Kombinatorische Optimierung - Blatt 2

Franziska Becker 851206
Tobias Loske 963067
Andrea Krusenbaum 964188

Aufgabe 3

a) Wie erhält man eine gute Startlösung?

→ Objekte nach $\frac{c_i}{w_i} \searrow$ sortieren und dann nacheinander einpacken, ~~bis ein~~ falls die Kapazität nicht überschritten ist.

$x_i \in \{0,1\}$ (0 = nicht einpacken, 1 = einpacken)

for i = 1 to n do
  if $w_i \leq W$
    $x_i = 1$
    $W = W - w_i$
endfor

Die Startlösung liefert eine vorläufige Upper Bound (UB).

Wie wird die Zerlegung in Teilprobleme realisiert? ✓

→ Objekte nach Sortier-Reihenfolge betrachten und dabei binär branchen ($x_i = 0$ nicht einpacken, $x_i = 1$ einpacken)

Wie sieht die Branching Reihenfolge aus? ✓

→ Sortierung nach $\frac{c_i}{w_i} \searrow$ folgend

Branch mit höchstem UB zuerst betrachten ✓

Wie werden sinnvolle obere Schranken berechnet?

→ $UB := \sum_{i=1}^{k} c_i \bar{x}_i + \left(W - \sum_{i=1}^{k} w_i \bar{x}_i\right) \cdot \frac{c_{k+1}}{w_{k+1}}$

bessere Schranke: löse fraktionales Rucksackpr. für k+1,n

$\bar{x}_i$:= alle bisher ~~fest~~ eingepackten/festgelegten Elemente $\in \{0,1\}$

k := zuletzt betrachtetes Objekt, das festgelegt wurde (nach Sortier-Reihenfolge)

$UB \in \mathbb{R}$: das 'letzte' Element wird fraktional betrachtet kann jedoch die Kapazität nicht überschreiten

3/4

A3	A4	Σ	P2
7/8	7/7	14/15	5.5/6

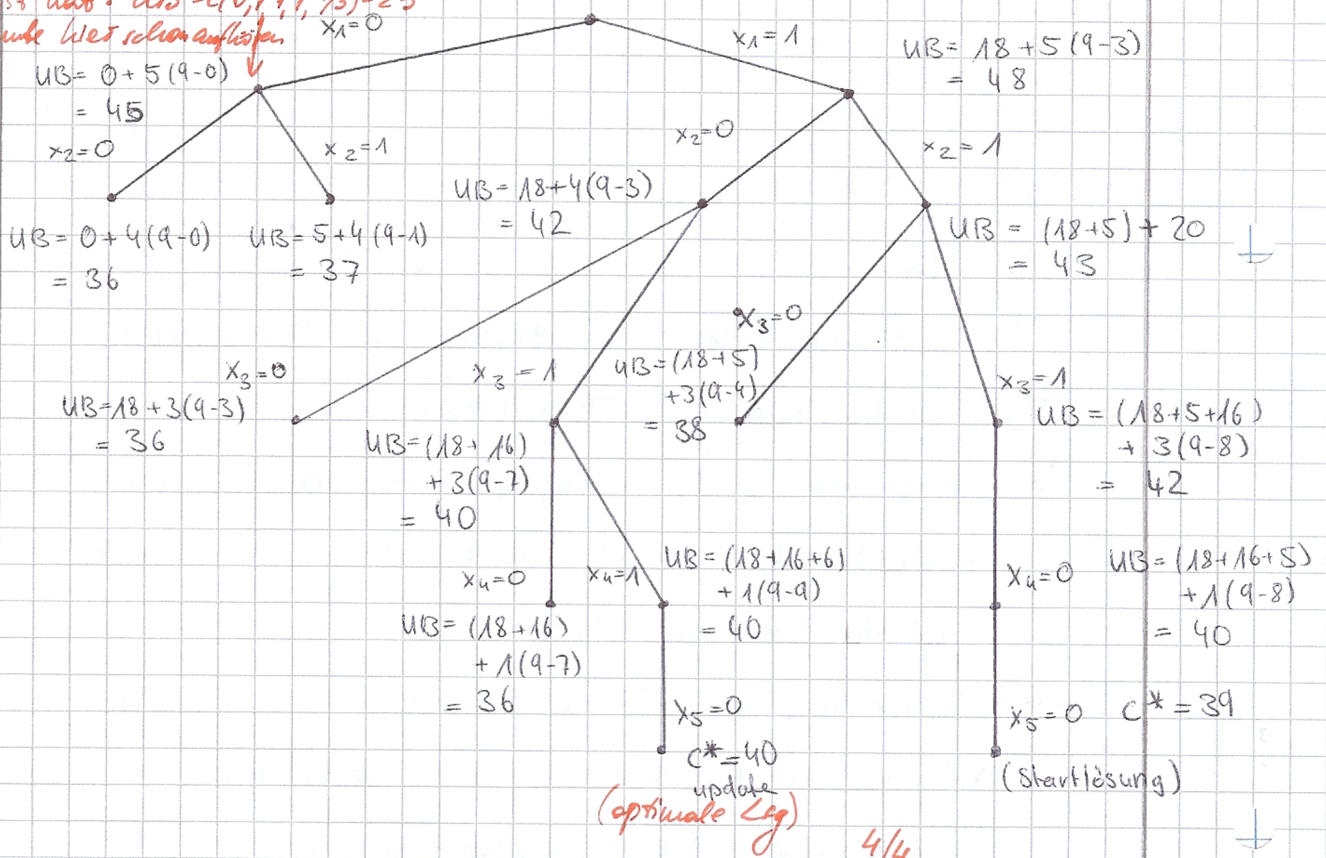
b) Sortierte Objekte

Gegenstand i	1	2	3	4	5
c_i	18	5	16	6	6
w_i	3	1	4	2	6
c_i/w_i	6	5	4	3	1

Startlösung $X = \{1,1,1,0,0\}$ mit $c^* = 39 = UB$

Wenn man ein frakt. Rucksackproblem hier gelöst hat: $UB = c(0,1,1,1,\frac{1}{3}) = 29$

↳ man könnte hier schon aufhören



4/4

4)

Variablen

x_1 = Fermat $\quad$ y_1 = ungerade Zahlen

x_2 = Abel $\quad$ y_2 = Kommutative Gruppen

x_3 = Cantor $\quad$ y_3 = Wurzeln ziehen

x_4 = Galois $\quad$ y_4 = Primzahlen

x_5 = Euler $\quad$ y_5 = Große Zahlen

Domänen: $\forall_i : x \in [1,..,5] \wedge y \in [1,..,5]$ ✓

Constraints:

$\forall_i \exists_j : x_i = y_j$ ✓

$\forall_{i,j}\ j \neq i : y_i \neq y_j \wedge x_i \neq x_j$ ✓

Reihenfolge dem Aufgabenblatt folgend:

1.) $y_1 \% 2 \neq 0$ ✓

2.) $x_1 \neq x_2 + 1 \wedge x_2 \neq x_1 + 1$ ✓

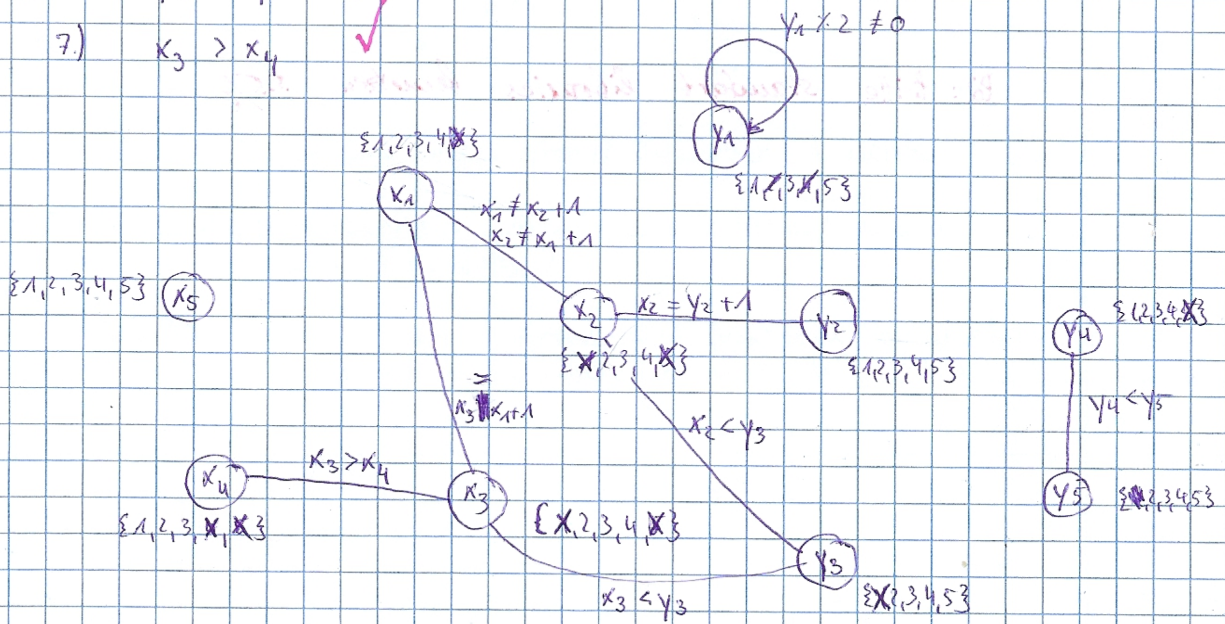
3) $x_3 = x_1 + 1$ ✓

4.) $x_2 = y_2 + 1$ ✓

5.) $x_2 < y_3 \wedge x_3 < y_3$ ✓

6.) $y_4 < y_5$ ✓

7.) $x_3 > x_4$ ✓



Algorithmus

$c^* := 0 \quad k := 0$

for $i = k+1$ to n do:

if $w_i \leq W$:

$x_i = 1$

$W = W - w_i$

$c^* += c_i x_i$

if $\sum_{i=1}^{n} c_i x_i > c^*$

$c^* = \sum_{i=k}^{n} c_i x_i$

end for

Stattlösung

$c^* = 0, \ k = 0$

for $i = k+1$ to n do

if $w_i \leq W$

$x_i = 1$

$W = W - w_i$

$c^* = c^* + x_i c_i$

endfor

Lösung zu Aufgabe 4

$x_1 = 2$	$y_1 = 1$
$x_2 = 4$	$y_2 = 3$
$x_3 = 3$	$y_3 = 5$
$x_4 = 1$	$y_4 = 2$
$x_5 = 5$	$y_5 = 4$

✓

Galois → Fermat → Cantor → Abel → Euler

ungerade Zahlen → Primzahlen → komm. Gruppen

→ große Zahlen → Wurzelziehen ✓

7/7

A2: Bitte standart libraries benutzen 5.5/6