

Vorlesung 08: Ganzzahlige Optimierung

Grundlagen Operations Research

Ganzzahlige Optimierung

Nur ganze Einheiten

Was ist neu?

Für einige oder alle Variablen sind nur **ganze Zahlen** zulässig:

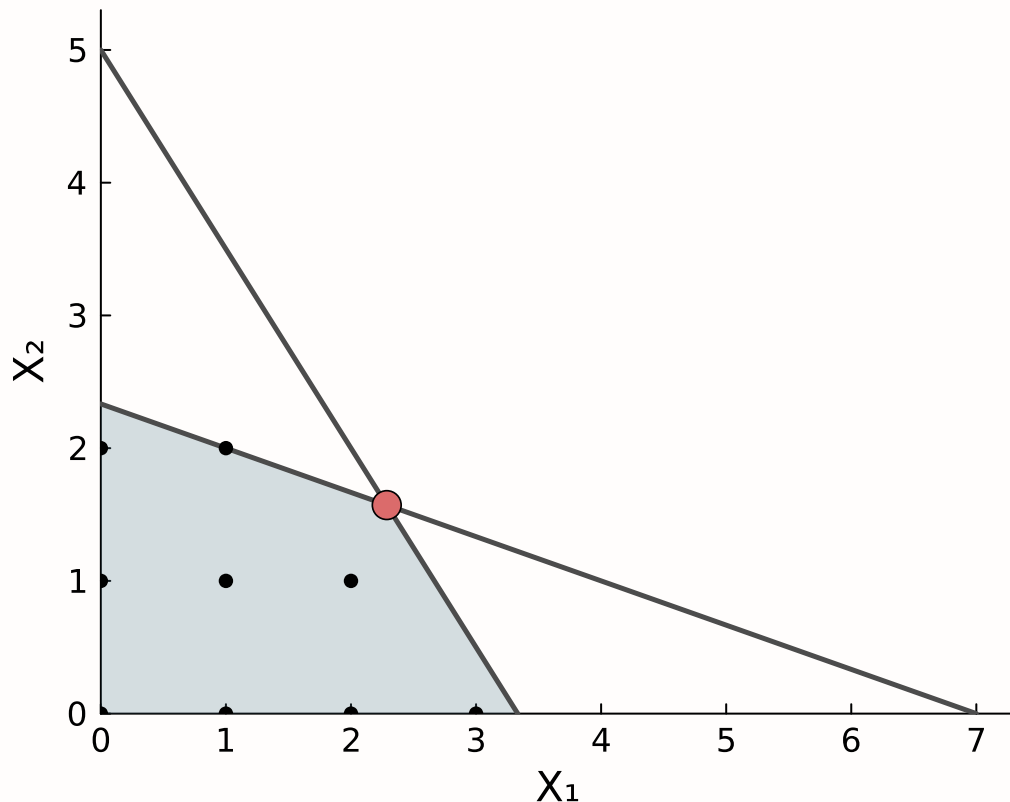
- **Ganzzahlig:** $X \geq 0$ und ganzzahlig (Stückzahlen, Fahrzeuge, ...).
- **Binär:** $X \in \{0, 1\}$ für Ja/Nein-Entscheidungen.
- **Gemischt-ganzzahlig:** ganzzahlige **und** kontinuierliche Variablen.

Lässt man die Ganzzahligkeit weg, entsteht die **LP-Relaxation**, ein gewöhnliches lineares Problem.

Warum schwierig?

- Ganzzahlige Probleme gelten als **schwer**: kein Algorithmus mit polynomialem Aufwand für die optimale Lösung bekannt.
- Naiv: **vollständige Enumeration** aller ganzzahligen Möglichkeiten.
- In der Praxis: **Meta-Heuristiken** (Genetische Algorithmen, Tabu-Suche, ...) oder das exakte **Branch-&-Bound-Verfahren**.

LP-Relaxation grafisch



Die schwarzen Punkte sind die **ganzzahligen** Kandidaten.

Das LP-Optimum $(2,29; 1,57)$ ist **nicht** ganzzahlig.

Frage: **Warum nicht einfach runden?**

Die Antwort folgt am Ende des Kapitels.

Branch & Bound

Ausgangsproblem P_0

$$\begin{aligned}\max F &= X_1 + 2X_2 \\ X_1 + 3X_2 &\leq 7 \\ 3X_1 + 2X_2 &\leq 10 \\ X_1, X_2 &\geq 0 \text{ und ganzzahlig}\end{aligned}$$

LP-Relaxation P'_0 : $X_1^* = 2,286$, $X_2^* = 1,571$, $F_0^* = 5,429 \Rightarrow$ **obere Schranke** $\overline{F}_0 = 5,429$.

Schranken (Bounding)

- Jede LP-Relaxation liefert eine **obere Schranke** für ihr Teilproblem.
- Jede bekannte **ganzzahlige** Lösung liefert eine globale **untere Schranke** $\underline{F}$.
- Start: $X_1 = X_2 = 0$ ist ganzzahlig zulässig $\Rightarrow \underline{F} = 0$.

Ein Teilproblem ist **ausgelotet**, wenn seine obere Schranke $\underline{F}$ nicht mehr übertrifft.

Die Auslotungsregeln

Ein Teilproblem P_i wird **nicht weiter verzweigt**, wenn:

- **(3a)** $\bar{F}_i \leq \underline{F}$: es kann nichts Besseres mehr liefern.
- **(3b)** die Relaxationslösung **ganzzahlig** ist und $\underline{F}$ **verbessert** ($\bar{F}_i > \underline{F}$): neue beste Lösung.
- **(3c)** die Relaxation **unzulässig** ist: dann auch P_i .

Bei **Minimierung** tauschen obere und untere Schranke die Rollen.

Relaxationen

Die LP-Relaxation ist nur **eine** Möglichkeit, Schranken zu gewinnen:

- **LP-Relaxation**: Ganzzahligkeit weglassen (unser Standard).
- **Weglassen von Nebenbedingungen**: jedes entspannte Problem schrankt.
- **Lagrange-Relaxation**: verletzte Restriktionen wandern bestraft in die Zielfunktion.

Verzweigen (Branching)

Wähle eine **nicht ganzzahlige** Variable und verzweige, um den nicht ganzzahligen Bereich auszuschließen:

$$P_1 : P_0 \text{ und } X_1 \leq 2 \quad P_2 : P_0 \text{ und } X_1 \geq 3$$

Übliche Regel: die Variable mit dem **geringsten Abstand** zur nächsten ganzen Zahl, hier X_1 ($0,286 < 0,429$).

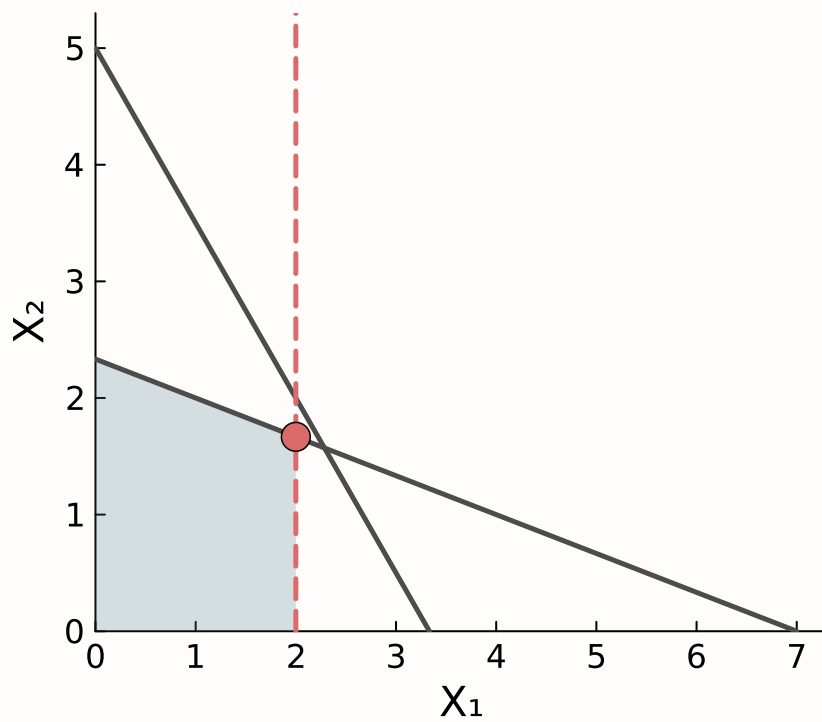
Die optimale ganzzahlige Lösung von P_0 steckt in P_1 **oder** P_2 .

Suchstrategien

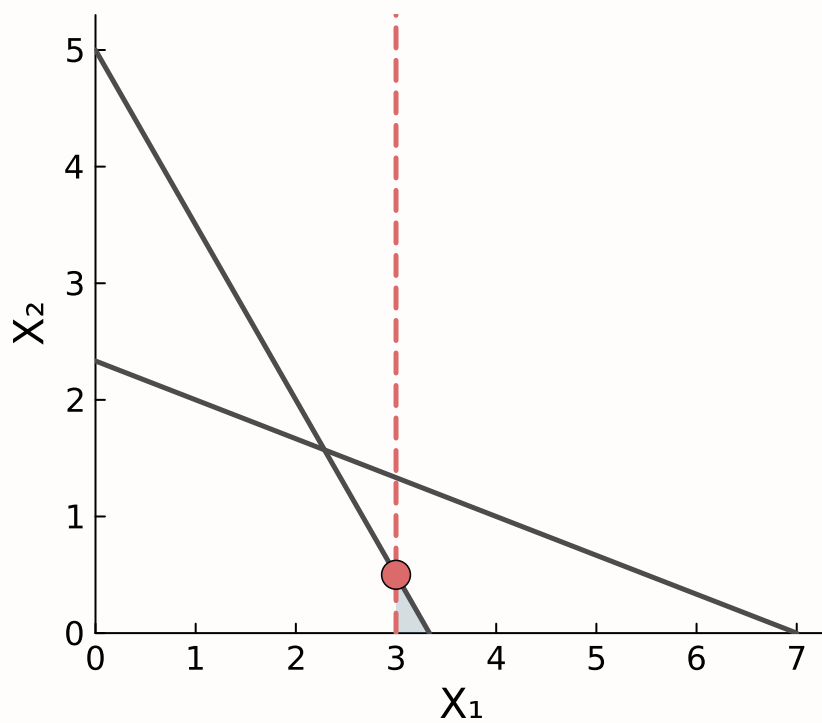
Frage: **Welches offene Teilproblem als Nächstes?**

- **Reine Tiefensuche**: einen Ast bis zum Ende, schnell eine erste Lösung.
- **Tiefensuche + MUB**: nach vollständiger Verzweigung den Ast mit **größter** oberer Schranke (Maximum Upper Bound).
- **Breitensuche**: unter **allen** offenen Knoten den mit größter Schranke.

Teilprobleme P_1 und P_2



$P_1 (X_1 \leq 2): (2, 1, 67), \bar{F}_1 = 5,33$



$P_2 (X_1 \geq 3): (3, 0, 5), \bar{F}_2 = 4$

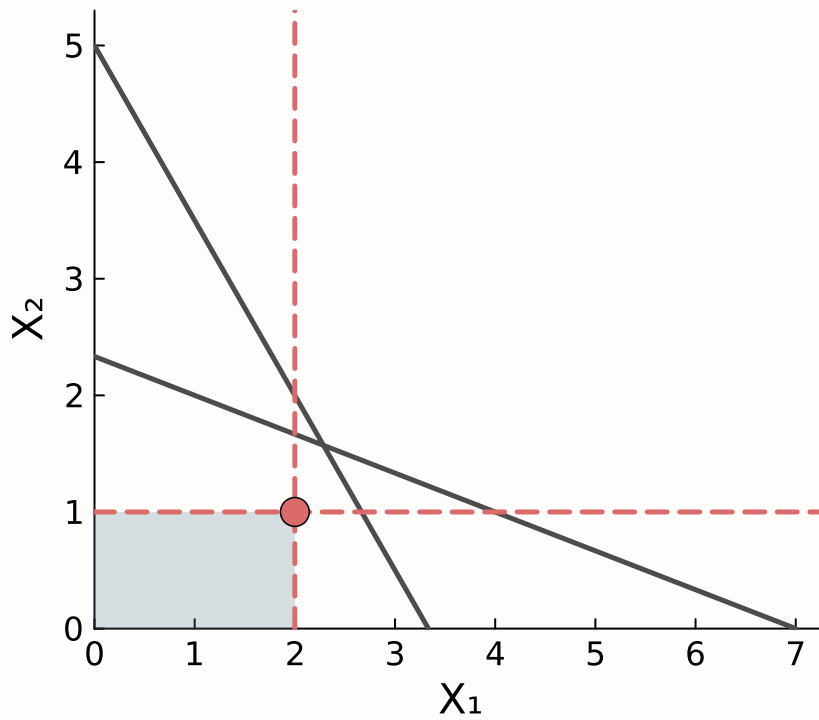
Weiter mit dem besten Ast

$\bar{F}_1 = 5,33 > \bar{F}_2 = 4 \Rightarrow$ verzweige zuerst P_1 (Maximum-Upper-Bound).

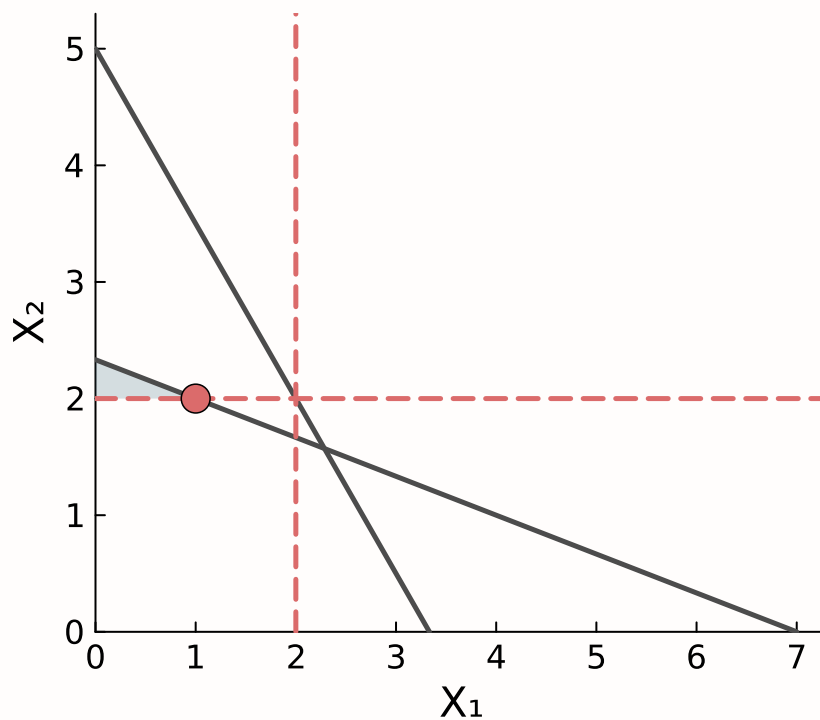
In P_1 ist $X_2 = 1,667$ nicht ganzzahlig:

$$P_3 : P_1 \text{ und } X_2 \leq 1 \quad P_4 : P_1 \text{ und } X_2 \geq 2$$

Teilprobleme P_3 und P_4



$P_3: (2,1), F_3 = 4, \text{ ganzzahlig} \Rightarrow \underline{F} = 4$



$P_4: (1, 2), F_4 = 5, \text{ ganzzahlig} \Rightarrow \underline{F} = 5$

P_2 wird ausgelotet

P_3 und P_4 sind ganzzahlig gelöst. Die beste untere Schranke ist jetzt $\underline{F} = 5$.

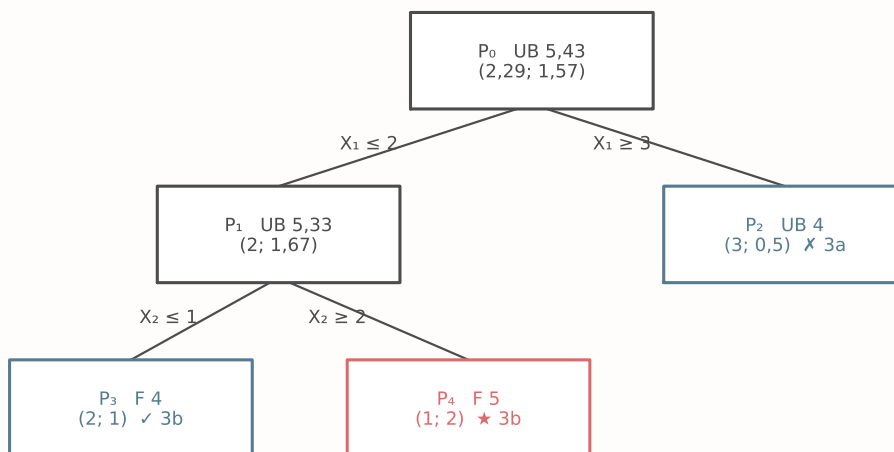
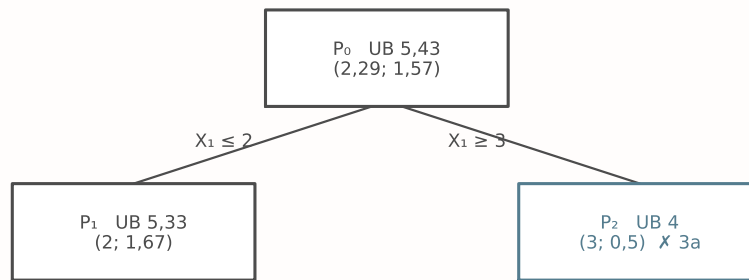
$$\overline{F}_2 = 4 \leq \underline{F} = 5 \Rightarrow P_2 \text{ ausgelotet}$$

P_2 kann **keine** bessere Lösung mehr enthalten. Es muss nicht weiter untersucht werden.

Der Branch-&-Bound-Baum

$$P_0 \quad \text{UB } 5,43$$

$$(2,29; 1,57)$$



P_0 : LP-Optimum gebrochen, UB ($= \overline{F}$) 5,43, verzweigen nach X_1 .

P_1 verspricht mehr (MUB). P_2 wartet.

P_3 und P_4 sind ganzzahlig (3b); $\underline{F} = 5$ lotet P_2 aus (3a).

Optimale Lösung

Alle Teilprobleme sind ausgelotet. Die beste ganzzahlige Lösung ist optimal:

! Optimale Lösung

$$X_1^* = 1, \quad X_2^* = 2, \quad F_0^* = 5$$

Beachten Sie: das **gerundete** LP-Optimum (2, 2) ist gar nicht zulässig. Runden führt oft in die Irre.

Kontrolle mit Julia

Ganzzahlig mit JuMP

Ganzzahligkeit ist in JuMP nur ein Zusatz. HiGHS führt Branch & Bound automatisch aus:

```
using JuMP, HiGHS

modell = Model(HiGHS.Optimizer); set_silent(modell)
@variable(modell, X1 >= 0, Int)           # ganzzahlig
@variable(modell, X2 >= 0, Int)
@objective(modell, Max, X1 + 2X2)
@constraint(modell, X1 + 3X2 <= 7)
@constraint(modell, 3X1 + 2X2 <= 10)

optimize!(modell)
println("F* = ", objective_value(modell),
        " mit X1 = ", value(X1), ", X2 = ", value(X2))
```

```
F* = 5.0 mit X1 = 1.0, X2 = 2.0
```

Binäres Beispiel

Auftragsauswahl

Ein Fuhrunternehmer kann vier Aufträge annehmen. Welche maximieren den Erlös?
Kapazität: **30 t** und **50 Paletten**.

Auftrag	1	2	3	4
Gewicht [t]	10	15	12	14
Paletten	12	30	16	10
Erlös	200	350	250	180

Das Binärmodell

Mit $X_j \in \{0, 1\}$ (Auftrag j annehmen?):

$$\begin{aligned}\max F &= \sum_j e_j X_j \\ \sum_j g_j X_j &\leq G(\text{Gewicht}) \\ \sum_j p_j X_j &\leq P(\text{Paletten}) \\ X_j &\in \{0, 1\}\end{aligned}$$

LP-Relaxation: statt $X_j \in \{0, 1\}$ nur $0 \leq X_j \leq 1$.

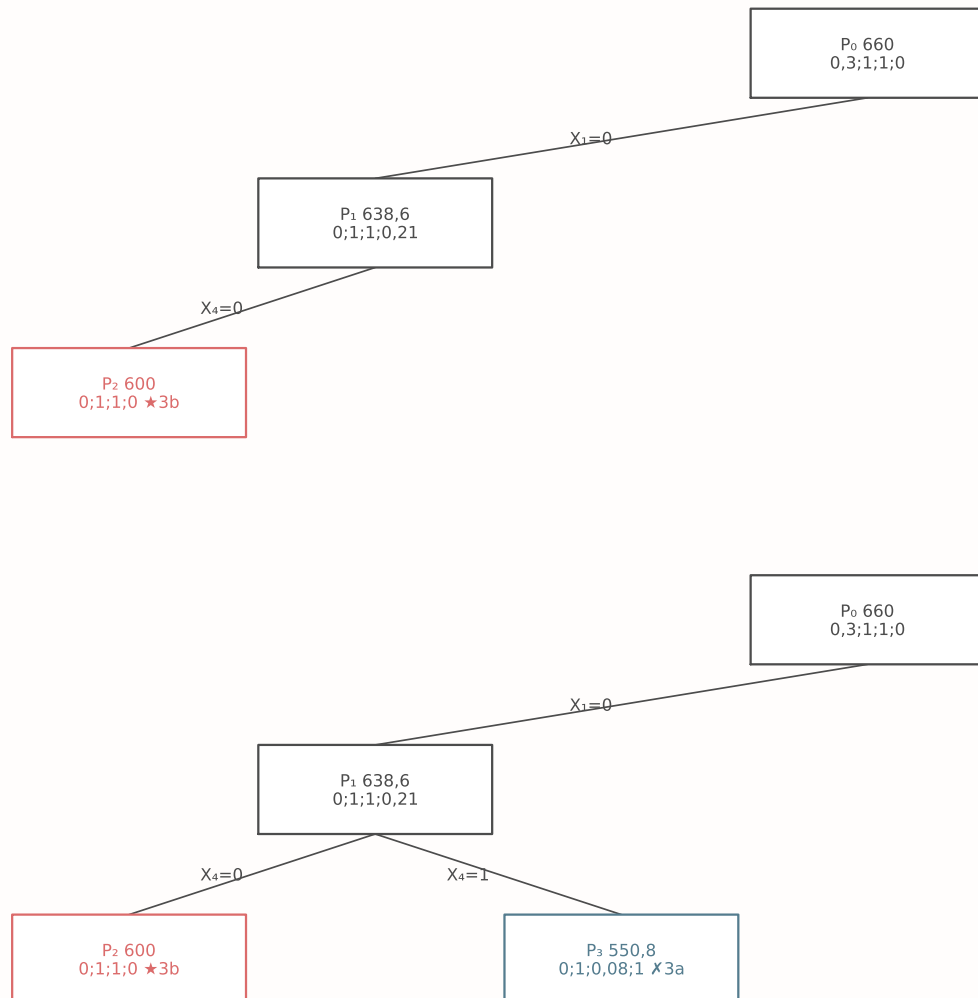
LP-Relaxation lösen

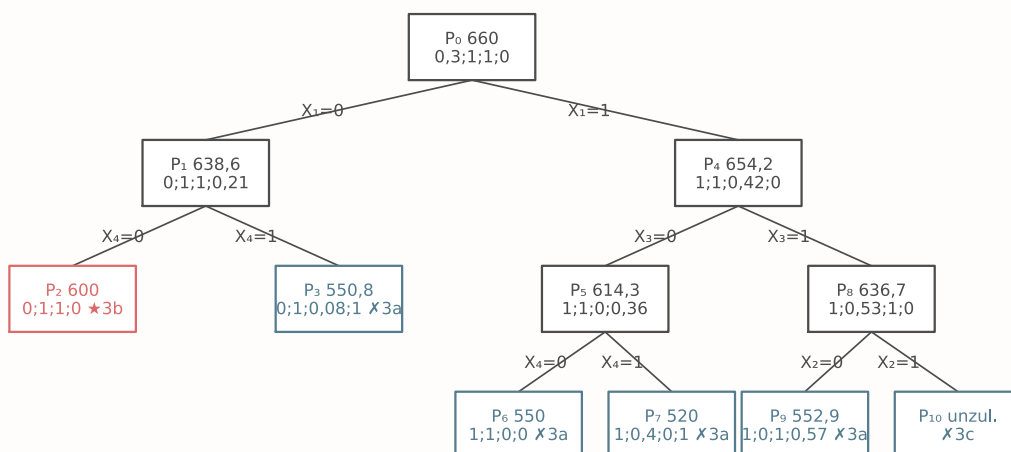
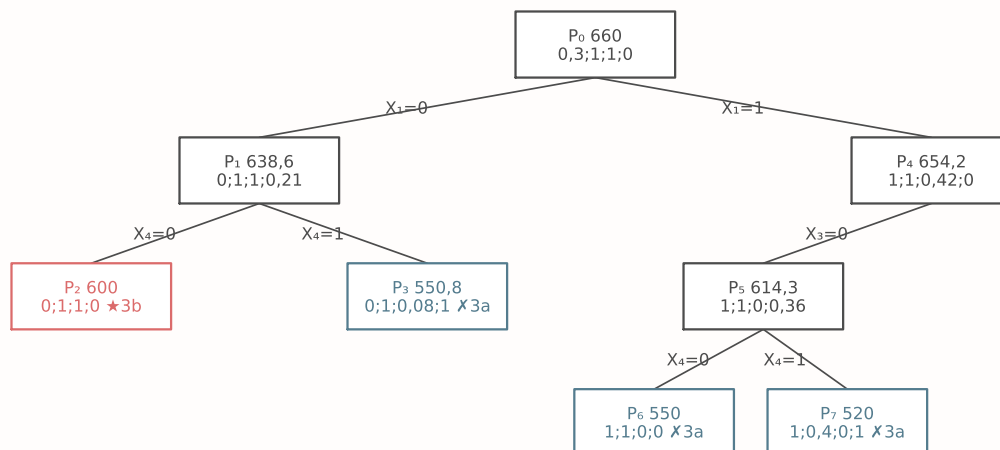
Die LP-Relaxation ist nicht ganzzahlig:

$$X^* = (0, 3, 1, 1, 0), \quad F_0^* = 660$$

$X_1 = 0,3$ ist gebrochen $\Rightarrow$ verzweige nach X_1 , und zwar, da binär, in die Äste $X_1 = 0$ und $X_1 = 1$.

Reine Tiefensuche





Immer $X_j = 0$ zuerst, tief hinab: $P_2 = (0, 1, 1, 0)$ ist ganzzahlig (3b) $\Rightarrow \underline{F} = 600$.

Zurück und rechts: P_3 mit 550, $8 \leq 600$, ausgelotet (3a).

Rechter Ast: P_5 verspricht 614, 3, aber P_6 und P_7 (beide 3a) bringen nichts Besseres.

P_9 (3a) und P_{10} (unzulässig, 3c) beenden die Suche. P_2 **bleibt** optimal.

Optimum

! Beste Auftragsauswahl

Aufträge **2 und 3**: $X^* = (0, 1, 1, 0)$, $F^* = 600$.

Von $2^4 = 16$ möglichen Auftragskombinationen hat Branch & Bound nur einen Teil **explizit** geprüft. Der Rest war implizit ausgelotet: **implizite vollständige Enumeration**.

Kontrolle mit Julia

```
using JuMP, HiGHS
e = [200,350,250,180]; g = [10,15,12,14]; p = [12,30,16,10]
G, P, J = 30, 50, 1:4

modell = Model(HiGHS.Optimizer); set_silent(modell)
@variable(modell, X[J], Bin)
@objective(modell, Max, sum(e[j]*X[j] for j in J))
@constraint(modell, sum(g[j]*X[j] for j in J) <= G)
@constraint(modell, sum(p[j]*X[j] for j in J) <= P)

optimize!(modell)
println("F* = ", round(Int, objective_value(modell)),
        ", angenommen: ", [j for j in J if value(X[j]) > 0.5])
```

```
F* = 600, angenommen: [2, 3]
```

Zusammenfassung

Das Wichtigste auf einen Blick

- Die **LP-Relaxation** lässt die Ganzzahligkeit weg und liefert eine **obere Schranke**.
- **Branch & Bound** verzweigt nach nicht ganzzahligen Variablen und **lotet** Teilprobleme über Schranken aus.
- Ein Teilproblem ist erledigt, wenn es ganzzahlig ist **oder** seine obere Schranke $\underline{F}$ nicht übertrifft.
- **Runden** der LP-Lösung ist keine verlässliche Methode.

Literatur

Literaturverzeichnis

Bibliografie