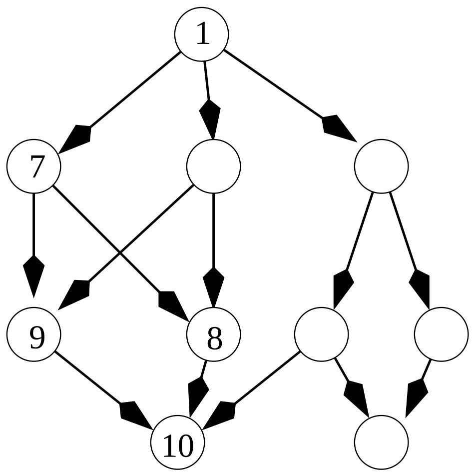
1 Pfade in azyklischen Graphen

Sei wieder ein gerichteter Graph mit Kantengewichten gegeben, der diesmal aber keine Kreise enthält, also azyklisch ist. Für solche Graphen lassen sich kürzeste Pfade einfacher berechnen als in beliebigen gerichteten Graphen. Wir wollen das single-source-Problem für diesen Spezialfall effizient lösen.

Wir nehmen an, dass der Startknoten v der einzige Knoten in G mit Eingangsgrad 0 (d.h. ohne eingehende Kante) ist und dass alle anderen Knoten von v aus erreichbar sind. (Sonst könnten wir einfach den Teilgraph aller von v aus erreichbaren Knoten betrachten.)

Eine hilfreiche Tatsache über azyklische gerichtete Graphen G ist, dass man eine Nummerierung bzw. Reihenfolge v_1, v_2, ..., v_n der Knoten von G finden kann, so dass alle Kanten des Graphen von einem Knoten v_i zu einem Knoten v_j mit $j > i$ laufen. Eine solche Reihenfolge nennt man auch *topologische Sortierung*. Sie lässt sich mit verschiedenen Methoden effizient in linearer Zeit bestimmen, etwa mit einer modifizierten Tiefensuche, bei der die Nummern in absteigender Reihenfolge am Ende der *visit*-Funktion vergeben werden. In einer topologischen Sortierung unter den obigen Annahmen muss der Startknoten v die Nummer 1 erhalten, da er der einzige Knoten ohne eingehende Kanten ist.

Die folgende Abbildung zeigt ein Beispiel eines azyklischen Graphen, bei dem die Nummern in den Knoten eine gültige topologische Sortierung darstellen:



Um die kürzesten Pfade in G zu berechnen, betrachten wir die Knoten von G in der Reihenfolge der topologischen Sortierung. Sei der aktuell bearbeitete Knoten v_i und seien die Abstände und kürzesten Pfade für $v_1, \ldots, v_{i-1}$ bereits berechnet. Da alle in v_i hineinlaufenden Kanten an einem der bereits erledigten Knoten beginnen, erfüllt der Abstand des aktuellen Knotens v_i von v die folgende Gleichung:

$$\texttt{dist}[v_i] = \min_{\substack{j=1,\ldots,i-1 \\ (v_j,v_i)\in E}} \{\texttt{dist}[v_j] + c(v_j, v_i)\}$$

Wenn der aktuelle Knoten v_i also d eingehende Kanten hat, so kann unter Zuhilfenahme von d bereits berechneten Werten der Wert $\texttt{dist}[v_i]$ bestimmt werden. Analog lassen sich nebenbei auch die `from`-Werte berechnen.

Mit diesen Hinweisen sollte es nicht schwer sein, ein Programm zu implementieren, das in Zeit $O(|V|+|E|)$ eine topologische Sortierung bestimmt und anschließend in Zeit $O(|V|+|E|)$ die kürzesten Pfade von v zu allen anderen Knoten.

Ebenso ist nicht schwer zu sehen, dass auf dieselbe Art und Weise auch die *längsten* Pfade von v zu allen anderen Knoten berechnet werden können, wenn man in der obigen Formel das Maximum statt des Minimums bildet. Die Berechnung längster Pfade kann also in azyklischen Graphen in linearer Zeit erfolgen, obwohl dieses Problem in beliebigen Graphen $\mathcal{NP}$-hart ist.

Literatur

[1] Volker Turau. *Algorithmische Graphentheorie.* Addison-Wesley, Bonn, 1996. S. 250–254.

2 Färbung planarer Graphen

2.1 Problemstellung und Ziel

In diesem Abschnitt befassen wir uns mit dem Färben von planaren Graphen. Wir betrachten die folgenden zwei Probleme:

- Können die Länder eine Landkarte mit nur 4 Farben so gefärbt werden, dass benachbarte Länder unterschiedliche Farben erhalten?
- Wie müssen die Frequenzen in Mobilfunknetzen den Sendemasten zugeteilt werden, dass möglichst wenige Frequenzen benötigt werden?

Eine Färbung eines ungerichteten Graphen $G = (V, E)$ ordnet jedem Knoten v (Kante e, Fläche f) aus V $(E,\ F)$ eine natürliche Zahl $c(v) \in IN$ $(c(e),\ c(f))$ zu, die auch Farbe genannt wird. Eine Färbung heißt zulässig, falls je zwei benachbarten Knoten (Kanten, Flächen) eine unterschiedliche Farbe zugewiesen wird. Man unterscheidet zwischen Knoten-, Kanten- oder Flächen-Färbung. Eine Färbung, die k verschiedene Farben verwendet, heißt k-Färbung. $\gamma(G) = \min\{k \mid G \text{ ist } k\text{-färbbar}\}$ heißt chromatische Zahl.

Im Allgemeinen ist das Problem, ob ein Graph k-färbbar ist, *NP*-schwer.

Definition 1 (Planarer Graph) *Ein Graph G ist planar, wenn*

- *$v \in V$ in der Ebene ($\mathbb{R}^2$) eingebettet werden können*
- *so dass sich die Kanten $e \in E$ nur in den Knoten kreuzen.*

Wir betrachten im Folgenden die Knoten-Färbung eines planaren Graphen G.

Satz 2 (Satz von Euler) *Seien n, e und f die Anzahl der Knoten, Kanten und Flächen eines planaren Graphen G. Dann gilt*

$$n - e + f = 2$$

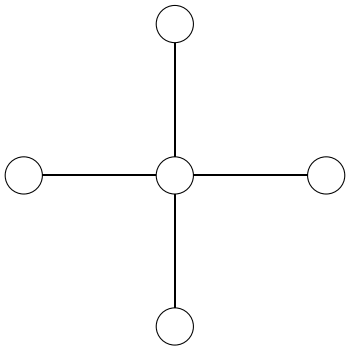
Beweis: Beweis durch Induktion über n.
Für $n = 1$ besteht der Graph G nur aus einer Menge von Schlingen. Falls $e = 0$, so existiert genau eine Fläche. Für $e > 0$ teilt jede Schleife eine Fläche in zwei Flächen, und somit ist die Induktionsannahme für $n = 1$ und $e \geq 0$ erfüllt.

Sei $n > 1$. Falls G zusammenhängend ist, können wir eine Kante finden, die keine Schleife ist. Anschließend können wir diese Kante "zusammenziehen" und wir erhalten einen Graphen G' mit n' Knoten, e' Kanten und f' Flächen. Durch das "Zusammenziehen" ändert sich nicht die Anzahl der Flächen, die Anzahl der Knoten und Kanten verringert sich aber um 1. Eingesetzt in die Hypoythese: $n - e + f = n' + 1 - (e' + 1) + f = 2$ $\square$

Der Satz von Euler ist nur gültig in zusammenhängenden Graphen. Die allgemeine Formel für einen planaren Graph G mit k Zusammenhangskomponenten lautet:

$$n - e + f = k + 1$$

Jeder Graph G mit maximalem Grad d kann mit $d + 1$ Farben gefärbt werden.



Satz 3 (Heawood) *Jeder planare Graph ist 5-färbbar.*

Beweis: $|V| \leq 5$: trivial.
Sei $G = (V, E)$ ein minimaler nicht 5-färbbarer planarer Graph. Sei $v \in V$ ein Knoten mit $deg(v) \geq 5$. Durch die Wahl von G ist der Graph $G - v$ 5-färbbar. Sei $f : V \backslash \{v\} \rightarrow \{1, ...5\}$ eine 5-Färbung von $G - v$. Da G nicht 5-färbbar ist, ist jeder Nachbar $N(v)$ mit einer anderen Farbe gefärbt. Wir färben die Nachbarn $n_1, ..., n_5$ von v im Uhrzeigersinn mit den Farben $c_1, ..., c_5$.

Sei $G_{i,j}$ der Teilgraph von G, der alle Knoten mit den Farben c_i und c_j enthält. In diesem Graphen können wir die beiden Farben jeder Komponente austauschen, und wir erhalten eine neue 5-Färbung von $G - v$. Die Komponente, die Knoten v_i enthält, muss auch Knoten v_j enthalten, ansonsten können wir die Komponente, die v_j enthält so umfärben, dass $v_j \in N(v)$ Farbe $c[v_i]$ erhält und der Graph G somit 5-färbbar ist.

Sei $P_{i,j}$ der Pfad in $G_{i,j}$ von Knoten v_i nach v_j. Betrachte nun den Kreis C, der durch den Pfad $P_{1,3}$ und Konten v definiert ist. Dieser Kreis trennt die Knoten v_2 und v_4. Aus diesem Grund muss der Pfad $P_{2,4}$ den Pfad $P_{1,3}$ kreuzen. Da G ein planarer Graph ist, ist eine Kreuzung nur in einem Knoten $v \in V$ zulässig. Da die Knoten aus $P_{1,3}$ die Farben 1 oder 3 und $P_{2,4}$ die Farben 2 oder 4 besitzen, kann eine Kreuzung der Kanten an einem Knoten nicht erfolgen. $\square$

Satz 4 (4-Farben-Satz) *Jeder planare Graph ist 4-färbbar.*

Anmerkung: Erst 1977 konnten Ken Appel und Wolfgang Haken einen Beweis finden. Der Beweis reduzierte die Anzahl der problematischen Fälle von Unendlich auf 1936 (eine spätere Version sogar 1476), die durch einen Computer einzeln geprüft wurden. 1996 konnten Neil Robertson, Daniel Sanders, Paul Seymour und Robin Thomas einen modifizierten Beweis finden, der die Fälle auf 633 reduzierte. Auch diese mussten per Computer geprüft werden.

2.2 Schema des Algorithmus

Betrachten wir den in Abb. 1 angegebenen Greedy-Ansatz:

func Greedy-Färbung((V, E))

$\forall v \in V$ do
    färbe v mit der nächsten freien Farbe
od

Abbildung 1: Greedy-Färbung 1

Klar ist, dass dieser Algorithmus eine gültige Färbung des Graphen G bestimmt, mit einer Laufzeit von $O(|V|+|E|)$. Diese Färbung kann aber beliebig schlecht werden. Siehe dazu Abb. 2. Obwohl nur 2 Farben nötig sind, verwendet der Algorithmus 6 Farben.

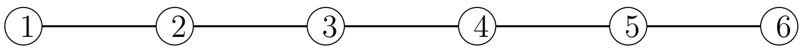


Abbildung 2: Beispiel

Der Algorithmus in Abb. 1 kann durch eine einfache Modifizierung verbessert werden. Betrachte dazu den Algorithmus in Abb. 3

func Greedy-Färbung$((V, E))$

$\forall v \in V$ do
    $c[v] = \min\{k \in IN \mid k \neq c[w] \quad \forall e = \{v, w\}$
od

Abbildung 3: Greedy-Färbung 2

Der Algorithmus in Abb. 3 benötigt im Allgemeinen weniger Farben als der Algorithmus in Abb. 1, bei gleicher Laufzeit $O(|V| + |E|)$. Für das Beispiel in Abb. 2 benötigt dieser Algorithmus nur 2 Farben. Aber auch hier kann die Färbung beliebig schlecht werden.

Für jeden Graphen $G = (V, E)$ gibt es eine Anordnung σ der Knoten $v \in V$, so dass der Greedy-Algorithmus in Abb. 3 eine optimale Färbung bestimmt.
Beweis: Seien $c_1, ..., c_{\gamma(G)}$ die Farben des optimal gefärbten Graphen G. Sei L_{c_i} die Liste der Knoten mit Farbe c_i Wähle σ wie folgt: $\sigma = L_{c_1} \circ ... \circ L_{c_{\gamma(G)}}$. Auf dieser Anordnung der Knoten liefert der Greedy-Algorithmus 3 eine optimale Färbung. □

Leider können wir eine optimale Anordung der Knoten aus V nicht im Voraus bestimmen. In vielen Fällen reicht es aus, eine „gute“ Färbung des Graphen G zu bestimmen. Dafür ist es ausreichend, eine „gute“ Anordnung der Knoten zu finden. Betrachtet man einen zu färbenden Knoten $v \in V$, so sind alle Farben $c[n_i]$ der Nachbarknoten $N(v)$ „verboten“. Das bedeutet, dass für Knoten mit hohem Grad viele Farben „verboten“ sind.

Daraus lässt sich der Algorithmus 4 ableiten: Dieser Algorithmus bestimmt im Allgemeinen eine „gute“ Färbung eines Graphen G in Laufzeit $O(|V| \log |V| + |E|)$

func Greedy-Färbung$((V, E))$

Ordne die Knoten $v \in V$ absteigend nach $deg(v) \rightarrow \sigma[.]$

$for\, i = 1\, to\, |\sigma|$ do
    $c[\sigma[i]] = \min\{k \in IN \mid k \neq c[n] \quad \forall e = \{\sigma[i], n\}$
od

Abbildung 4: Greedy-Färbung 3

Literatur

[1] K. Appel and W. Haken, *Every Planar Map is Four Colorable, Part I: Discharging.* Illinois J. Math., 21, 429-490, 1977.

[2] L. Euler, *Demonstratio Nonullarum Insignium Proprietatum Quibus Solida Hedris Planis Inculsa Sunt Praedita.* Novi Comm. Acad. Sci. Imp. Petrol, 4, 140-160, 1758.

[3] P.J. Heawood, *Map-Color Theorem.* Q. J. Math., 24, 332-339, 1890.