Lösungsvorschläge – Blatt 8

Zürich, 15. April 2025

Lösung zu Aufgabe 9

- (a) Sei $C_{\max}$ eine Clique maximaler Grösse in G , $I_{\max}$ ein Independent-Set maximaler Grösse und sei (C, I) eine Split-Partition von G . Es kann höchstens ein Knoten von $C_{\max}$ in I sein, ansonsten wäre I nicht unabhängig. Das bedeutet, C muss mindestens $|C_{\max}| - 1$ Knoten von $C_{\max}$ enthalten. Ausserdem kann C höchstens einen zusätzlichen Knoten enthalten, ansonsten wäre die Maximalität von $|C_{\max}|$ verletzt. Analog dazu stellen wir fest, dass I mindestens $|I_{\max}| - 1$ Knoten von $I_{\max}$ enthält, sowie höchstens einen weiteren Knoten.

Der Graph G enthält also neben $C \cap C_{\max}$ und $I \cap I_{\max}$ höchstens zwei Knoten. Da $|C \cap C_{\max}| \geq |C_{\max}| - 1$, gibt es höchstens $|C_{\max}| + 1$ Möglichkeiten für die Wahl von $C \cap C_{\max}$. Analog gibt es höchstens $|I_{\max}| + 1$ Möglichkeiten, $I \cap I_{\max}$ zu wählen. Für jede dieser Aufteilungen gibt es höchstens zwei Möglichkeiten, die verbliebenen Knoten von G auf C und I aufzuteilen. Es gibt also höchstens $2 \cdot (|C_{\max}| + 1) \cdot (|I_{\max}| + 1) \in \mathcal{O}(|V|^2)$ Split-Partitionen von G .

Mit einer etwas detaillierteren Analyse können wir sogar zeigen, dass G höchstens $|V| + 1$ Split-Partitionen hat. Sei $C_{\max}$ eine Clique maximaler Grösse in G und sei (C, I) eine Split-Partition von G . Wie oben dargelegt, müssen alle Knoten von $C_{\max}$ bis auf möglicherweise eine Ausnahme in C sein. Sei $x \in C_{\max}$ eine solche Ausnahme, also $x \in C_{\max} \cap I$. Dann muss $N[x] \subseteq C_{\max}$ gelten: Es kann keine Nachbarn von x in I geben, ansonsten wäre I nicht unabhängig. Falls x einen Nachbarn y in $C \setminus C_{\max}$ besitzt, dann ist y auch Nachbar von allen Knoten in $C_{\max} \setminus \{x\}$, weil $C_{\max} \setminus \{x\} \subseteq C$ und C eine Clique ist. Das bedeutet aber, dass $C_{\max} \cup \{y\}$ eine Clique ist, im Widerspruch zur Maximalität von $C_{\max}$.

Falls es keine Knoten $x \in C_{\max}$ mit $N[x] \subseteq C_{\max}$ gibt, muss $(C_{\max}, V \setminus C_{\max})$ also die einzige Split-Partition sein. Falls es genau einen solchen Knoten x gibt, kann höchstens ein Knoten aus $V \setminus C_{\max}$ in C sein. Ansonsten wäre $|C| \geq |C_{\max} \setminus \{x\}| + 2 = |C_{\max}| + 1$, im Widerspruch zur Maximalität von $|C_{\max}|$. Das heisst, die einzigen Möglichkeiten für C sind $C_{\max}$, $C_{\max} \setminus \{x\}$ und $(C_{\max} \setminus \{x\}) \cup \{y\}$ für $y \in V \setminus C_{\max}$. Weil $C_{\max}$ mindestens einen Knoten enthält, ist $|V \setminus C_{\max}| \leq |V| - 1$, also gibt es höchstens $|V| + 1$ Split-Partitionen. Falls es $\ell \geq 2$ solcher Knoten $x_1, \dots, x_\ell$ gibt, kann höchstens ein Knoten x_i in I sein, weil sie alle benachbart sind. Das bedeutet auch, dass mindestens

ein weiterer Knoten x_j mit $j \neq i$ in C enthalten ist. Weil $N[x_j] \subseteq C_{\max}$, kann C also keine Knoten enthalten, die nicht in $C_{\max}$ sind. Die einzigen Möglichkeiten für C sind also $C_{\max}$ und $C_{\max} \setminus \{x_i\}$ für $1 \leq i \leq \ell$. Somit gibt es höchstens $|V| + 1$ Split-Partitionen.

Wir zeigen jetzt noch, dass wir in Polynomialzeit alle Split-Partitionen von G generieren können. Sei $V = \{v_1, \dots, v_n\}$. Wir werden nacheinander für jedes $1 \leq i \leq n$ alle Split-Partitionen von $G[\{v_1, \dots, v_i\}]$ generieren. Die Split-Partitionen von $G[\{v_1\}]$ sind offensichtlich genau $(\{v_1\}, \emptyset)$ und $(\emptyset, \{v_1\})$. Sei jetzt $1 \leq i < n$. Wenn wir aus einer Split-Partition von $G[\{v_1, \dots, v_{i+1}\}]$ den Knoten v_{i+1} entfernen, erhalten wir eine Split-Partition von $G[\{v_1, \dots, v_i\}]$. Wir können also alle Split-Partitionen (C, I) von $G[\{v_1, \dots, v_i\}]$ mit $i < n$ durchgehen und jeweils überprüfen, ob $C \cup \{v_{i+1}\}$ eine Clique ist und/oder $I \cup \{v_{i+1}\}$ ein Independent-Set ist. Damit generieren wir alle Split-Partitionen von $G[\{v_1, \dots, v_{i+1}\}]$. Weil wir gezeigt haben, dass es für jedes i höchstens $i < |V|$ solche Split-Partitionen gibt, können wir alle Split-Partitionen von $G = G[\{v_1, \dots, v_n\}]$ in Zeit $\mathcal{O}(|V|^3)$ berechnen.

- (b) Wir betrachten eine beliebige Split-Partition (C_1, I_1) von $H[V \setminus T]$. Es kann höchstens ein Knoten v_C aus C_1 in I_X und höchstens ein Knoten v_I aus I_1 in C_X liegen. Wir werden alle dieser $\mathcal{O}(|V|^2)$ Möglichkeiten durchgehen. Wir betrachten also $C^I = \emptyset$ oder $C^I = \{v_I\}$ für ein $v_I \in I_1$, sowie $I^C = \emptyset$ oder $I^C = \{v_C\}$ für ein $v_C \in C_1$. Wenn $C_0 \cup C^I$ keine Clique oder $I_0 \cup I^C$ kein Independent-Set ist, ignorieren wir diese Möglichkeit. Sei nun $\hat{C} = C_0 \cup C^I \cup (C_1 \setminus I^C)$ und $\hat{I} = I_0 \cup I^C \cup (I_1 \setminus C^I)$. Wenn es eine Partition (C_X, I_X) mit den entsprechenden Eigenschaften gibt, dann gilt $C_X \subseteq \hat{C}$ und $I_X \subseteq \hat{I}$ für mindestens eine Wahl von C^I und I^C . Weil C_X eine Clique sein muss, entfernen wir alle Knoten in $C_1 \setminus I^C$, die nicht mit einem Knoten in $C_0 \cup C^I$ verbunden sind. Ebenso entfernen wir alle Knoten in $I_1 \setminus C^I$, die mit einem Knoten in $I_0 \cup I^C$ verbunden sind. Bezeichnen wir mit $\hat{X}$ die Menge dieser Knoten. Wir finden nun entweder ein $\hat{X}$ mit $\hat{X} \cap T = \emptyset$ und $|\hat{X}| \leq r$, oder stellen fest, dass es keines gibt. Wenn wir aber ein solches $\hat{X}$ finden, ist es genau die Menge, die wir suchen. Die Partition $(\hat{C} \setminus \hat{X}, \hat{I} \setminus \hat{X})$ ist eine Split-Partition. So ist $\hat{C} \setminus \hat{X}$ sicher eine Clique: $C_0 \cup C^I$ ist eine Clique, $C_1 \setminus I^C$ ist eine Clique und jede Kante von $C_1 \setminus I^C$ nach $C_0 \cup C^I$ ist auch vorhanden, sonst hätten wir die entsprechenden Knoten in C_1 mit X entfernt. Aus demselben Grund ist auch $\hat{I} \setminus \hat{X}$ ein Independent Set. Wir setzen also $X = \hat{X}$. Für jedes C^I und I^C ist diese Analyse in $\mathcal{O}(|V|)$ durchführbar, der ganze Prozess also in $\mathcal{O}(|V|^3)$.

- (c) Stellen wir uns vor, wir hätten bereits eine solche Lösung S' und betrachten die Teilmenge $Y = S' \cap S$. Dann ist $S \setminus Y \subseteq V \setminus S'$. Weil $G[V \setminus S']$ ein Split-Graph ist, muss also auch $G[S \setminus Y]$ ein Split-Graph sein. Wir wissen, dass die Knoten aus Y Teil unserer Lösung sind. Wir betrachten nun für ein festes $Y \subseteq S$ mit $|Y| \leq k$ den Graphen $H = G[V \setminus Y]$. Auf diesem Graphen können wir mit $T = S \setminus Y$ genau die Methode aus (c) anwenden:

Einerseits ist $H[T] = H[S \setminus Y]$ ein Split-Graph, weil $G[S \setminus Y]$ ein Split-Graph ist. Andererseits ist $H[V(H) \setminus T] = G[(V \setminus Y) \setminus (S \setminus Y)] = G[V \setminus S]$ ebenfalls ein Split-Graph. Wir setzen $r = k - |Y|$. Für die Split-Partition (C_0, I_0) von $H[T]$ probieren wir wieder alle Möglichkeiten aus, davon gibt es nach Teilaufgabe (a) höchstens $\mathcal{O}(k^2)$. Der Algorithmus aus Teilaufgabe (b) liefert uns dann eine Teilmenge X , sodass $H[V \setminus X] = G[(V \setminus Y) \setminus X] = G[V \setminus (X \cup Y)]$ ein Split-Graph ist. Ausserdem gilt $|X \cup Y| \leq |X| + |Y| \leq (k - |Y|) + |Y| = k$, also haben wir eine zulässige Lösung

$$S' = X \cup Y.$$

Im schlimmsten Fall müssen wir $\binom{k+1}{0} + \binom{k+1}{1} + \dots + \binom{k+1}{k} \in \mathcal{O}(2^k)$ Teilmengen Y ausprobieren. Für jede dieser Instanzen benötigen wir nach Teilaufgabe (b) polynomiell viel Zeit.

- (d) Sei $V = \{v_1, \dots, v_n\}$. Dann berechnen wir nach und nach eine Lösung für die Instanzen $G[\{v_1, \dots, v_i\}]$. Für $G_1 = G[\{v_1\}]$ ist offensichtlich $S_1 = \emptyset$ eine zulässige Lösung. Wenn wir eine Lösung S_i^* mit $|S_i^*| \leq k$ für den Graphen G_i berechnet haben, bildet offensichtlich $S_{i+1} = S_i^* \cup \{v_{i+1}\}$ eine Lösung für den Graphen G_{i+1} mit $|S_{i+1}| \leq k + 1$. Falls $|S_{i+1}| \leq k$, setzen wir $S_{i+1}^* = S_{i+1}$. Ansonsten berechnen wir mit Teilaufgabe (c) eine Lösung S_{i+1}^* für G_{i+1} mit $|S_{i+1}^*| \leq k$ oder erkennen, dass es keine solche Lösung für G_{i+1} und damit auch keine für G gibt. Die Lösung für die Instanz (G, k) bildet schliesslich S_n^* . Damit erhalten wir einen Algorithmus mit Laufzeit in $\mathcal{O}(2^k \cdot k^2 \cdot p(|V|))$ für eine polynomielle Funktion $p(|V|)$.