Übung 09: Netzplantechnik

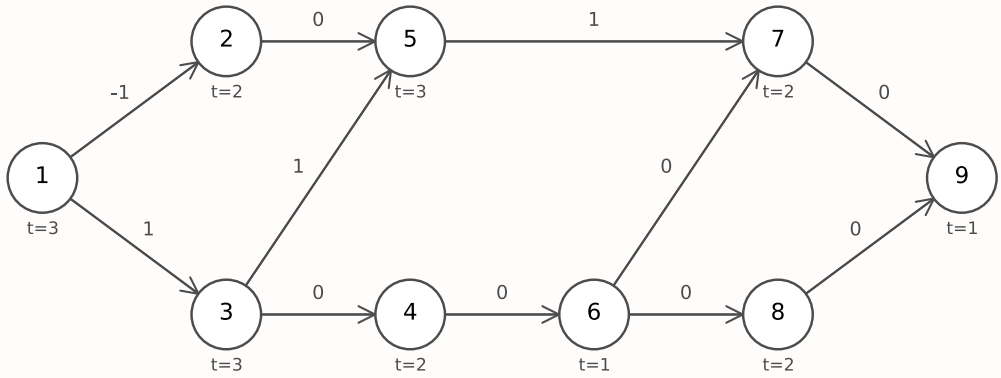
Grundlagen Operations Research

Video zur Übung

Aufgabe 1

Wir betrachten wieder das Beispiel **„Bau einer Lagerhalle"** aus Vorlesung 9. Der Baubeginn ist zum Zeitpunkt $\mathrm{FAZ}_1 = 5$ angesetzt worden. Mit dem Management ist eine Projektdauer von 18 ZE vereinbart worden (d. h. $\mathrm{SEZ}_9 = \mathrm{FAZ}_1 + 18 = 23$). Ferner gilt nun, dass vom Ende der Errichtung des Mauerwerks (Vorgang 3) bis zur Fertigstellung des äußeren Verputzes (Vorgang 8) höchstens 7 ZE vergehen dürfen (d. h. $\bar{d}^E_{3,8} = 7$).

Zur Erinnerung der Netzplan des Projekts (Dauern t_i an den Knoten, Mindestabstände d_{hi} an den Pfeilen):



Dieselben Daten als Tabellen: Dauern t_i und Mindestabstände d_{hi} (Normalfolge):

i	1	2	3	4	5	6	7	8	9
t_i	3	2	3	2	3	1	2	2	1

Pfeil $h \rightarrow i$	1→2	1→3	2→5	3→4	3→5	4→6	5→7	6→7	6→8	7→9	8→9
d_{hi}	–1	1	0	0	1	0	1	0	0	0	0

1. Bestimmen Sie die frühesten und spätesten Anfangs- und Endzeitpunkte der Vorgänge.

2. Bestimmen Sie die Pufferzeiten.
3. Überprüfen Sie Ihre Ergebnisse mit Julia.

> **Maximalabstand**
>
> Den Maximalabstand $\bar{d}^E_{3,8} = 7$ bilden Sie wie in Vorlesung 9 als **Rückwärtspfeil** von Vorgang 8 zu Vorgang 3 ab. Sein Mindestabstand ist $d_{8,3} = -(\bar{d}^E_{3,8} + t_3)$. Der Netzplan enthält damit einen Zyklus; rechnen Sie mit dem FIFO-Verfahren.

Ihre Lösung in Julia

Formulieren Sie die Vorwärts- und die Rückwärtsrechnung als **lineares Programm** (vgl. Vorlesung 9): zuerst $\min \sum_i \mathrm{FEZ}_i$ mit $\mathrm{FAZ}_1 = 5$, danach $\max \sum_i \mathrm{SAZ}_i$ mit $\mathrm{SEZ}_9 = 23$. Nennen Sie beide Modelle `modell`:

```
using JuMP, HiGHS

t = [3, 2, 3, 2, 3, 1, 2, 2, 1]
pfeile = [(1,2,-1), (1,3,1), (2,5,0), (3,4,0), (3,5,1), (4,6,0),
          (5,7,1), (6,7,0), (6,8,0), (7,9,0), (8,9,0)]
# Ergänzen Sie den Rückwärtspfeil für den Maximalabstand!

modell = Model(HiGHS.Optimizer)
set_silent(modell)

# IHR CODE HIER

optimize!(modell)
```

> **Selbstkontrolle: Vorwärtsrechnung**
>
> ```
> # Nach min sum(FEZ) mit FAZ[1] = 5:
> @assert isapprox(objective_value(modell), 130.0; atol = 1e-4) "Noch nicht
> korrekt, fehlt der Rückwärtspfeil 8→3 mit d = -10?"
> println("Vorwärtsrechnung      korrekt,      FEZ-Summe      =      ",
> objective_value(modell))
> println("Vorwärtsrechnung korrekt.")
> ```

Selbstkontrolle: Rückwärtsrechnung

```
# Nach max sum(SAZ) mit SEZ[9] = 23:
@assert isapprox(objective_value(modell), 148.0; atol = 1e-4) "Noch nicht
korrekt, ist SEZ9 auf 23 fixiert?"
println("Rückwärtsrechnung      korrekt,      SAZ-Summe      =      ",
objective_value(modell))
```

Bibliografie