Einführung in die lineare und kombinatorische Optimierung
Serie 8

Sven-Maurice Althoff (FU 4745454)
Michael R. Jung (HU 502133)
Felix Völker (TU 331834)

12. Dezember 2014

Aufgabe 29

Zeige zunächst: $a_{n+2} = a_n - a_{n+1}$:
$a_{n+2} = r^{n+2} = r^n \cdot r^2 = r^n \left(\frac{\sqrt{5}-1}{2}\right)^2 = r^n \left(\frac{1}{4}(5 - 2\sqrt{5} + 1)\right) = r^n \left(\frac{1}{4}(6 - 2\sqrt{5})\right)$
$= r^n \left(\frac{1}{2}(3 - \sqrt{5})\right) = r^n \left(1 + \frac{1}{2}(1 - \sqrt{5})\right) = r^n \left(1 - \frac{1}{2}(\sqrt{5} - 1)\right) = r^n(1 - r)$
$= r^n - r^{n+1} = a_n - a_{n+1}$.
Betrachte den Weg $p = (s32t)$ Setzen wir auf diesem Weg den Fluss gleich 1, so sind die Residualkapazitäten der Bögen e_1, e_2, e_3 gleich $(a_1, a_0, 0)$. Im Folgenden nennen wir das die *Situation* $(a_1, a_0, 0)$.
Betrachten wir nun die Pfade (in den entsprechenden Restnetzwerken)

$$\begin{aligned} p_1 &= (s1234t) \\ p_2 &= (s321t) \\ p_3 &= p_1 \\ p_4 &= (s432t) \end{aligned}$$

Sei das Netzwerk in der Situation $(a_n, a_{n+1}, 0)$, dann können wir den Fluss via p_1 um die Residualkapazität des Bogens e_2 $(= a_n)$ erhöhen und gelangen in die Situation $(a_n - a_{n+1} = a_{n+2}, 0, a_{n+1})$. Nun können wir allerdings den Fluss via p_2 um die Residualkapazität des Bogens $e_3 (= a_{n+1})$ erhöhen und gelangen in die Situation $(a_{n+2}, a_{n+1}, 0)$. Hier können wir den Fluss via p_3 um die Residualkapazität des Bogens $e_1 (= a_{n+2})$ erhöhen und gelangen in die Situation $(0, a_{n+1} - a_{n+2} = a_{n+3}, a_{n+2})$. Zuletzt erhöhen wir den Fluss via p_4 nun um die Residualkapazität des Bogens $e_3 (= a_{n+2})$ und gelangen in die Situation

$(a_{n+2}, a_{n+3}, 0)$. Hier sieht man nun, dass wir wieder in einer Situation $(a_k, a_{k+1}, 0)$ sind und, da wir zu Beginn via p auch in eine solche können, sieht man nun dass wir für jede natürliche Zahl n eine Folge länger n von augmentierenden Pfaden (z.B $p(p_1, p_2, p_3, p_4)^n$) finden, sodass im aktuellen Restnetzwerk noch augmentierende Pfade vorhanden sind. Folglich gibt es eine unendliche Folge von augmentierenden Pfaden, so dass der Algorithmus von Ford und Fulkerson nicht terminiert.

Aufgabe 30

Aufgabe 31

Aufgabe 32

Wir modellieren die Aufgabe als bipartites Matching-Problem. Wir haben auf der einen Seite die Geschenke G_m und auf der anderen Seite alle Fertigstellungszeitpunkte T_n. Die Bögen bekommen eine Kapazität von 1 und einen Kostenkoeffizienten der durch die Kostenfunktion $c(a) = c_g(t_g)$ festgellegt wird. Dann werden die super-Quelle(s) und -Senke(t) hinzugefügt und wir sind fertig.