ETH zürich
Departement Informatik

Algorithmik für schwere Probleme

Prof. Dr. Dennis Komm
Dr. Hans-Joachim Böckenhauer
Dr. Richard Kralovič
`https://courses.algodi.inf.ethz.ch/schwereprob-2025`

Lösungsvorschläge – Blatt 5

Zürich, 25. März 2025

Lösung zu Aufgabe 6

(a) Offensichtlich ist ein Turnier, das ein gerichtetes Dreieck enthält, nicht azyklisch, weil es einen gerichteten Kreis aus drei Knoten enthält. Sei nun $G = (V, E)$ ein Turnier, das einen gerichteten Kreis enthält. Sei $(v_1, \ldots, v_r)$ ein gerichteter Kreis minimaler Länge in G. Falls $r = 3$, enthält G ein Dreieck. Falls $r \geq 4$ betrachten wir die Kante zwischen v_2 und v_r. Weil G ein Turnier ist, ist entweder $(v_2, v_r) \in E$ oder $(v_r, v_2) \in E$. Falls $(v_2, v_r) \in E$, ist (v_1, v_2, v_r) ein Dreieck in G. Falls $(v_r, v_2) \in E$, ist $(v_2, \ldots, v_r)$ ein gerichteter Kreis der Länge $r - 1$ in G. Beides steht im Widerspruch dazu, dass $r \geq 4$ minimal ist. Dieser Fall kann also nicht auftreten.

(b) Sei A ein Feedback Arc Set in G mit $|A| \leq k$. Wenn wir eine Kante $f \neq e$ umdrehen, können wir höchstens eines der $k + 1$ Dreiecke, in denen e enthalten ist, entfernen. Die Kante e muss also sicher in A enthalten sein. Damit ist A also von der Form $A' \cup \{e\}$, wobei A' ein Feedback Arc Set für $G(\bar{e})$ ist. Umgekehrt erhalten für eine beliebige Kante $e \in E$ und jedes Feedback Arc Set A' für $G(\bar{e})$ ein Feedback Arc Set $A' \cup \{e\}$ für G.

(c) Es gilt $V_x^- \cup V_x^+ = V \backslash \{x\}$. Weil x nicht Teil eines Dreiecks ist, gibt es in G keine Kanten von V_x^+ nach V_x^-. Wir definieren $E_x^+ = E(G[V_x^+])$ als die Menge der Kanten zwischen Knoten von V_x^+. Analog definieren wir E_x^-. Sei $A \subseteq E$ ein Feedback Arc Set für G mit $|A| \leq k$. Sei $A^+ = A \cap E_x^+$ und $A^- = A \cap E_x^-$. Dann ist $A^+ \cup A^- \subseteq A$ ein Feedback Arc Set für $G - x$: Wenn die Kanten von $A^+ \cup A^-$ umgedreht werden, gibt es weiterhin keine Kanten von V_x^+ nach V_x^-, das heisst, es gibt im entstehenden Graphen keine Dreiecke, die Knoten aus V_x^+ *und* V_x^- enthalten. Alle Dreiecke, die nur aus Knoten von V_x^+ bestehen, mussten in G durch umdrehen einer Kante in $A \cap E_x^+ = A^+$ entfernt werden, sind also auch in $G - x$ nicht mehr vorhanden. Dasselbe gilt für V_x^- und A^-.

Sei umgekehrt A' ein Feedback Arc Set für $G - x$. Wir betrachten $A^+ = A' \cap E_x^+$ und $A^- = A' \cap E_x^-$. Dann ist $A^+ \cup A^- \subseteq A'$ ein Feedback Arc Set für G: $A^+ \cup A^-$ enthält keine Kanten von V_x^+ nach V_x^- und keine Kanten, die zu x inzident sind. Somit entstehen in G durch das Umdrehen dieser Kanten keine neuen Dreiecke, die

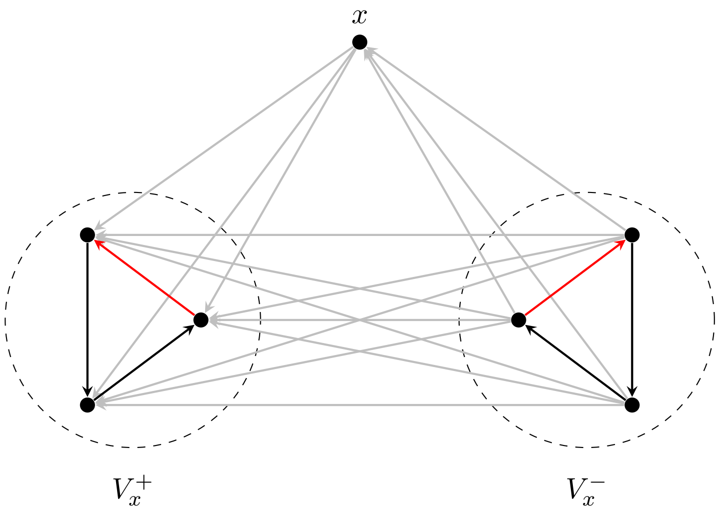

Abbildung 1: Dreiecke in G können nur innerhalb von V_x^+ und V_X^- existieren. Durch das Umdrehen der roten Kanten werden diese Dreiecke entfernt und es können auch keine weiteren Dreiecke in G entstehen.

x enthalten oder Knoten aus V_x^+ *und* V_x^- enthalten. Alle anderen Dreiecke in G bestehen entweder nur aus Knoten von V_x^+ oder nur aus Knoten von V_x^-, werden also durch $A^+ \cup A^-$ entfernt.

(d) Sei $(G, k) \in$ FAST, sodass G keine Kanten wie in (b) und keine Knoten wie in (c) enthält. Wir zeigen, dass G höchstens $k \cdot (k+2)$ Knoten hat. Sei A ein Feedback Arc Set für G mit $|A| \leq k$. Weil die Regel aus (b) nicht mehr anwendebar ist, kann jede Kante $e \in A$ Teil von höchstens k Dreiecken sein, die insgesamt höchstens $k+2$ Knoten enthalten (die zwei Knoten von e, sowie k weitere). Weitere Dreiecke kann es nicht geben, weil A ein Feedback Arc Set ist. Weil (c) nicht mehr anwendbar ist, muss jeder Knoten in G Teil eines Dreiecks sein, also kann G also höchstens $|A| \cdot (k+2) = k \cdot (k+2)$ Knoten enthalten.

Damit ist klar, wie wir den Kern berechnen können. Wir überprüfen jeweils, ob (G, k) eine der Bedingungen von (b) oder (c) erfüllt. Wenn diese beiden Regeln nicht mehr anwendbar sind, betrachten wir die resultierende Instanz (G', k'). Falls $|V'| > k \cdot (k+2)$, geben wir eine triviale Instanz $I \notin$ FAST zurück. Schliesslich müssen wir noch überprüfen, dass wir den Kern in polynomieller Zeit in n und k berechnen können. Da wir in jedem solchen Schritt entweder k oder n reduzieren, können wir beide Regeln höchstens $\max(k, n) \leq (k+n)$ mal anwenden. Sowohl (b) als auch (c) lassen sich in Zeit $\mathcal{O}(n^3)$ überprüfen. Somit lässt sich der Kern in Zeit $\mathcal{O}((k+n) \cdot n^3)$ berechnen.