

Lösungsstrategien für NP-schwere Probleme der Kombinatorischen Optimierung

— Übungsblatt 7 —

Walter Stieben
(4stieben@inf)

Tim Reipschläger
(4reipsch@inf)

Louis Kobras
(4kobras@inf)

Hauke Stieler
(4stieler@inf)

Abgabe am: 6. Juni 2016

Aufgabe 7.1

Algorithm 1 ApproxWightedHittingSet

```

1: procedure APPROXWIGHTEDHITTINGSET( $A, B$ )
2:   while  $B \neq \emptyset$  do
3:      $a := \text{null}$                                      // the element with the best quality
4:      $q_{\min} := \infty$ 
5:     for all  $a_i \in A$  do
6:        $n := \text{amountOfAppearances}(a_i)$ 
7:        $quality := \text{weight}(a_i)/n$ 
8:       if  $quality < q_{\min}$  then
9:          $q_{\min} \leftarrow quality$ 
10:         $a \leftarrow a_i$ 
11:      end if
12:    end for
13:     $H \leftarrow H \cup \{a\}$ 
14:     $B \leftarrow B \setminus \text{allSetsHitBy}(a)$ 
15:     $A \leftarrow A \setminus a$ 
16:  end while
17: end procedure

```

Beschreibung

Zu Beginn jedes Schleifendurchgangs (Zeile 3-15) wird das Element aus A bestimmt, welches die beste Qualität hat. Die Qualität beschreibt wie viel Gewicht pro getroffenem $B_j \in B$ aufgenommen wird. Je weniger Gewicht aufgenommen wird, desto besser das Resultat. Dadurch hat ein kleinerer Wert eine höhere (bessere) Qualität.

Bestimmt wird das Element a mit der besten Qualität in Zeile 5-12 in der alle verbleibenden $a_i \in A$ durchgegangen werden. Die Funktion `amountOfAppearances` berechnet dabei die Anzahl der getroffenen Mengen $B_j \in B$ durch das übergebene Element a_i .

Am Ende (Zeile 13-15) wird das gefundene Optimum für diesen Durchlauf (a) in das Hitting Set H aufgenommen. Zudem wird aus B jede Menge entfernt, die durch a getroffen wurde. Auch wird a aus A entfernt, sodass nun ein neues Optimum berechnet werden kann.

Terminierung

Der angegebene Algorithmus terminiert, da in jedem $B_i \in B$ nur Elemente aus A vorkommen. Da im worst-case jedes $a_i \in A$ einmal die beste Qualität hat, wird jedes B_i getroffen. Elemente mit gleicher Qualität werden nacheinander behandelt, sodass es auch dort keine Terminierungsprobleme geben kann.

Laufzeitanalyse

Auch wenn die Bedingung der **while**-Schleife $B \neq \emptyset$ lautet, so wird diese maximal $|A|$ mal ausgeführt, da bei jedem Durchlauf A um ein Element verkleinert wird. Ist $A = \emptyset$, so ist auch $B = \emptyset$ (s. Abschnitt Terminierung oben).

Die innere **for**-Schleife wird ebenfalls $|A|$ mal ausgeführt. In ihr wird **amountOfAppearances** aufgerufen, was die Anzahl der "Hits" ausgibt. Dabei wird in einer intuitiven Implementation jede Menge in B in jedes Element in dieser Menge durchgegangen. Jede Menge in B kann dabei maximal $|A|$ viele Elemente beherbergen, wodurch sich eine Laufzeit ergibt, die in $\mathcal{O}(|B| \cdot |A|)$ ist. Alle anderen Schritte haben eine konstante Laufzeit.

Am Ende (Zeile 13-15) wird a in H aufgenommen, was in konstanter Zeit machbar ist und a aus A gelöscht, was in linearer Zeit machbar ist (sofern man bei beiden von einer verketteten Liste ausgeht). Die Funktion **allSetsHitBy**, welche aufgerufen wird funktioniert in einer intuitiven Implementation genauso wie **amountOfAppearances**, nur wird hier ein anderes Ergebnis zurückgegeben. Die Laufzeit von **allSetsHitBy** ist somit ebenfalls in $\mathcal{O}(|B| \cdot |A|)$.

Insgesamt ergibt sich also eine in der Eingabe polynomielle Laufzeit in

$$\mathcal{O}(|A| \cdot (|A| \cdot (|B| \cdot |A|) + (|B| \cdot |A|))) = \mathcal{O}(|A|^3 \cdot |B| + |A|^2 \cdot |B|).$$

Beweis: ApproxWightedHittingSet ist ein b -Approximationsalgorithmus

ApproxWightedHittingSet ist ein b -Approximationsalgorithmus. Trivial.

;)
□

Aufgabe 7.2

a.) Lösung mit Dynamic Programming 1:

4	0	2	2	3	3	3	5	7	7	<u>8</u>
3	0	2	2	3	3	3	5	7	7	<u>8</u>
2	0	0	1	1	1	1	5	5	<u>6</u>	6
1	0	0	0	0	0	0	<u>5</u>	5	5	5
0	<u>0</u>	0	0	0	0	0	0	0	0	0
	0	1	2	3	4	5	6	7	8	9

Es werden die Items 1,2 und 3 genommen, die zusammen das Gewicht 9 und den Wert 8 haben.

b.) Lösung mit Dynamic Programming 2:

4	0	1	1	3	6	6	8	8	<u>8</u>	13	13	13
3	0	1	1	3	8	8	8	8	<u>8</u>	∞	∞	∞
2	0	2	7	7	7	7	<u>7</u>	∞	∞	∞	∞	∞
1	0	6	6	6	6	<u>6</u>	∞	∞	∞	∞	∞	∞
0	<u>0</u>	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
	0	1	2	3	4	5	6	7	8	9	10	11

Es werden wieder die Items 1,2 und 3 genommen, die zusammen das Gewicht 9 und den Wert 8 haben. Die Einträge mit dem Gewicht 13 müssen hier ausgelassen werden, da $13 > 9 = W$.