

Präsenzblatt 4

Dieses Blatt dient der persönlichen Vorbereitung. Es wird nicht abgegeben und geht nicht in die Bewertung ein. Die Besprechung der Aufgaben erfolgt in den kleinen Übungen am 13.07. und 15.07.22

Präsenzaufgabe:

Wir betrachten das folgende Problem.

Gegeben: Eine endliche Menge U , eine Familie \mathcal{F} von Teilmengen von U und eine Zahl $k \in \mathbb{N}$.

Gesucht: Ein *Set Cover* von (U, \mathcal{F}) der Größe höchstens k . Ein Set Cover ist eine Teilfamilie $F \subseteq \mathcal{F}$, die U überdeckt, d.h. für jedes Element $u \in U$ gibt es eine Menge $M \in F$ mit $u \in M$. Die Größe eines Set Covers ist die Anzahl an Mengen in F , d.h. $|F|$.

Wir nehmen an, dass jedes Element aus U in einer Menge aus \mathcal{F} vorkommt.

Als Beispiel betrachte $U := \{1, 2, 3, 4, 5, 6\}$ und $\mathcal{F} := \{\{1, 2\}, \{1, 4\}, \{3, 6\}, \{2, 3, 4\}, \{1, 2, 5\}, \{2, 3\}\}$, sowie $k = 3$. $F := \{\{1, 4\}, \{3, 6\}, \{1, 2, 5\}\}$ ist ein Set Cover von (U, \mathcal{F}) . Es kann schnell überprüft werden, dass es für $k = 2$ kein Set Cover gibt. Eine graphische Darstellung dieser Instanz ist in Abbildung 1 abgebildet.

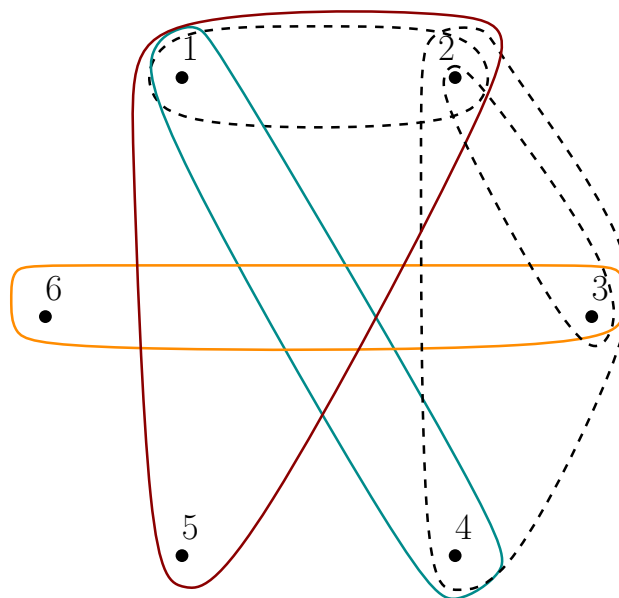


Abbildung 1: Beispiel einer Instanz von SET COVER. Punkte entsprechen den Elementen in U , Kreise entsprechen den Mengen in \mathcal{F} . Die farbige Auswahl entspricht einem Set Cover.

a) Zeige, dass Set Cover NP-schwer ist. (Hinweis: Nutze VERTEX COVER.)

Da SET COVER also NP-schwer ist, bietet es sich an, Approximationsalgorithmen zu betrachten, um das kleinste k zu finden. Der folgende Algorithmus (GREEDYSC) versucht ein möglichst kleines Set Cover zu bestimmen.

Algorithmus 1 Algorithmus GREEDYSC zum Finden eines Set Covers. In jeder Iteration wird diejenige Menge aufgenommen, die die meisten nicht überdeckten Elemente besitzt.

```
1: function GREEDYSC( $U, \mathcal{F}$ )
2:    $C := \emptyset$  ▷ Menge der bereits überdeckten Elemente
3:    $\bar{C} := U$  ▷ Menge der noch zu überdeckenden Elemente
4:    $SC := \emptyset$  ▷ Set Cover

5:   while  $C \neq U$  do
6:      $S := \operatorname{argmax}_{M \in \mathcal{F}} |M \cap \bar{C}|$  ▷ Menge mit den meisten nicht überdeckten Elementen

7:      $\alpha := 1/|S \cap \bar{C}|$ 
8:     For each  $s \in S \cap \bar{C}$  do  $\text{kosten}(s) := \alpha$ 

9:      $C := C \cup S$ 
10:     $\bar{C} := \bar{C} \setminus S$ 
11:     $SC := SC \cup \{S\}$ 

12:  return  $SC$ 
```

b) Wende GREEDYSC auf folgende Instanz an: $U := \{1, \dots, 10\}$, $\mathcal{F} := \{F_1, \dots, F_5\}$ mit $F_1 = \{1, 2, 3, 7, 9\}$, $F_2 = \{4, 5, 6, 8, 10\}$, $F_3 = \{1, 2, 3, 4, 5, 6\}$, $F_4 = \{7, 8\}$ und $F_5 = \{9, 10\}$.

Gib dabei nach jeder Iteration der while-Schleife S , α , C sowie \bar{C} an. Führe nach Ablauf des Algorithmus für alle $s \in U$ die $\text{kosten}(s)$ auf.

c) Zeige: In jeder Iteration von GREEDYSC gilt $\alpha \leq \text{OPT}/|\bar{C}|$.

(Hinweis: In jeder Iteration kann \bar{C} mit $\leq \text{OPT}$ Elementen überdeckt werden.)