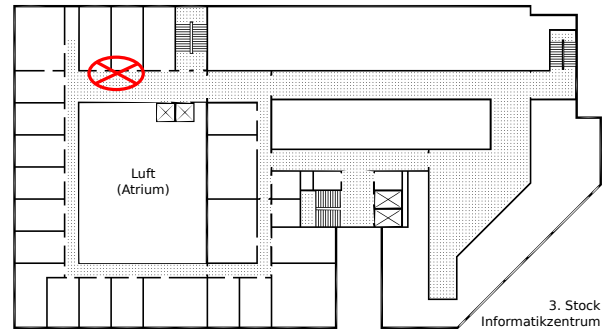


Hausaufgabenblatt 3

Abgabe der Lösungen bis zum 12.06.2026 um 13:15 Uhr im Hausaufgabenschrank bei Raum IZ 337 (siehe Skizze rechts). Es werden nur mit einem dokumentenechten Stift (kein Rot!) geschriebene Lösungen gewertet.

Bitte Namen, Matrikelnummer und Gruppennummer auf die Abgabe schreiben. Mehrere Blätter tackern!



Hausaufgabe 1 (Branch-and-Bound):

(8+3+3 Punkte)

In dieser Aufgabe betrachten wir den Branch-and-Bound-Algorithmus für MAXIMUM KNAPSACK aus der Vorlesung. Wir benutzen GREEDY_0 (siehe Blatt 1) als untere Schranke (P) und den (abgerundeten Wert des) Greedy-Algorithmus für FRACTIONAL KNAPSACK als obere Schranke (U). Im Folgenden nennen wir ein Item i *fixiert*, wenn der dazugehörigen Variable b_i durch Branch & Bound bereits ein Wert zugewiesen wurde.

- a) Wende den Branch-and-Bound-Algorithmus auf folgende Instanz für MAXIMUM KNAPSACK an.

i	1	2	3	4	
z_i	14	7	11	9	und $Z = 20$
p_i	15	7	10	8	

Beachte folgende Punkte:

- Benutze den Enumerationsbaum¹ auf Seite 4 aus Abbildung 1.
- Beschrifte die Kanten mit der Auswahl, die getroffen wurde und die Knoten mit den aktuell besten Schranken (obere (U) und untere (P)).
- Ist eine Auswahl unzulässig, beschrifte den jeweiligen Knoten mit *unzulässig*.
- Sollten Kanten nicht benutzt werden, streiche sie durch.
- Beschrifte zulässige Blätter mit dem erreichten Wert.
- Halte in einer Tabelle fest, welche Objekte eine neue, beste Lösung liefern.

¹Hier gibt es den Baum noch mal so zum Herunterladen.

- b) Zeige oder widerlege: Für jedes $n \geq 1$ gibt es eine Instanz mit n Objekten, bei der der Branch-and-Bound-Algorithmus keine rekursiven Aufrufe startet.
- c) Der Algorithmus aus der Vorlesung trifft Entscheidungen auf den einzelnen Variablen stets in der Reihenfolge, in der sie in der Instanz auftauchen. Insbesondere kommen auf jedem Pfad von der Wurzel des Entscheidungsbaums zu dessen Blättern die Variablen immer in der gleichen Reihenfolge vor.

Der Branch & Bound-Algorithmus bleibt korrekt, wenn man an jedem Suchbaumknoten nach irgendeinem Verfahren individuell eine noch nicht fixierte Variable b_i auswählt und hierfür als nächstes die Möglichkeiten $b_i = 0$ und $b_i = 1$ ausprobiert.

Zeige oder widerlege: Das Verfahren, nach dem wir die nächste fixierte Variable wählen, hat keinen Einfluss auf die Anzahl Knoten des Suchbaums, die wir besuchen. Gib also entweder eine Beispielinstantz an, bei der sich durch eine Änderung des Verfahrens zur Auswahl der nächsten fixierten Variablen die Anzahl besuchter Knoten des Suchbaums ändert, oder begründe, warum es eine solche Instanz nicht geben kann.

Hausaufgabe 2 (Linear/Integer Programming): (6 Punkte)

Viele Optimierungsprobleme lassen sich als Integer Program (IP) formulieren, die dann mit entsprechenden Solvern in der Praxis oft gut gelöst werden können. Intern kombinieren diese Solver ein Branch-and-Bound-Verfahren mit LP-Relaxierungen des Problems (welche effizient lösbar sind), um schließlich auf eine ganzzahlige Lösung zu kommen. In dieser Aufgabe probieren wir diese Technik an einem kleinen Beispiel aus.

Wir betrachten das folgende Integer Program ($x_i \in \mathbb{N}$).

$$\begin{aligned}
 &\text{maximiere} && 20x_1 + 25x_2 \\
 &\text{sodass} && 3x_1 + 4x_2 \leq 25 \\
 &&& x_1 \leq 2 \\
 &&& 2x_2 \leq 12 \\
 &\text{mit} && x_1, x_2 \in \mathbb{N}_0^+
 \end{aligned}$$

Löse das obige Integer Program mit Hilfe des Branch-and-Bound-Algorithmus. Löse dafür in jedem Knoten das relaxierte Linear Program ($x_i \in \mathbb{R}$). Wenn eine Variable x_i in der Lösung nicht ganzzahlig ist, branche und betrachte zunächst den Fall $x_i \leq \lfloor x_i \rfloor$ und danach $x_i \geq \lceil x_i \rceil$. Sind beide Variablen nicht ganzzahlig, branche über x_1 . Gib in jedem Knoten an, welche obere Schranke U und welche untere Schranke P gilt. Wir erhalten eine untere Schranke, wenn die Variablen der optimalen LP-Lösung ganzzahlig ist. Ist noch keine untere Schranke bekannt, gilt $P = 0$.

Ihr braucht die LPs nicht von Hand zu lösen. Nutzt stattdessen einen Solver wie den unter <https://online-optimizer.appspot.com>. Auf dieser Webseite könnt ihr das relaxierte Modell unter „Model“ wie folgt eintragen:

```

var x1 >= 0;
var x2 >= 0;

maximize z:      20*x1 + 25*x2;

subject to constraint1: 3*x1 + 4*x2 <= 25;
subject to constraint2:   x1          <=  2;
subject to constraint3:          2*x2 <= 12;

end;

```

Um das LP zu lösen, muss man dann unten links auf „Solve Model“ klicken. Unter „Solution“ → „Variables“ steht dann die Belegung der Variablen der optimalen Lösung. Unter „Model Overview“ ist dann der Wert der Lösung „Optimal objective value“

Um einen Branch (z.b. $x_2 \leq 4$) weiter zu betrachten, kann im Model ein neuer Constraint hinzugefügt werden:

```
subject to branch1: x2 <= 4;
```

Möchte man den anderen Branch (also $x_2 \geq 5$) testen, wird der Constraint ausgetauscht gegen:

```
subject to branch1: x2 >= 5;
```

Wenn es mit den gegebenen Constraints keine gültige Lösung mehr gibt, gibt der Solver ‘infeasible’ aus.

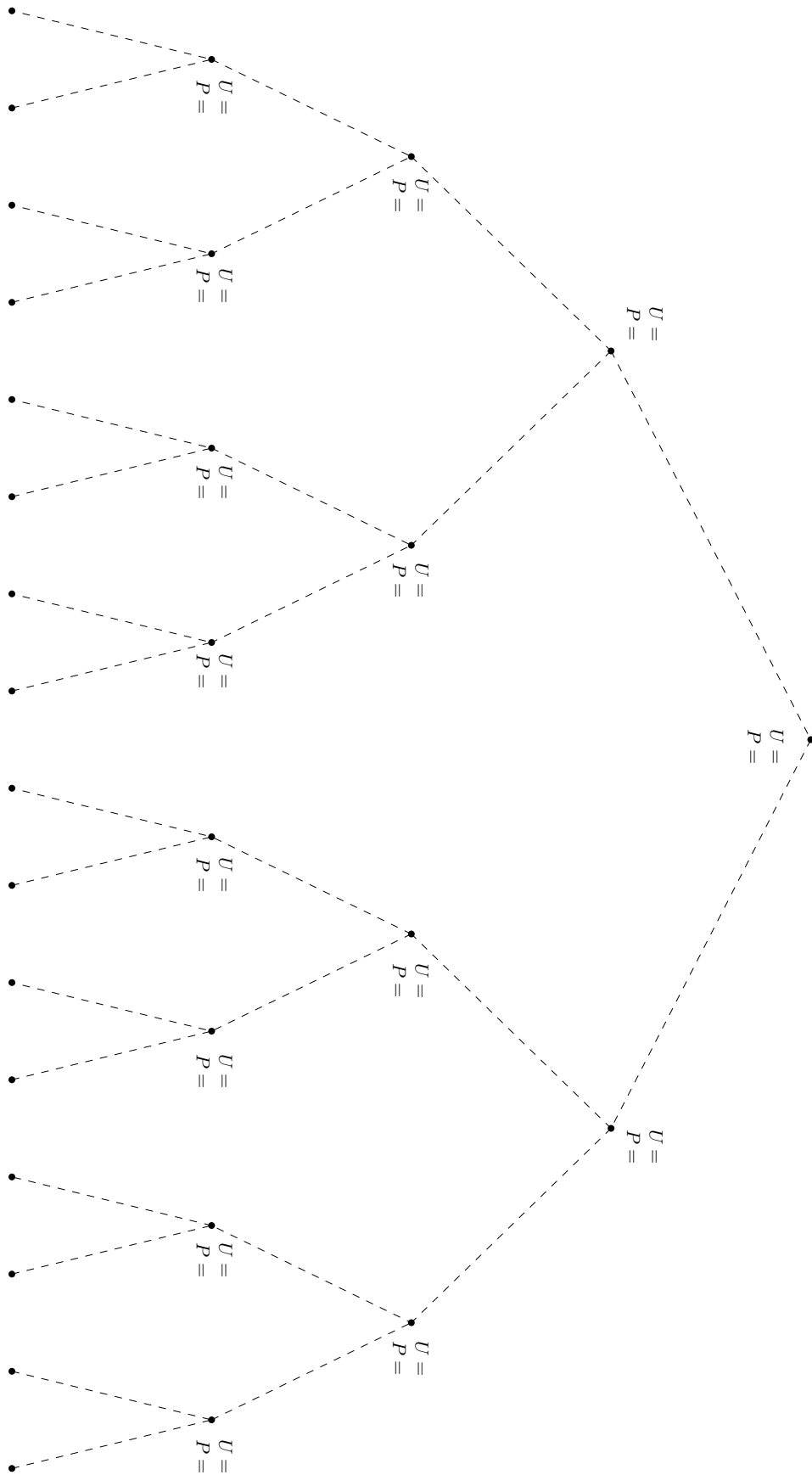


Abbildung 1: Ein Entscheidungsbaum.