{z}_k
 - Wenn $\bar{z}_k \leq \underline{z}$: \underline{z} ist optimal, \underline{x} ist die optimale Lösung \rightarrow Ende
 - Führe eine Separation bezüglich der gebrochenen Variable in \bar{x}^k durch. Für jeden der zwei entstehenden Knoten i mit Problem P_i :
 - Bilde die ganzzahlige Lösung \underline{x}^i mit Wert \underline{z}_i und die relaxierte Lösung \bar{x}^i mit Wert \bar{z}_i
 - Wenn P_i unlösbar ist: i ist ausgelotet: Gehe zu 3.
 - Wenn $\underline{z}_i \geq \underline{z}$: setze $\underline{z} = \underline{z}_i$ und $\underline{x} = \underline{x}^i$
 - Wenn \bar{x}^i ganzzahlig oder $\bar{z}_i \leq \underline{z}$: i ist ausgelotet: Gehe zu 3
3. Wenn noch nicht alle Blattknoten ausgelotet sind, gehe zu 2.
 - Falls $\underline{z} > -\infty$, so ist \underline{x} die optimale Lösung, sonst existiert keine zulässige Lösung

Überblick

1. Lineare Optimierungsprobleme mit ganzzahligen Variablen
2. Das Branch and Bound Verfahren
3. Lösung des Rucksackproblems mit Branch and Bound
4. **Komplexitätstheorie**

LP vs. MILP: Komplexitätstheorie

Lineare Programme mit stetigen Variablen gehören zur Komplexitätsklasse P:

- Es gibt einen Algorithmus, der auch im schlechtesten Fall eine polynomielle Laufzeit aufweist (d.h., dass sich das Verhältnis von Problemgröße und Laufzeit durch eine polynomielle Funktion beschreiben lässt.)

Lineare Programme mit ganzzahligen Variablen gehören zur Klasse NP-Schwere:

- Es gibt wahrscheinlich **keinen** Algorithmus, der auch im schlechtesten Fall eine polynomielle Laufzeit für die Lösung einer Probleminstance aufweist.
- Als Entscheidungsproblem liegen sie in der Klasse NP.
- Es gibt Sonderfälle, die in polynomieller Zeit gelöst werden können.

Die Laufzeit auf einem Computer zur Lösung einer Instanz eines Problems wird abgeschätzt durch Polynom n^c oder aber mit c^n im NP-Fall, wobei c eine Konstante und n die Problemgröße darstellt.

Beispiel Enumeration mit $x = \begin{pmatrix} x_1 \\ \vdots \\ x_5 \end{pmatrix}$ und $x_i \in \{0,1\} \Rightarrow 2^5 = 32$ Kombinationsmöglichkeiten

\Rightarrow allgemein: 2^n Kombinationsmöglichkeiten

Sonderfall: Reduktion auf lineare Optimierungsprobleme

- Einige ganzzahlige Optimierungsprobleme lassen sich mit den Methoden der linearen Programmierung (z. B. Simplex-Algorithmus) lösen.
- Bedingungen hierfür sind:
 - Die Matrix A muss *vollständig unimodular* sein
 - Unimodular: A ist ganzzahlige Matrix mit Determinante $+1$ oder -1
 - Vollständig unimodular: Jede quadratische Submatrix ist unimodular
 - Die Einträge vollständig unimodularer Matrizen nehmen nur die Werte 0 , 1 oder -1 an
 - Der Vektor b muss *ganzzahlig* sein
- Alle Basislösungen solcher Probleme sind ganzzahlig
- Beispiele hierzu sind:
 - Transportproblem
 - Zuordnungsproblem
- Häufig wird das Transportproblem als Teilproblem einer komplexeren Problemstellung gelöst
→ divide and conquer!

LP vs MIP: Praxis / Benchmarks

Lineare Programme:

Aktuelle Benchmarks für Software unter: <http://plato.asu.edu/ftp/barrier.html>

Software: Solver ILOG CPLEX (IBM), Version 12.6.0

Hardware: 3.47 GHz Intel Xeon X5690 (6 Kerne, 48 GB RAM)

Modell	Variablen	Restriktionen	Nonzeros	Zeit(min)
in	1.526.203	1.449.074	6.813.128	2,9
pde_2	26.993.999	18.005.996	80.970.001	82,2

(Gemischt-) Ganzzahlige lineare Programme:

Aktuelle Benchmarks: <http://plato.asu.edu/ftp/milpc.html>, <http://plato.asu.edu/ftp/path.html>

Software: Solver Gurobi, Version 5.6.0

Hardware: 3.47 GHz Intel Xeon X5680 (2*6 Kerne, 32 GB RAM)

Modell	Variablen	Restriktionen	Nonzeros	Zeit(min)
map18	164.547	328.818	549.920	0,7
ns2267839	3265	395	19.324	74,21

Zusammenfassung

- Ganzzahlige Optimierung
 - Ganzzahlig lineare Probleme
 - Binäre lineare Probleme
 - Gemischt-ganzzahlige lineare Probleme
- Branch and Bound Verfahren
 - Relaxation, Separation, Auslotung
- Komplexitätstheorie