Viertes Übungsblatt

Ausgabe: 17. Dezember 2014
Abgabe: Keine, Besprechung am 7. Januar 2015

1 Feedback Arc Set

In der Vorlesung wurden die beiden folgenden Probleme Minimum Feedback Arc Set und Minimum Feedback Set vorgestellt.

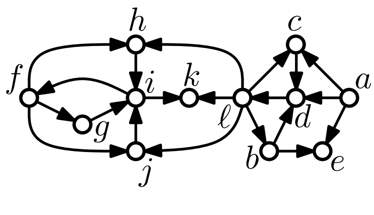
Problem 1.1 (Minimum Feedback Arc Set). *Sei $D = (V, A)$ ein gerichteter Graph. Bestimme eine Menge $A_f \subset A$ minimaler Kardinalität, so dass $D_f = (V, A \setminus A_f)$ azyklisch ist.*

Problem 1.2 (Minimum Feedback Set). *Sei $D = (V, A)$ ein gerichteter Graph. Bestimme eine Menge $A_r \subset A$ minimaler Kardinalität, so dass $D_r = (V, A \setminus A_r \cup \mathrm{rev}(A_r))$ azyklisch ist. Dabei bezeichne $\mathrm{rev}(A)$ die Kantenmenge, die man erhält wenn die Richtung jeder Kante der Menge A invertiert wird.*

Wir haben gesehen, dass jedes Feedback Set ein Feedback Arc Set ist, umgekehrt gilt dies im Allgemeinen jedoch nicht. Für minimale Mengen gilt die Äquivalenz: Zeigen Sie, dass die Lösungsmengen der beiden Probleme 1.1 und 1.2 identisch sind.

2 Lagenlayout

Führen Sie die einzelnen Schritte des Sugiyama-Frameworks zur Generierung von Lagenlayouts exemplarisch für den nebenstehenden Graphen aus. Entfernen Sie dazu gerichtete Kreise indem Sie für eine möglichst kleine Menge an Kanten die Richtung umkehren, finden Sie eine Lagenzuordnung minimaler Höhe bei maximaler Breite 4 und ordnen sie die Knoten innerhalb der Lagen so an, dass die Anzahl an Kreuzungen minimal ist.
Hinweis: Bestimmen Sie dazu in jedem Schritt manuell eine optimale Lösung des jeweiligen Teilproblems.



3 Fehlstände Zählen

(a) Sei $\pi : \{1, \dots, n\} \to \{1, \dots, n\}$ eine Permutation. Ein Paar (i, j) mit $1 \leq i < j \leq n$ heißt Inversion, wenn $\pi(i) > \pi(j)$. Entwerfen Sie einen Algorithmus, der die Anzahl der Inversionen einer Permutation von n Elementen in $O(n \log n)$ Zeit berechnet.
Hinweis: Denken Sie an Sortieralgorithmen wie Mergesort.

bitte umblättern

(b) Gegeben sei ein einfacher, bipartiter Graph $G = (V, E)$, dessen Knoten gemäß der Bipartition auf zwei parallele Geraden verteilt sind. Die Knoten seien disjunkt und die Kanten geradlinig gezeichnet. Entwerfen Sie einen Algorithmus, der die Anzahl der Kreuzungen in einer Laufzeit $O(|E| \log |V|)$ bestimmt. Begründen Sie, warum es nicht möglich ist, in dieser worst-case-Laufzeit alle Kreuzungen (d.h. die Paare betroffener Kanten) *auszugeben*.

4 Kreuzungen bei Lagenlayouts

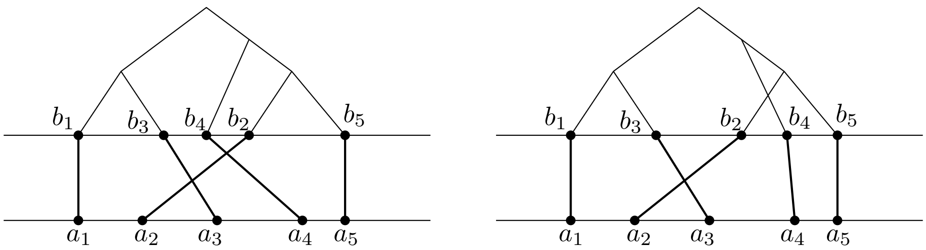
Zeigen Sie, dass die Baryzenter-Heuristik zur einseitigen Kreuzungsreduktion die optimale Lösung liefert, falls diese keine Kreuzung enthält.

5 Tanglegram Layout

Das einseitige Kreuzungsminimierungsproblem in einem „Zweilagengraph“ ist bereits NP-schwer. Betrachte das folgende verwandte Problem:

Problem 5.1 (One Tree Crossing Minimization (OTCM))**.** *Gegeben sei ein perfektes Matching M auf $2n$ Knoten $A = \{a_1, \ldots, a_n\}$ und $B = \{b_1, \ldots, b_n\}$, mit Matchingkanten $E = \{(a_i, b_i) \mid i = 1, \ldots, n\}$. Bestimme für eine feste Permutation π_1 der Menge A eine Permutation π_2 der Menge B, die die Anzahl der Kreuzungen von M minimiert, wobei A und B geordnet nach π_1 und π_2 auf zwei horizontalen Geraden platziert sind. Dabei muss π_2* kompatibel *zu einem Binärbaum T mit Blattmenge B sein.*

Die Permutation π_2 heißt dabei *kompatibel* zu T, wenn eine planare Zeichnung von T in der Halbebene $y \geq 0$ existiert, die die Blätter in der Reihenfolge π_2 auf der x-Achse platziert und alle inneren Knoten im Bereich $y > 0$.



kompatible Permutation π_2 Permutation π_2 nicht kompatibel

(a) Zeigen Sie, dass das Problem 5.1 in polynomieller Zeit gelöst werden kann. Welchen asymptotischen Zeitaufwand hat Ihr Algorithmus?

Hinweis: Verwenden Sie z.B. dynamische Programmierung.

(b) Angenommen beide Permutationen π_1 und π_2 sind variabel und jeweils durch Kompatibilität mit zwei Binärbäumen T_1 und T_2 auf den Blättern A bzw. B eingeschränkt. Zeigen Sie, dass in polynomieller Zeit entschieden werden kann, ob das Matching kreuzungsfrei gezeichnet werden kann.