Prof. Dr. M. Grohe E. Fluck, A. Riazsadri, J. Feith

Übungsblatt 10 mit Lösungen

Abgabetermin: Mittwoch, der 18. Januar 2023 um 14:30

Aufgabe 3 (Quadratisches Zuordnungsproblem) **6 Punkte**

Wie üblich bezeichnen wir mit der Notation $[n]$ die Menge $\{1, \ldots, n\}$.

QUADRATICASSIGNMENT

Eingabe: Nicht-negative Integer Kosten $c_{ij}, i, j \in [n]$, Distanzen $d_{kl}, k, l \in [m]$ (mit $m \geq n$) und eine Schranke $b \in \mathbb{Z}^+$.

Frage: Gibt es eine injektive Abbildung $\sigma : [n] \to [m]$, so dass

$$\sum_{i=1}^{n} \sum_{\substack{j=1 \\ j \neq i}}^{n} c_{ij} d_{\sigma(i)\sigma(j)} \leq b?$$

Eine mögliche Interpretation des Problems ist es eine Menge $[n]$ von Fabriken so an den m möglichen Standorten anzusiedeln, dass die gesamten Transportkosten minimal sind. Dabei gibt c_{ij} die Menge Material an, das zwischen den Fabriken i und j transportiert werden muss und d_{kl} die Distanz zwischen dem k-ten und l-ten Standort.
Dieses Problem ist insbesondere beim Entwerfen von Platinen auch von praktischer Relevanz.
Zeigen Sie, dass das Problem QUADRATICASSIGNMENT NP-vollständig ist.
Hinweis: Wir empfehlen eine Reduktion vom Problem TSP.

Lösung:

In NP Wir zeigen durch die Angabe eines Zertifikats und Verifizieres, dass QUADRATICASSIGNMENT in NP liegt.

Zertikat: Sei σ eine gültige Lösung, dann ist das Zertifikat:

$$\text{bin}(\sigma(1))\,\text{bin}(\sigma(2))\ldots\text{bin}(\sigma(n)),$$

wobei wir eine Binärkodierung mit führenden Nullen nutzen, sodass alle Binärkodierten Zahlen die gleiche Länge haben und wir auf Trennzeichen verzichten können.

Verifizierer: Prüfe zuerst ob das Zertifikat die richtige Länge hat und ob es eine injektive Funktion kodiert. Anschließend berechne den Zielfunktionswert $\sum_{i=1}^{n} \sum_{\substack{j=1 \\ j \neq i}}^{n} c_{ij} d_{\sigma(i)\sigma(j)}$ und vergleiche mit b.

Laufzeit: Das Zertifikat hat die Länge $O(n \log(m))$.

Um die Länge des Zertifikats und die Injektivität der Funktion zu prüfen, muss man für maximal n mal das Zertifikat durchsuchen (einmal für jedes $\text{bin}(\sigma(i))$) und beim ersten Durchlauf prüft man direkt auf die richtige Länge. Auch der Zielfunktionswert lässt sich in polynomieller Zeit berechnen, na man $O(n^2)$ viele Summanden berechnen muss, welche sich wiederum durch konstant viele Lookups in der Eingabe bzw dem Zertifikat und einer

Multiplikation berechnen lassen. Zuletzt muss man den Vergleich machen. Alles in allem hat der Verifizierer polynomielle Laufzeit.

NP-Schwer Reduktion vom Problem TSP. Die Idee der Reduktion ist es mit Hilfe der Gewichte c_{ij} eine Reihenfolge (hier $1, 2, \dots, n, 1$) über die Elemente zu definieren und die Gewichte d_{ij} entsprechen den Abständen im TSP. Die injektive Abbildung σ kodiert dann die Zuordnung der Knoten in die Reihenfolge, also die Permutation der Knoten.

Gegeben eine symmetrische Gewichtungsfunktion $c\colon [N] \times [N] \to \mathbb{N}$.

Wir berechnen daraus eine Instanz von QuadraticAssignment. Dabei wählen wir $n = m = N$. Die Kosten definieren wir als

$$c_{ij} = \begin{cases} 1 & j = i + 1 \bmod n \\ 0 & sonst \end{cases}$$

und entsprechend die Distanzen als

$$d_{kl} = c(k, l).$$

Die Schranke b behalten wir bei.

Hinweis: die Rollen von c und d sind beliebig austauschbar.

Korrektheit

Angenommen es gibt eine Tour der Länge höchstens b und sei π die Permutaion auf $[N]$, die diese Tour angibt. Wir definieren $\sigma(i) = \pi(i)$, für alle $i \in [N]$. Dann gilt

$$\begin{aligned} \sum_{i=1}^{n} \sum_{\substack{j=1 \\ j \neq i}}^{m} c_{ij} d_{\sigma(i)\sigma(j)} &= c_{n1} d_{\sigma(n)\sigma(1)} + \sum_{i=1}^{n-1} c_{ii+1} d_{\sigma(i)\sigma(i+1)} \\ &= c(\pi(n), \pi(1)) + \sum_{i=1}^{n-1} c(\pi(i), \pi(i+1)) \leq b. \end{aligned}$$

Andersherum sei σ eine Lösung der Instanz von QuadraticAssignment. Eine injektive Abbildung zwischen zwei gleich großen, endlichen Mengen ist automatisch auch bijektiv. Daher ist σ eine Permutation auf $[N]$ und wir definieren $\pi(i) = \sigma(i)$. Dann gilt

$$\begin{aligned} c(\pi(n), \pi(1)) + \sum_{i=1}^{n-1} c(\pi(i), \pi(i+1)) &= c_{n1} d_{\sigma(n)\sigma(1)} + \sum_{i=1}^{n-1} c_{ii+1} d_{\sigma(i)\sigma(i+1)} \\ &= \sum_{i=1}^{n} \sum_{\substack{j=1 \\ j \neq i}}^{m} c_{ij} d_{\sigma(i)\sigma(j)} \leq b. \end{aligned}$$

Laufzeit

Die Laufzeit der Konstruktion ist offensichtlich polynomiell.

Aufgabe 4 (Set Cover) 6 Punkte

Wir betrachten das folgende Entscheidungsproblem.

> SetCover
>
> **Eingabe:** Eine endliche Menge U, eine Menge $\mathcal{S} \subseteq \text{Pot}(U) := \{V \mid V \subseteq U\}$ und $k \in \mathbb{N}$.
>
> **Frage:** Gibt es eine Menge $\mathcal{C} \subseteq \mathcal{S}$, sodass $|\mathcal{C}| \leq k$ und jedes Element von U durch $\mathcal{C}$ abgedeckt ist, d.h.
>
> $$U = \bigcup_{V \in \mathcal{C}} V\,.$$

Zeigen Sie, dass das SetCover-Problem NP-vollständig ist.

Lösung:

In NP Wir zeigen durch Angabe eines Zertifikats und Verifizieres, dass SetCover in NP liegt.

Zertikat: Sei $\mathcal{C}$ ein Set Cover. Ein 0-1-String y der Länge $|\mathcal{S}|$, wobei eine 1 an der i-ten Stelle bedeutet, dass die i-te Menge aus $\mathcal{S}$ im Set Cover $\mathcal{C}$ enthalten ist.

Verifizierer: Prüfe zunächst, ob der String y die Länge $|\mathcal{S}|$ und höchstens k Einsen enthält. Dann initialisiere einen String $z = 0^{|U|}$. Laufe über den String y und führe für jedes i mit $y_i = 1$ folgende Subroutine aus: Bestimme die i-te Menge M aus $\mathcal{S}$ und setze für jedes $u \in M$ das zugehörige Bit von z auf 1. Am Ende prüft der Algorithmus, ob der String z noch eine Null enthält. Falls ja, dann verwirft der Verifizierer, sonst akzeptiert er.

Laufzeit: Sei n die Eingabelänge und m die Länge von y. Die Kardinalität von $\mathcal{S}$ lässt sich in Zeit $\mathcal{O}(n \log n)$ bestimmen und das Zählen der Eisen in y geht in Zeit $\mathcal{O}(m \log m)$. Das Durchführen einer Subroutine kann in polynomieller Zeit durchgeführt werden (es müssen $|M|$ Bits gesetzt werden und $|M| \leq n$). Da die Subroutine höchstens m mal durchgeführt läuft auch der zweite Schritt in polynomieller Zeit. Für den letzte Schritt genügt es einmal über den String z zu laufen, das geht in $\mathcal{O}(n)$ Schritten. Insgesamt erhalten wir also polynomielle Laufzeit.

NP-Schwer Dazu reduzieren wir VertexCover auf SetCover. Da VertexCover NP-schwer ist nach Aufgabe 11.2 erhalten wir somit die NP-Schwere von SetCover. Die Idee dazu ist, als Universum U die Menge aller Kanten zu nutzen und jeder Knoten entspricht einer Teilmenge in $\mathcal{S}$. Diese Teilmenge enthält genau die Kanten die inzident sind zu dem Knoten.

Sei (G, k), wobei $G = (V, E)$ ein Graph und $k \in \mathbb{N}$, die Eingabe von VertexCover. Wir setzen $U = E$, $\mathcal{S} = \{E_v \mid v \in V\}$ mit $E_v = \{(v, w) \in E\}$ und übernehmen den Wert k.

Offensichtlich ist die beschriebene Reduktion in polynomieller Zeit berechenbar.

Sei C ein Vertex Cover von G der Größe höchstens k. Setze $\mathcal{C} = \{\{(v,w) \in E\} \mid v \in C\}$. Dann ist $|\mathcal{C}| \leq |C| \leq k$. Außerdem ist jedes $e = (v,w) \in U = E$ durch $\mathcal{C}$ abgedeckt, da $v \in C$ oder $w \in C$.
Umgekehrt sei $\mathcal{C} \subseteq \mathcal{S}$ ein Set Cover von U der Größe höchstens k. Sei $C \subseteq V$, sodass für jedes $M \in \mathcal{C}$ ein $v \in V$ existiert mit $M = E_v$. Da $|\mathcal{C}| \leq k$ können wir C so wählen, dass C höchstens Größe k hat. Sei nun $e \in E$, dann gibt es ein $v \in C$, sodass $E_v \in \mathcal{C}$ und $e \in E_v$. Dann ist aber e inzident zu v. Also ist C ein Vertex Cover.

Aufgabe 5 (Abschlusseigenschaften) **3(2+1) Punkte**

Wir definieren $\mathsf{coNP} = \{\overline{L} \mid L \in \mathsf{NP}\}$, die Klasse der Komplemente aller Sprachen, welche in NP liegen.

a) Beweisen Sie, dass die Komplexitätsklasse NP unter den Operationen Vereinigung und Durchschnitt abgeschlossen ist. Folgern Sie daraus, dass auch coNP unter Operationen Vereinigung und Durchschnitt abgeschlossen ist.

b) Es seien $L_1, L_2 \subseteq \Sigma^*$ zwei Sprachen mit $L_1, L_2 \in \mathsf{NP} \cap \mathsf{coNP}$. Beweisen Sie, dass dann die Sprache

$$L_1 \oplus L_2 \;:=\; \{x \in \Sigma^* \mid x \in L_1 \setminus L_2 \text{ oder } x \in L_2 \setminus L_1\}$$

ebenfalls in $\mathsf{NP} \cap \mathsf{coNP}$ liegt.

Lösung:

a) Zuerst zeigen wir, dass NP unter Durchschnitt abgeschlossen ist. Seien $A, B \in \mathsf{NP}$. Dann gibt es NTMs M_A und M_B, die die Sprachen A und B entscheiden. Wir geben eine NTM M_C an, welche die Sprache $C := A \cap B$ entscheidet. Die NTM M_C simuliert zuerst M_A auf der Eingabe. Falls die Simulation akzeptiert, dann wird M_B auf der Eingabe simuliert. Falls auch die zweite Simulation akzeptiert, dann akzeptiert M_C. Die Laufzeit von M_C ist daher polynomiell in der Summe der Laufzeiten von M_A und M_B beschränkt. Somit haben wir gezeigt, dass NP unter Durchschnitt abgeschlossen ist.

Nun zeigen, wir dass NP unter Vereinigung abgeschlossen ist. Seien $A, B \in \mathsf{NP}$. Dann gibt es NTMs M_A und M_B, die die Sprachen A und B entscheiden. Wir geben eine NTM M_C an, welche die Sprache $C := A \cup B$ entscheidet. Die NTM M_C simuliert die NTMs M_A und M_B parallel (dies bedeutet, dass M_C abwechselnd einen Schritt von M_A und M_B simuliert). Die NTM M_C akzeptiert, sobald eine der NTMs M_A, M_B akzeptiert. Der Lauf des kürzesten Berechnungspfades ist daher polynomiell in der Summe der Laufzeiten von M_A und M_B beschränkt. Somit haben wir gezeigt, dass NP unter Vereinigung abgeschlossen ist.

Nun folgern wir den Abschluss unter Vereinigung für coNP. Seien $A, B \in \mathsf{coNP}$. Wir zeigen $A \cup B \in \mathsf{coNP}$. Per Definition gilt $\bar{A}, \bar{B} \in \mathsf{NP}$. Daher gilt $C := \bar{A} \cap \bar{B} \in \mathsf{NP}$ (Abschluss unter Vereinigung von NP). Somit ist $\bar{C} = A \cup B \in \mathsf{coNP}$, was zu zeigen war.

Indem wir die Rollen von Vereinigung und Durchschnitt vertauschen, lässt sich zeigen, dass coNP unter Durchschnitt abgeschlossen ist. Seien hierfür $A, B \in \mathsf{coNP}$. Wir zeigen $A \cap B \in \mathsf{coNP}$. Per Definition gilt $\bar{A}, \bar{B} \in \mathsf{NP}$. Daher gilt $C := \bar{A} \cup \bar{B} \in \mathsf{NP}$ (Abschluss unter Durchschnitt von NP). Somit ist $\bar{C} = A \cap B \in \mathsf{coNP}$, was zu zeigen war.

b) Da NP und coNP unter Vereinigung und Durchschnitt abgeschlossen sind, ist auch die Komplexitätsklasse $\mathcal{C} := \mathsf{NP} \cap \mathsf{coNP}$ unter Vereinigung und Durchschnitt abgeschlossen. Außerdem ist $\mathcal{C}$ unter Komplement abgeschlossen, da $\mathcal{C} = \mathsf{NP} \cap \mathsf{coNP} = \mathsf{coNP} \cap \mathsf{NP} = \mathrm{co}\,\mathcal{C}$.

Es gilt $L_1 \oplus L_2 = (L_1 \cap \bar{L}_2) \cup (L_2 \cap \bar{L}_1)$ mit $L_1, L_2 \in \mathcal{C}$. Somit ist $\bar{L}_2, \bar{L}_1 \in \mathcal{C}$ (Abschluss unter Komplement). Somit ist $(L_1 \cap \bar{L}_2), (L_2 \cap \bar{L}_1) \in \mathcal{C}$ (Abschluss unter Durchschnitt). Somit ist $L_1 \oplus L_2 = (L_1 \cap \bar{L}_2) \cup (L_2 \cap \bar{L}_1) \in \mathcal{C} = \mathsf{NP} \cap \mathsf{coNP}$ (Abschluss unter Vereinigung).