ETH Zürich
Institut für Theoretische Informatik
Prof. Dr. Angelika Steger, Dr. Johannes Lengler
Übungsleitung: Florian Meier

HS 2019

Algorithmen und Komplexität
Lösungsvorschlag zu Übungsblatt 5

Lösungsvorschlag zu Aufgabe 1

Wir modellieren das Problem mit einem Zustandsgraphen und suchen dann einen kürzesten Weg vom Anfangszustand zum Endzustand. Die Knoten des Graphen entsprechen den möglichen Zuständen, und die Kanten den möglichen Übergängen von einem Zustand in den anderen (was jeweils einer Flussüberquerung entspricht). Die Kanten sind ungerichtet, da jeder Übergang in beide Richtungen möglich ist.

Es sei $S = \{M, W, Z, K\}$, wobei M, W, Z und K jeweils den Mann, den Wolf, die Ziege und den Kohl bezeichnen. Jeden Zustand identifizieren wir mit einer Teilmenge X von S, wobei X die Menge der auf der Ausgangsseite Anwesenden beschreibt. Die Anzahl der Zustände ist deshalb gleich der Kardinalität der Potenzmenge von S, d.h. gleich $2^4 = 16$.

Einen Zustand nennen wir *zulässig*, falls weder der Wolf und die Ziege noch die Ziege und der Kohl unbewacht auf einer Seite allein sind. Man sieht sofort, dass genau die 6 Zustände $\{W, Z, K\}, \{M\}, \{W, Z\}, \{M, K\}, \{Z, K\}, \{M, W\}$ nicht zulässig sind. Die verbleibenden 10 Zustände sind in einem bipartiten Graph angeordnet, da der Mann in jedem Schritt die Seite wechselt. Der resultierende Zustandsgraph G ist in Abb. 1 gezeigt.

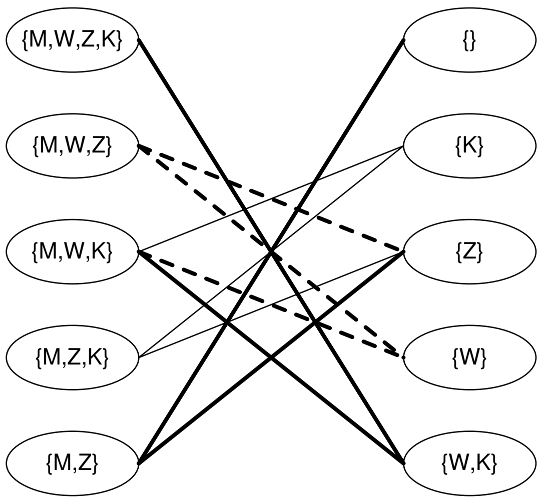

Abbildung 1: Der Zustandsgraph zum Wolf-Ziege-Kohl-Problem. Die fetten (durchgezogenen und gestrichelten) Kanten bilden einen kürzesten Weg vom Anfangs- zum Endzustand

Man sieht leicht, dass ein kürzester Weg in G vom Anfangszustand $\{M, W, Z, K\}$ zum Endzustand $\{\}$ die Länge 7 besitzt. Man sieht auch schön, dass das Problem symmetrisch bezüglich Vertauschung von Wolf und Kohl ist: Wenn die gestrichelten Kanten durch die nicht benutzten Kanten ersetzt werden, ergibt sich wieder ein kürzester Pfad.

Lösungsvorschlag zu Aufgabe 2

Da die Übertragung entlang der einzelnen Kanten unabhängig voneinander funktioniert, ist die Wahrscheinlichkeit, dass die Übertragung auf einem gegebenen Pfad funktioniert, das Produkt der Erfolgswahrscheinlichkeiten aller Kanten des Pfades.

Wir definieren auf G die modifizierte Gewichtsfunktion

$$h : E \to [0, \infty), \quad h(e) = -\log(w(e)).$$

Da alle Wahrscheinlichkeiten $0 \leq w(e) \leq 1$ erfüllen, sind in der Tat alle Kantengewichte $h(e)$ positiv. Wegen der Definition von h und der Eigenschaft des Logarithmus ($\log ab = \log a + \log b$) ist ein Pfad mit maximaler Übertragungswahrscheinlichkeit genau ein kürzester Pfad unter der Gewichtsfunktion h. Wir können also einfach den Algorithmus von Dijkstra mit den Gewichten h verwenden, um den zuverlässigsten Pfad zu finden.

Alternativ kann man auch direkt mit der Kantengewichtung w arbeiten und eine Modifikation des Dijkstra-Algorithmus verwenden. Man sucht dann einen *längsten* Pfad in G, wobei die Länge eines Pfades als das *Produkt* seiner Kantenlängen definiert ist. Das algorithmische Vorgehen von Dijkstra lässt sich analog realisieren und auch die Korrektheit lässt sich ähnlich wie beim Algorithmus von Dijkstra beweisen. Statt des Minimums ist dabei natürlich in jeder Iteration ein Maximum zu suchen.

Lösungsvorschlag zu Aufgabe 3

(a) Wir zeigen per Induktion, dass nach i Iterationen der For-Schleife folgende Invariante gilt: Falls ein $s - v$ Weg mit i Kanten existiert, dann ist $dist[v]$ die Länge des kürzesten $s - v$ Weges mit maximal i Kanten.

Induktionsanfang: Für $i = 0$ ist die Aussage wahr, da s der einzige Knoten ist, der mit 0 Kanten von s aus erreicht werden kann.

Induktionsschritt ($i \to i + 1$): Betrachte einen kürzesten s-v-Weg $(s, u_1, \ldots, u_k, v)$ mit maximal $i + 1$ Kanten. Dann ist $(s, u_1, \ldots, u_k)$ ein kürzester s-u_k- Weg mit maximal i Kanten (falls es einen kürzeren gäbe, könnte man durch anhängen von v einen kürzeren s-v-Weg erzeugen). Nach Induktionsannahme ist $dist[u_k]$ nach der i-ten Iteration gleich der Länge eines kürzesten s-u_k-Weges mit maximal i Kanten. Nach dem Betrachten der Kante (u_k, v) während der der $(i + 1)$-ten Iteration gilt $dist_{new}[v] \leq dist[u_k] + \ell(u_k, v)$. Deshalb ist $dist_{new}[v]$ kleiner gleich der Länge des kürzesten s-v-Weges mit maximal $i + 1$ Kanten. Andererseits ist es klar, dass ein s-v-Weg der Länge $dist_{new}[v]$ mit maximal $i + 1$ Kanten existiert: Sei u der Vorgänger von v, dann gibt es per Induktions Annahme einen Weg der Länge $dist[u]$ mit maximal i Kanten von s nach u, an welchen man v anhängen und somit einen Weg der Länge $dist[u] + \ell(u, v) = dist_{new}[v]$ erhält. Es folgt, dass $dist_{new}[v]$ gleich der Länge des kürzesten s-v-Weges mit maximal $i + 1$ Kanten ist.

Falls kein negativer gerichteter Kreis existiert, gilt dass ein kürzester s-v-Weg mit maximal $|V| - 1$ Kanten ein kürzester s-v-Pfad ist. Somit zeigt obiger Induktionsbeweis, dass in diesem Fall die Längen der kürzesten Pfade ausgegeben werden.

Falls ein negativer gerichteter Kreis $u_1, \ldots, u_k$ existiert, so gilt $0 > \sum_{i=1}^{k} \ell(u_i, u_{i+1})$ (wobei wir zur Vereinfachung der Notation $v_{k+1} = v_1$ definieren). Dann gilt $\sum_{i=1}^{k} dist[v_{i+1}] > \sum_{i=1}^{k} dist[v_i] + \sum_{i=1}^{k} \ell(u_i, u_{i+1})$, woraus folgt dass für mindestens einen Knoten im Kreis $dist[v_{i+1}] > dist[v_i] + \ell(u_i, u_{i+1})$ gilt und somit 'Error' ausgegeben wird.

(b) Die Laufzeit des Bellman-Ford Algorithmus beträgt $\Theta(|V| \cdot |E|)$. Für zusammenhängende Graphen gilt $|E| = \Omega(|V|)$ und $|E| = \mathcal{O}(|V|^2)$. Deshalb befindet sich seine Laufzeit zwischen $\Theta(|V|^3)$ von Floyd-Warshall und $\Theta(|V|^2)$ von Dijkstra. Beachte allerdings, dass Floyd-Warshall alle u-v-Distanzen berechnet während Dijkstra und Bellman-Ford nur die s-v-Distanzen für einen gegebenen Startknoten s berechnen.

Der Algorithmus von Dijkstra kann nur für Netzwerke mit positiver Gewichtsfunktion verwendet werden. Die Algorithmen von Bellman-Ford und modifizierter Floyd-Warshall (Algorithmus 2.7 im Skript) können für beliebige gerichtete Netzwerke verwendet werde und stellen zusätzlich fest, ob es negative Kreise gibt (in diesem Fall ist die Distanz zweier Knoten nicht klar definiert).