

Lösungsvorschläge – Blatt 3

Zürich, 14. März 2024

Lösung zu Aufgabe 3

- (a) Sei (A, B) der vom Algorithmus konstruierte Schnitt. Wir definieren für jede Kante $e \in E$ die Zufallsvariable

$$X_e = \begin{cases} 1 & e \in E(A, B) \\ 0 & e \notin E(A, B) \end{cases}$$

Weil die Endknoten jeder Kante unabhängig voneinander mit Wahrscheinlichkeit $1/2$ in A sind, gilt $\text{Prob}(X_e = 1) = \text{Prob}(X_e = 0) = 1/2$ für alle Kanten $e \in E$. Damit gilt auch $\mathbb{E}[X_e] = \text{Prob}(X_e = 1) = 1/2$.

Wir können die Anzahl Kanten in $E(A, B)$ berechnen als $\sum_{e \in E} X_e$. Es gilt also

$$\mathbb{E}[|E(A, B)|] = \mathbb{E}[\sum_{e \in E} X_e] = \sum_{e \in E} \mathbb{E}[X_e] = \sum_{e \in E} \frac{1}{2} = \frac{|E|}{2}.$$

Weil der optimale Schnitt nicht mehr Kanten enthalten kann als der Graph selbst, ist die Approximationsgüte des Algorithmus im Erwartungswert also höchstens 2.

- (b) Sei $e = \{v_i, v_j\}$ eine Kante in G . Wir verwenden jetzt ein Resultat aus der Analyse des Algorithmus aus der Vorlesung. Die Wahrscheinlichkeit, dass die Kante e im Schnitt ist, ist $\cos^{-1}(\vec{x}_i \cdot \vec{x}_j)/\pi$. Es gilt aber $\vec{x}_i \cdot \vec{x}_j = \vec{e}_i \cdot \vec{e}_j = 0$. Damit ist

$$\text{Prob}(e \in E(A, B)) = \cos^{-1}(0)/\pi = (\pi/2)/\pi = 1/2.$$

Mit derselben Methode aus Teilaufgabe (a) ergibt sich damit eine Approximationsgüte von 2.

- (c) Wir könnten wieder $x_i = 1 \Leftrightarrow v_i \in A$ definieren. Dann müssen wir irgendwie unterscheiden, ob eine Kante (v_i, v_j) von A nach B führt oder umgekehrt. Eine Möglichkeit wäre, den Ausdruck $(1 + x_i) \cdot (1 - x_j)$ zu betrachten. Dieser Ausdruck ist dann 0, wenn $x_i = x_j = -1$, $x_i = x_j = 1$ oder $x_i = -1$, $x_j = 1$. Der einzige Fall,

in dem er nicht verschwindet, ist $x_i = 1$, $x_j = -1$, also wenn tatsächlich $v_i \in A$ und $v_j \in B$. Wir könnten also versuchen, den Ausdruck

$$\frac{1}{4} \cdot \sum_{(i,j) \in E} (1 + x_i) \cdot (1 - x_j) = \frac{1}{4} \cdot \sum_{(v_i, v_j) \in E} (1 + x_i - x_j - x_i \cdot x_j)$$

zu maximieren. Dabei bekommen wir aber Probleme mit der *SDP*-Relaxierung, weil wir lineare Terme haben. Wenn wir die Variablen x_i durch Vektoren $\vec{x}_i$ ersetzen, erhalten wir den Ausdruck $1 + \vec{x}_i - \vec{x}_j - \vec{x}_i \cdot \vec{x}_j$, der nicht definiert ist.

Um dieses Problem zu umgehen, können wir die Variable x_0 verwenden. Diese Variable soll definieren, wofür die anderen Variablen stehen. Es soll gelten $v_i \in A \Leftrightarrow x_i = x_0$. Jetzt können wir überprüfen, dass

$$1 + x_0 \cdot x_i - x_0 \cdot x_j - x_i \cdot x_j = \begin{cases} 4 & x_i = x_0, x_i \neq x_j \\ 0 & \text{sonst} \end{cases}$$

für jede Kante $(v_i, v_j) \in E$, also genau dann, wenn die Kante im Schnitt (A, B) ist. Somit können wir das quadratische Programm wie folgt definieren:

Maximiere

$$\frac{1}{4} \cdot \sum_{(v_i, v_j) \in E} (1 + x_0 \cdot x_i - x_0 \cdot x_j - x_i \cdot x_j)$$

unter der Bedingung

$$x_i \in \{-1, 1\} \text{ für } i = 0, \dots, n.$$

Jetzt können wir die *SDP*-Relaxierung anwenden und erhalten das folgende Vektorprogramm:

Maximiere

$$\frac{1}{4} \cdot \sum_{(v_i, v_j) \in E} (1 + \vec{x}_0 \cdot \vec{x}_i - \vec{x}_0 \cdot \vec{x}_j - \vec{x}_i \cdot \vec{x}_j)$$

unter der Bedingung

$$\begin{aligned} \vec{x}_i &\in \mathbb{R}^n \text{ für } i = 0, \dots, n \\ \vec{x}_i \cdot \vec{x}_i &= 1 \text{ für } i = 0, \dots, n. \end{aligned}$$