

OPTIMIERUNG VON JOB-SHOP PROBLEMEN MIT REINFORCEMENT LEARNING

TOM KLEIN & KAI KLEMMER IM RAHMEN DES PROJEKTS ZUR DATENANALYSE (WS 19/20)

AGENDA

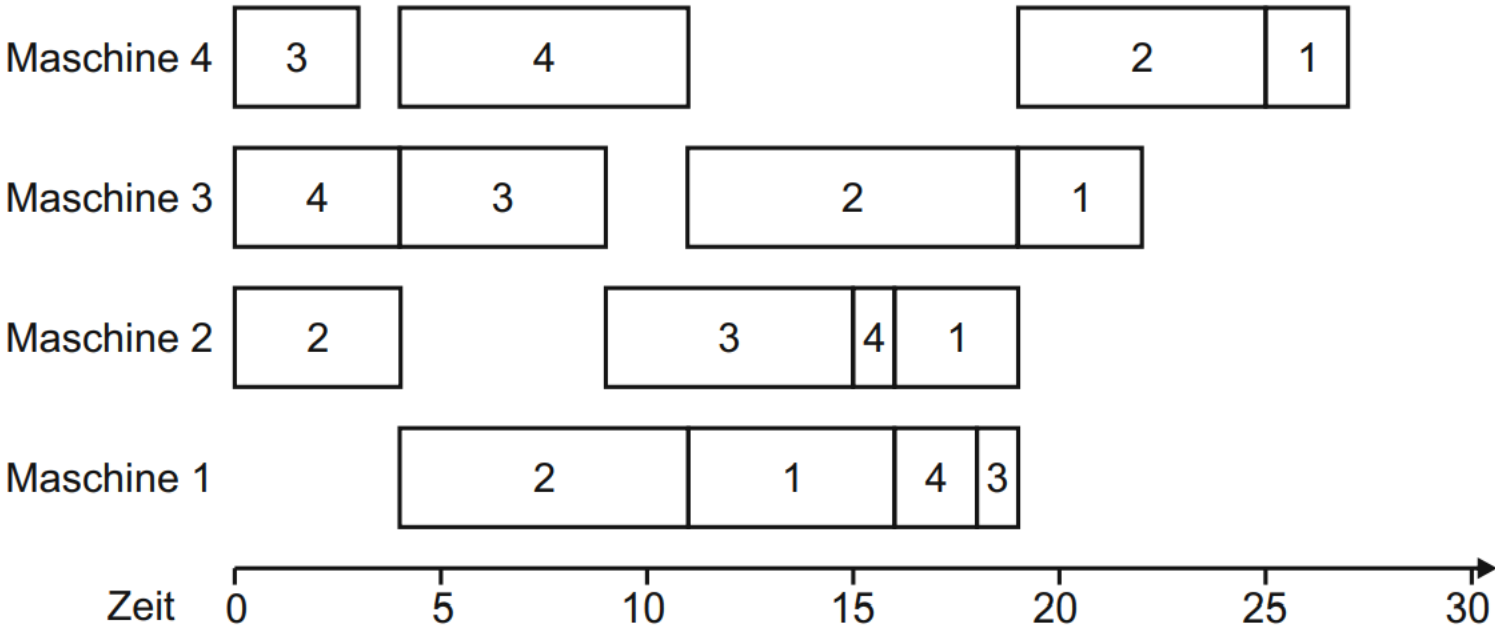
- ▶ Recap
 - ▶ Was ist nochmal ein Job-Shop?
 - ▶ Was ist nochmal Reinforcement Learning?
- ▶ Zielsetzung
- ▶ Modellierung
 - ▶ I/O via JSON
 - ▶ Q-Learning via Multi-Agent Learning
 - ▶ Reward-Funktion
- ▶ Probleme
- ▶ Demo
- ▶ Reflexion
- ▶ Quellen

DAS JOB-SHOP PROBLEM

- ▶ $\alpha = Jm, m \in \mathbb{N}$
- ▶ $n \in \mathbb{N}$ viele Aufträge
- ▶ Jeder Auftrag auf jeder Maschine
- ▶ Reihenfolge der Aufträge auf Maschinen ist vorgegeben
- ▶ Nicht alle Aufträge müssen die gleiche Reihenfolge haben

BEISPIEL

j	1	2	3	4		j	1	2	3	4
p_{1j}	5	7	1	2	1. zu besuchende Maschine	1	2	4	3	
p_{2j}	3	4	6	1	2. zu besuchende Maschine	2	1	3	4	
p_{3j}	3	8	5	4	3. zu besuchende Maschine	3	3	2	2	
p_{4j}	2	6	3	7	4. zu besuchende Maschine	4	4	1	1	



ABLAUF



Q-TABELLE, Q-FUNKTION

- ▶ Q-Tabelle (Zustand x Aktion)
- ▶ $Q'(s_t, a_t) = (1 - \alpha) \cdot Q(s_t, a_t) + \alpha \cdot \left(r_t + \gamma \cdot \max_a Q(s_{t+1}, a_{t+1}) \right)$
- ▶ $\alpha = \frac{1}{1 + \text{count}(s, a)}$ (Lernrate)
- ▶ γ ist "Discount"-Faktor

ZIELSETZUNG

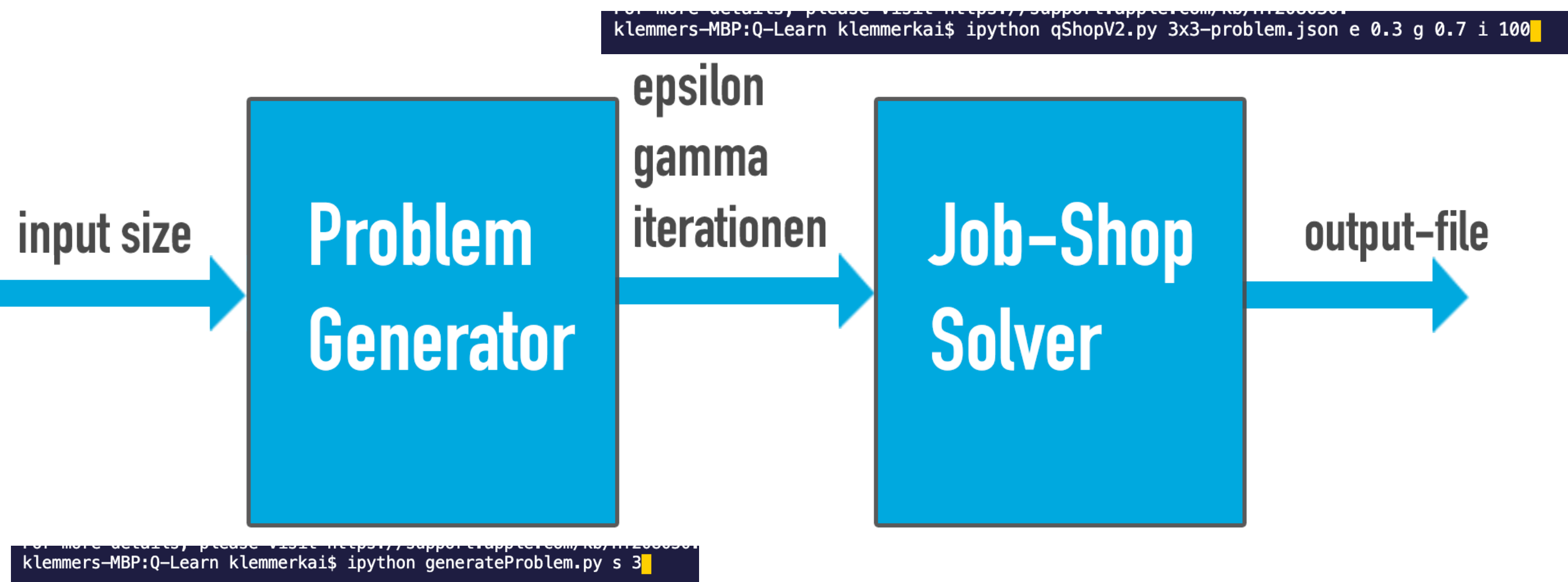
- ▶ Programm zur Lösung von Job-Shop Problemen
- ▶ Verwendung von Q-Learning zum Lernen von besten Aktionen
- ▶ I/O für Probleme und Lösungen
- ▶ Visualisierung der Lösung

MODELLIERUNG

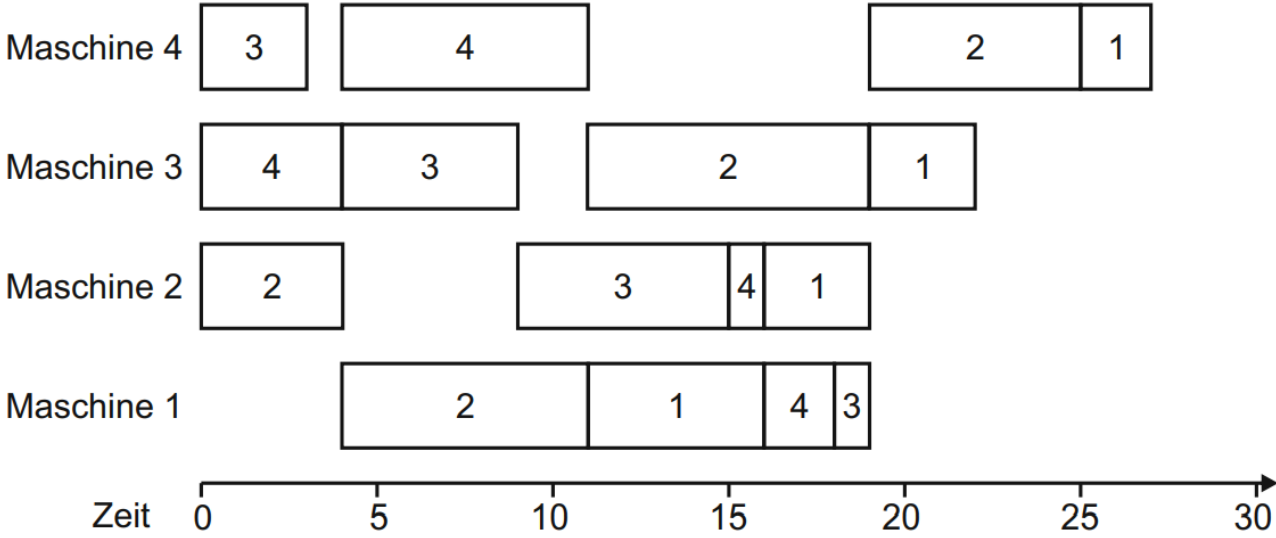
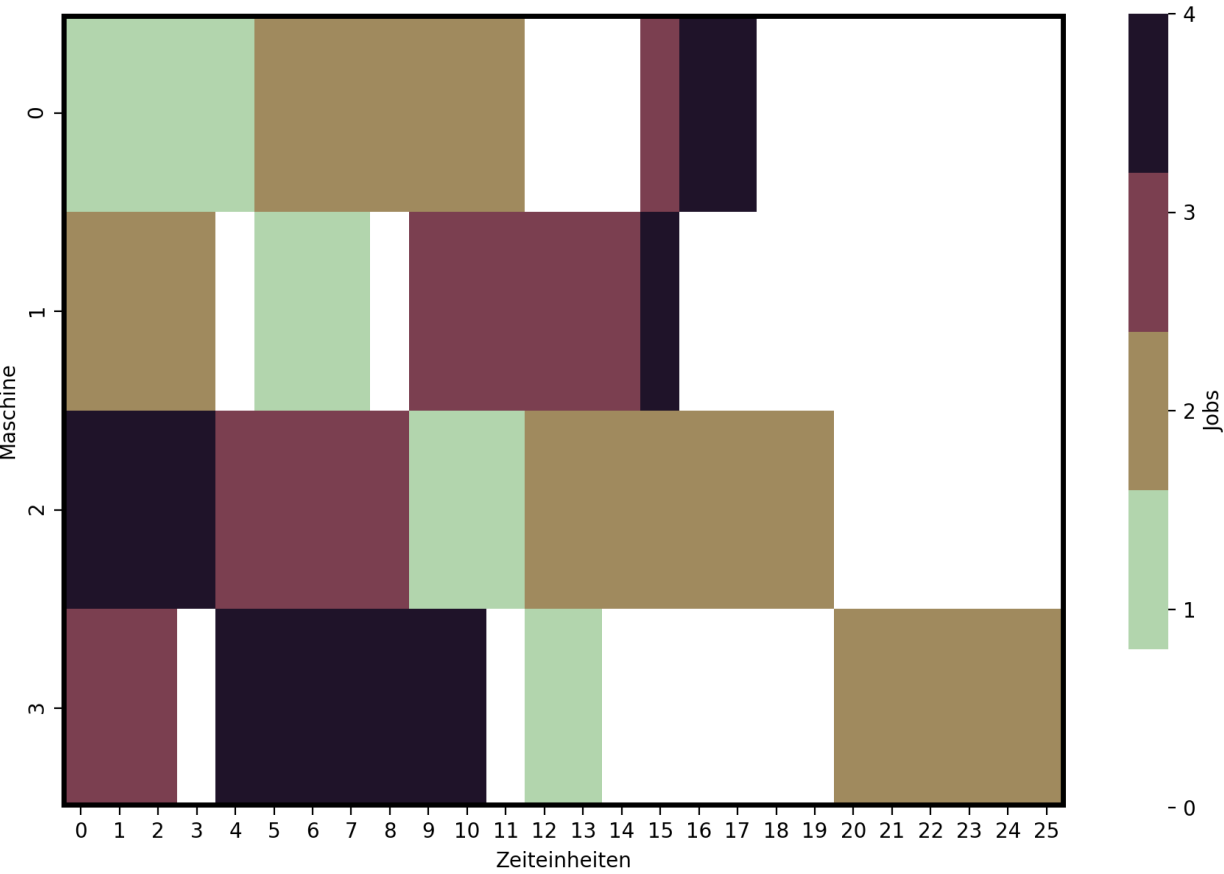
- ▶ Programmiersprache: Python
- ▶ File-gesteuertes I/O-System mit JSON
- ▶ Aufruf über Konsole

```
{  
  "name": "4M4T",  
  "kommentar": "",  
  "reihenfolge": [  
    [0, 1, 3, 2],  
    [1, 0, 2, 3],  
    [2, 2, 1, 1],  
    [3, 3, 0, 0]  
  ],  
  "bearbeitungszeiten": [  
    [5, 7, 1, 2],  
    [3, 4, 6, 1],  
    [3, 8, 5, 4],  
    [2, 6, 3, 7]  
  ],  
  "zielfunktion": "Cmax"  
}
```


WORKFLOW

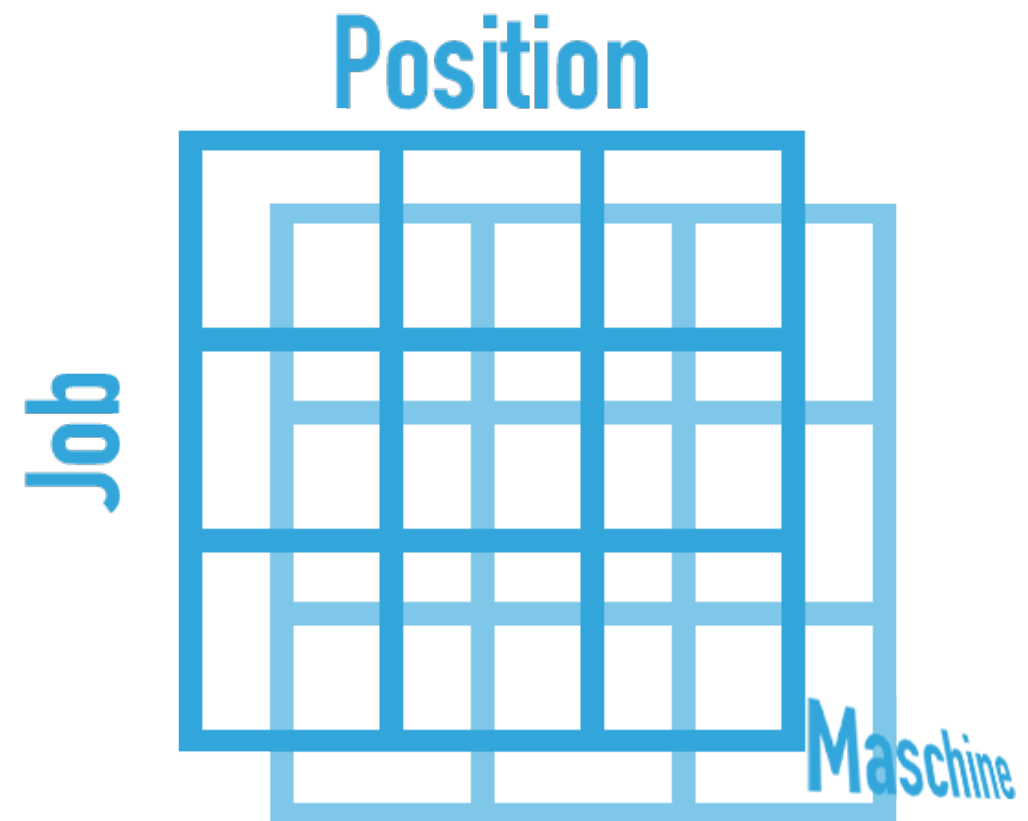


OUTPUT



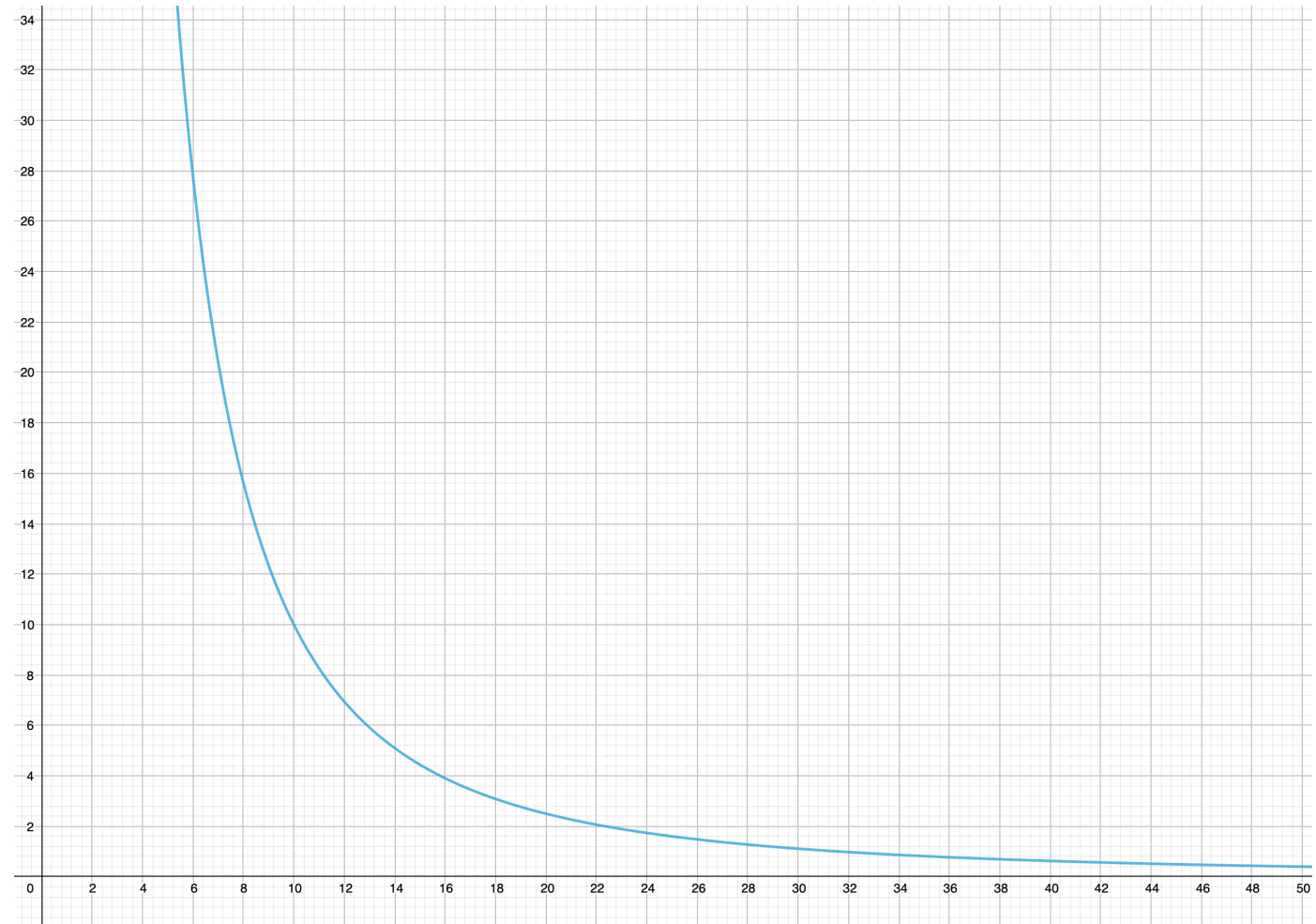
MULTI-AGENT REINFORCEMENT LEARNING

- ▶ Wahl der Dimensionen
 - ▶ Jobs, Maschinen, Zeit
- ▶ Eine Q-Table pro Maschine
- ▶ Random-Werte vs. Nullen
- ▶ Maximierung vs. Minimierung



DIE REWARD-FUNKTION

- ▶ Schätze Fertigungszeitpunkt für Maschine
- ▶ Infimum für Maschine = $\sum \text{Bearbeitungszeiten}$
- ▶ $x = \text{Schätzung} - \text{Infimum}$
- ▶ $r = \frac{1000}{x^2}$



PSEUDOCODE

Lese JSON

Solange $i < \text{maximale Iterationen}$:

Solange Ablaufplan nicht fertig:

wähle zulässiges Job-Maschinen-Paar
Update Q-Table

Wenn neuer Ablaufplan besser als bester Ablaufplan:

Setze $i = 0$
bester Ablaufplan = neuer Ablaufplan

$i++$

Visualisiere besten Ablaufplan und schreibe ihn in JSON

PROBLEME

- ▶ Job-Shops mit mehreren optimalen Lösungen

- ▶ Eindeutigkeit der Q-Table

469	0	471	0
672	0	689	0
0	895	0	913
0	992	0	962

- ▶ Optimale Lösung pro Maschine vs. Optimale globale Lösung

DEMO

REFLEXION

- ▶ Finden von zulässigen Lösungen
- ▶ Gelerntes oft nicht eindeutig
- ▶ Modellierung Q-Table nicht optimal
- ▶ Reward-Funktion zu spezifisch

QUELLEN

- ▶ Dörn, Sebastian (2018): Programmieren für Ingenieure und Naturwissenschaftler. Berlin, Heidelberg: Springer Berlin Heidelberg
- ▶ Jaehn, Florian; Pesch, Erwin (2019): Ablaufplanung. Berlin, Heidelberg: Springer Berlin Heidelberg
- ▶ Beke, Tom (2013): Multi-Agent Reinforcement Learning in a flexible Job-Shop Environment: The VCST Case. Universiteit Gent
- ▶ Zhang, Wei; Diettrich, Thomas G. (1995): A Reinforcement Learning Approach to Job-Shop Scheduling. Oregon State University

**VIELEN DANK FÜR
IHRE
AUFMERKSAMKEIT!**