An example which leads to a larger B&B Baum

max $3x_1 + x_2$ (geänderte Zielfunktion ursprünglich war $x_1 + x_2$

udNb

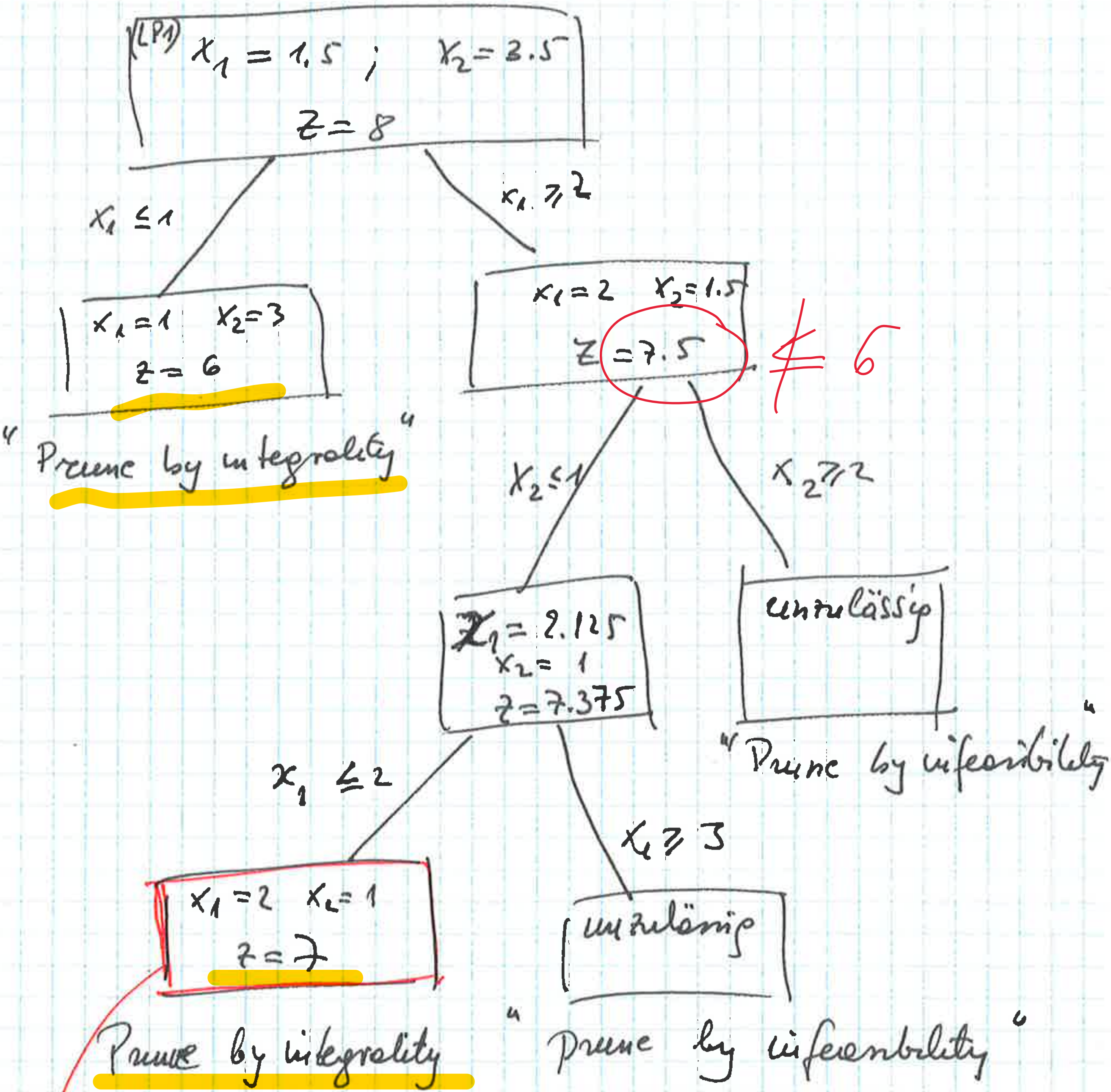
$$-x_1 + x_2 \le 2$$

(IP) $$8x_1 + 2x_2 \le 19$$

$$x_1, x_2 \ge 0$$

$$x_1, x_2 \in \mathbb{Z}$$

Sei LP_1 die lineare Relaxation von IP



Optimale Lösung $x_1 = 2$ $x_2 = 1$

mit optimalem Zielfunktionswert 7.

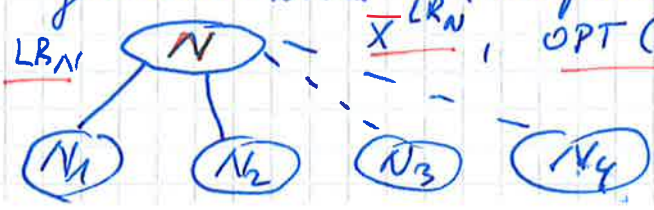
BBA

Der branch & bound Algorithmus zur Lösung von MILP

BBA löst mehrere (im Allgemeinen exponentiell viele) lineare Optimierungsprobleme um zur optimalen Lösung eines MILP zu gelangen.

Die Idee ist die Menge der zulässigen Lösungen des MILP stets in Teilmengen zu partitionieren. branch
Jede Teilmenge wird als Schnitt des $\mathbb{Z}^n$ (n ist die Anzahl der Variablen) mit einem Polytop/Polyeder, das als Menge von zulässigen Lösungen eines linearen Programs gegeben ist.
Über jede solche Partition der Menge der zulässigen Lösungen von MILP wird die ursprüngliche Zielfunktion optimiert. Das sind die sogenannten Teilprobleme. Dazu wird die entsprechende lineare Relaxation gelöst um Schranken (bounds) für den optimalen Zielfunktionswert zu berechnen. Die Teilprobleme werden in den Knoten eines Baums mit Wurzel organisiert.
In der Wurzel wird das ursprüngliche Problem abgebildet; die Konstruktion des Baums erfolgt derart, dass die Menge der zulässigen Lösungen des im Knoten N abgebildeten MILP (Teilproblem) gleich der Vereinigung der Mengen der zulässigen Lösungen der in den unmittelbaren Nachfolgerknoten von N dargestellten MILP (Teilprobleme) ist.

$MILP_N$, LR_N — $\underline{x}^{LR_N}$, $OPT(LR_N)$, $OPT(MILP_N)$



In jedem Knoten N wird die lineare Relaxation LR_N des dazugehörigen Problems $MILP_N$ gelöst. Der optimale Zielfnkt.-wert $OPT(LR_N)$ ist eine lokale untere bzw. obere Schranke für ~~den~~ den optimalen Zielfunktionswert von $MILP_N$ je nach dem ob $MILP_N$ ein Minimierungs- bzw. Maximierungsproblem ist.

(i) Wenn die Optimallösung $\bar{x}^{LR_N}$ von LR_N, $\bar{x}_J^{LR_N} \in \mathbb{Z}_J^n$, erfüllt, so ist $\bar{x}^{LR_N}$ auch eine Optimallösung von $MILP_N$, N braucht nicht weiter partitioniert werden (fathom by integrality). In diesem Fall is $OPT(LR_N)$ eine obere bzw. untere globale Schranke für den optimalen Zielfnk.wert $OPT(MILP)$ des ursprünglichen Problems, je nach dem ob letzteres ein Minimierungs- bzw. Maximierungsproblem ist. BBA speichert stets die beste gefundene globale obere bzw. untere Schranke („beste = kleinste" bzw „beste = größte"). Wir bezeichnen diese Schranke mit GB.

(ii) Wenn LR_N unzulässig, so ist auch $MILP_N$ unzulässig, d.h. der Knoten N braucht nicht weiter partitioniert werden (fathom by infeasibility)

(iii) Ansonsten sei $OPT(LR_N)$ der Opt. Zielfnkwert von LR_N. Wenn MILP ein Minimierungsproblem (Maximierungsproblem) ist und $OPT(LR_N) \geq GB$ ($OPT(LR_N) \leq GB$) gilt, dann kann keine zulässige Lösung von $MILP_N$ einen besseren Zielfnk.wert als GB liefern. N braucht nicht weiter partitioniert werden (fathom by bound).

Sonst $\exists\, j \in J$ (J ist die Menge der Indizes der ganzzahligen Variablen) mit $\bar{x}_j^{LR_N} \notin \mathbb{Z}$. Ein derartiger Index wird gewählt (variable selection) um zu partitionieren; $MILP_N$ kann zB in $MILP_{N^-}$ und $MILP_{N^+}$ partitioniert werden, wobei die zulässigen Lösungen von $MILP_{N^-}$ ($MILP_{N^+}$) durch die Restriktionen von $MILP_N$ zusätzlich $x_j \leq \lfloor \bar{x}_j^{LR_N} \rfloor$ ($x_j \geq \lceil \bar{x}_j^{LR_N} \rceil$) definiert sind. Die neuen Probleme $MILP_{N^-}$, $MILP_{N^+}$ werden dem B&B Baum als unmittelbare Nachfolger Knoten von N hinzugefügt. Nun wählt der Algorithmus nicht eliminiertes Blatt des B&B Baums, löst die dazugehörige lineare Relaxation usw. Dieses Prozedere wird wiederholt solange es noch nicht eliminierte Blätter im B&B Baum gibt.

Die Menge der Blätter des B&B Baums wird mit $\mathcal{L}$ bezeichnet; das ist eine Liste die stets aktualisiert wird. Die beste bereits gefundene Lösung von MILP wird mit x^* bezeichnet (incumbent solution)

GB wird mit dem Zielfunktionswert einer bekannten zulässigen Lösung initialisiert; wenn keine solche existiert wird mit $GB = +\infty$ im Fall eines Minimierungsproblems und mit $GB = -\infty$ im Fall eines Maximierungsproblems initialisiert.

Die Wurzel wird mit N_0 bezeichnet und stellt das ursprüngliche Problem MILP dar.

Im Folgenden wird ein generischer BBA für ein Minimierungsproblem dargestellt.

Ein generischer BBA für ein Minimierungsproblem

1: $\mathcal{L} := \{N_0\}$; $GB := +\infty$; $x^* := NULL$; /* Initialisierung */

2: If $\mathcal{L} = \emptyset$, output x^* als Optimallösung und terminiere; /* Terminierung */
end if

3: Wähle $N_i \in \mathcal{L}$ und lösche es aus $\mathcal{L}$; /* Knotenselektion */

4: Löse LR_{N_i}. If LR_{N_i} unzulässig, then goto 2; /* fathom by infeasibility */
end if

5: If $OPT(LR_{N_i}) \geq GB$, then goto 2; /* fathom by bound */
end if

6: If $\bar{x}_j^{LR_{N_i}} \in \mathbb{Z}$, $\forall j \in J$, then $GB := OPT(LR_{N_i})$
$x^* = \bar{x}^{LR_{N_i}}$;
goto 2;
/* aktualisiere Schranke und beste gefundene Lösung */

7: else wähle $j \in J$, mit $\bar{x}_j^{LR_{N_i}} \notin \mathbb{Z}$ /* Selektion der Variable */
und konstruiere N_i^-, N_i^+ durch Hinzufügen einer neuen zusätzlichen Restriktion
$$x_j \leq \lfloor \bar{x}_j^{LR_{N_i}} \rfloor \quad \text{bzw.} \quad x_j \geq \lceil \bar{x}_j^{LR_{N_i}} \rceil.$$
Füge N_i^-, N_i^+ der Liste $\mathcal{L}$ hinzu und goto 2;
/* Verzweigung aus N_i */
end if

Es gibt eine Reihe von konkreten Implementierungs-relevanten Fragestellungen, zB

(a) Selektion der Variable
(b) Selektion des Knoten

Ad (a): Seien $N_{i,j}^-$, $N_{i,j}^+$ die ~~Probleme~~, die ~~durch~~ Verzweigung mit Hilfe von x_j entstehen.

Sei $D_{ij}^- := -OPT(LR_{N_i}) + OPT(LR_{N_{ij}^-})$

$D_{ij}^+ := -OPT(LR_{N_i}) + OPT(LR_{N_{ij}^+})$

Eine heuristische Auswahlregel: $j \in \underset{j \in J;\ \bar{x}_j^{LR_{N_i}} \notin \mathbb{Z}}{\operatorname{argmax}} \min\{D_{ij}^-, D_{ij}^+\}$

Möglichkeiten zur Ausführung:
Berechne D_{ij}^-, D_{ij}^+ $\forall j \in J$ mit $\bar{x}_j^{LR_{N_i}} \notin \mathbb{Z}$
Das heißt Strong branching

Eine alternative Auswahlregel: simple branching

$$j \in \underset{j \in J,\ \bar{x}_j^{LR_{N_i}} \notin \mathbb{Z}}{\operatorname{argmax}} \min\left\{ \underbrace{\bar{x}_j^{LR_{N_i}} - \lfloor \bar{x}_j^{LR_{N_i}} \rfloor}_{=: f_j^i},\ \underbrace{\lceil \bar{x}_j^{LR_{N_i}} \rceil - \bar{x}_j^{LR_{N_i}}}_{= 1 - f_j^i} \right\}$$

(branch on the most fractional variable)

$x_1 = 1.6$ $\quad f_j = 0.6$
min $\to 0.4$

Eine hybride Auswahlregel
Strong branching nur für Variablen mit frationellem Teil bei ca. 0.5.

$x_2 = 1.7$ $\quad f_j^2 = 0.7$
min $\to 0.3$

Führe die sogenannten Pseudokosten ein:

$$P_j^- = \frac{D_{ij}^-}{f_j^i} \qquad P_j^+ = \frac{D_{ij}^+}{1 - f_j^i}$$

Empirische Beobachtung: P_j^-, P_j^+ bleiben ~~ab~~ einer gewissen Tiefe des B&B Baumes ungefähr konstant.

Am Anfang führt BBA ein Strong Branching durch (d.h. in der Wurzel und für die ersten c Mal bei denen x_j, $j \in J$ einen fraktionellen Wert in einem $\bar{x}^{LRN_i}$ annimmt). Ermittle $\forall j$ die durchschnittlichen Werte von P_j^-, P_j^+.

Im weiteren Verlauf werden D_{ij}^-, D_{ij}^+ durch

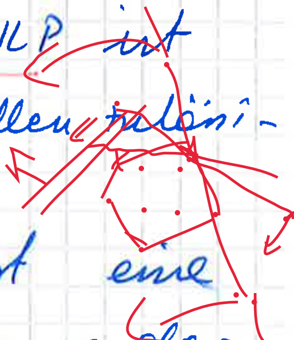
$$P_j^- \cdot f_j^i \quad \text{bzw.} \quad P_j^+(1-f_j^i) \quad \text{approximiert.}$$

Ad (b) Selektion des Knoten

$$\text{LIFO}, \ \text{FIFO}, \qquad i \in \arg\min_{N_i \in \mathcal{L}} OPT(LR_{N_i})$$

In diesem Fall muss die lineare Relaxation ~~in~~ einem Knoten unmittelbar nach der Erzeugung des Knoten und dessen Hinzufügung im B&B Baum gelöst werden.

Cutting planes (Schnittebenen)



Def: Eine gültige Ungleichung für ein MILP ist eine lineare Ungleichung, die von allen zulässigen Lösungen des MILP erfüllt ist.

Eine Schnittebene (~~a~~ cutting plane) ist eine gültige Ungleichung, die einige zulässige Lösungen der linearen Relaxation LR des MILP abschneidet, die für MILP nicht zulässig sind.

(abschneidet ≡ ist von der zul. Lösung nicht erfüllt)

Idee: Wenn die Opt.lösung $\bar{x}$ von LP nicht zulässig für das MILP ist (i.e. $\exists j \in J$ $\bar{x}_j \notin \mathbb{Z}$), dann bestimme eine Schnittebene zB $e^t x \le d$ und versuche im Weiteren

$$\begin{aligned} \min\ & c^t x \\ \text{udNb}\ & A x \ge b \\ & \boxed{e^t x \ge d} \\ & x \in \mathbb{R}^n_+ \\ & x_j \in \mathbb{Z},\ \forall j \in J \end{aligned} \qquad \text{statt} \qquad \begin{aligned} \min\ & c^t x \\ \text{udNb}\ & A x \ge b \\ & x \in \mathbb{R}^n_+ \\ & x_j \in \mathbb{Z},\ \forall j \in J \end{aligned}$$

zu lösen

Gomory's Ansatz (1960)

Betrachte die Restriktionen in der Form $Ax = b$. Insbesondere sei eine Gleichung in $Ax = b$ als

$$(*) \quad \sum_{j \in J} \underbrace{a_j}_{= f_j + \lfloor a_j \rfloor} x_j + \sum_{j \notin J} a_j x_j = b = f_0 + \lfloor b \rfloor \text{ gegeben}$$

$\lfloor a_j \rfloor x_j \in \mathbb{Z}$ für x zul für MILP

Sei $b \notin \mathbb{Z}$ und sei $f_0 := b - \lfloor b \rfloor$, $f_0 \in (0,1)$.

$\forall j \in J$ sei $f_j := a_j - \lfloor a_j \rfloor$, $f_j \in [0,1)$.

Setze in (*) ein

$$\sum_{j \in J:\, f_j \le f_0} f_j x_j + \sum_{j \in J,\, f_j > f_0} (f_j - 1) x_j + \sum_{j \notin J} a_j x_j = k + f_0 \quad \le -1 + f_0$$

wobei $k \in \mathbb{Z}$. Es gilt also $k \le -1$ oder $k \ge 0$.

$$- \sum_{j \in J:\, f_j \le f_0} \frac{f_j}{1-f_0} x_j + \sum_{\substack{j \in J \\ f_j > f_0}} \frac{1-f_j}{1-f_0} x_j - \sum_{j \notin J} \frac{a_j}{1-f_0} x_j \ge 1$$

oder

$$\sum_{\substack{j \in J \\ f_j \le f_0}} \frac{f_j}{f_0} x_j + \sum_{\substack{j \in J \\ f_j > f_0}} \frac{1-f_j}{f_0} x_j + \sum_{j \notin J} \frac{a_j}{f_0} x_j \ge 1$$

Das hat die Form $\sum_{j \in 1..n} e_j x_j \geq 1$ oder $\sum_j d_j x_j \geq 1$

und impliziert $\sum_{j \in 1..n} \max\{e_j, d_j\} x_j \geq 1$, weil $x_j \geq 0$ $\forall j$.

Welcher der Koeff. e_j, d_j ist der größte?

$\forall j$ einer der Koeff ist positiv und der andere negativ.

Wir erhalten somit

$$\sum_{\substack{j \in J \\ f_j \leq f_0}} \frac{f_j}{f_0} x_j + \sum_{\substack{j \in J \\ f_j > f_0}} \frac{1-f_j}{1-f_0} x_j + \sum_{\substack{j \notin J \\ a_j > 0}} \frac{a_j}{f_0} x_j - \sum_{\substack{j \notin J \\ a_j < 0}} \frac{a_j}{1-f_0} \geq 1 \qquad (**)$$

Diese Ungleichung heißt Gomory Schnitt mit $x_j \in \mathbb{Z}$ $\forall j \in J$ (Gomory cut) und ist $\forall x \in \mathbb{R}_+^n$, das die Gleichung $(*)$ erfüllt

<u>Beispiel</u>

max $x_1 + x_2$

u.d.NB

$$-x_1 + x_2 + x_3 = 2$$
$$8x_1 + 2x_2 + x_4 = 18$$
$$x_1, x_2, x_3, x_4 \geq 0$$
$$x_1, x_2 \in \mathbb{Z}$$

schlach

<u>Optimale Basis</u> (simplex) $B = \{1, 2\}$

$$\text{(LR)} \quad \begin{aligned} x_1 - 0.2\, x_3 + 0.1\, x_4 &= 1.5 \quad (1) \\ x_2 + 0.8\, x_3 + 0.1\, x_4 &= 3.5 \\ x_1, x_2, x_3, x_4 &\geq 0 \end{aligned}$$

mit Basislösung $x_1 = 1.5, x_2 = 3.5, \quad x_3 = x_4 = 0$ und ZFWert $= 5$. Nun erzeugen wir den Gomory Schnitt zur Restriktion $x_1 - 0.2\,x_3 + 0.1\,x_4 = 1.5$.

$f_0 = 0.5 \quad f_1 = f_2 = 0 \quad a_3 = -0.2 \quad a_4 = 0.1$

Mit der Formel (**) erhalten wir

$$\frac{0.2}{1-0.5}\,x_3 + \frac{0.1}{0.5}\,x_4 \geq 1 \iff 2\,x_3 + x_4 \geq 5 \quad (***)$$

Mit Hilfe von $x_3 = 2 + x_1 - x_2$ und $x_4 = 18 - x_1 - 2x_2$

erhalten wir aus (***)

$$3x_1 + 2x_2 \leq 9$$

Gomory Cut zu Restrikt. (1)

und die verbesserte LR:

$$(LR') \quad \begin{aligned} \max \quad & x_1 + x_2 \\ & -x_1 + x_2 \leq 2 \\ & 8x_1 + 2x_2 \leq 18 \\ & 3x_1 + 2x_2 \leq 9 \\ & x_1, x_2 \geq 0 \end{aligned}$$

Die optimale Lösung von LR' ist $x_1 = 1, \ x_2 = 3$ mit ZFWert 4; diese Lösung ist zulässig für das ursprüngliche MILP und somit auch optimal dafür.