

Job Shop Scheduling

Praktische Arbeit

Problem Solving and Search in AI

Sommersemester 2017

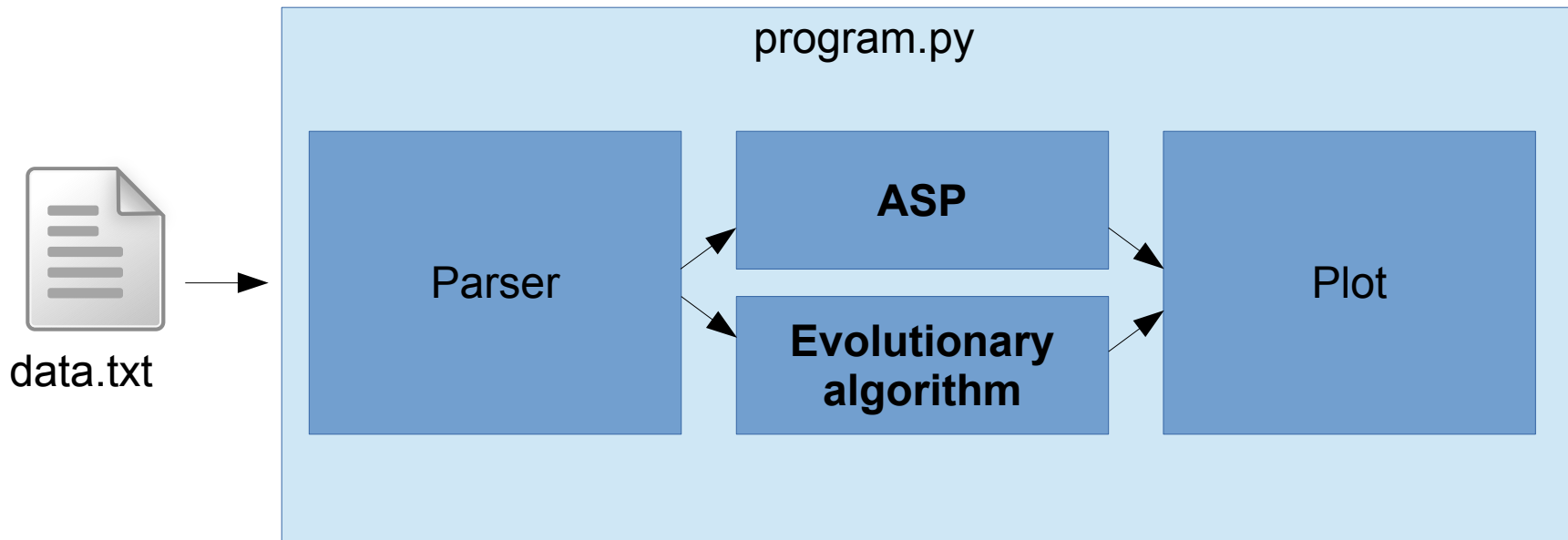
Sebastian Lindner & Dominik Olwig

Unser Programm

- Sprache: Python
- Zusätzliche Bibliotheken:
 - matplotlib.pyplot → graphische Ausgabe
 - Numpy → mathematische Berechnungen

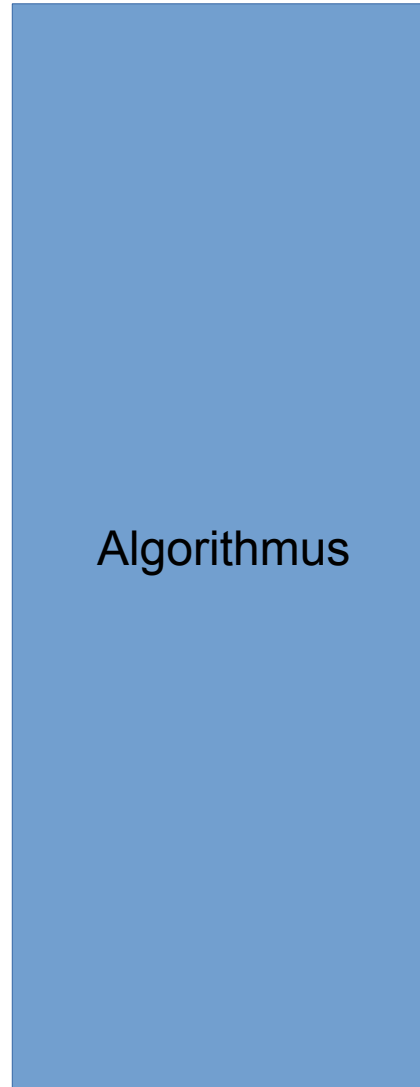
Unser Programm

- Sprache: Python
- Zusätzliche Bibliotheken:
 - matplotlib.pyplot → graphische Ausgabe
 - Numpy → mathematische Berechnungen



Repräsentation der Daten

- Machine_count (int)
- [job₁, job₂, ...]



- [machine_job₁,
machine_job₂,
...]

→ job_i = [
 (machineNr, length),
 (machineNr, length), ...
]

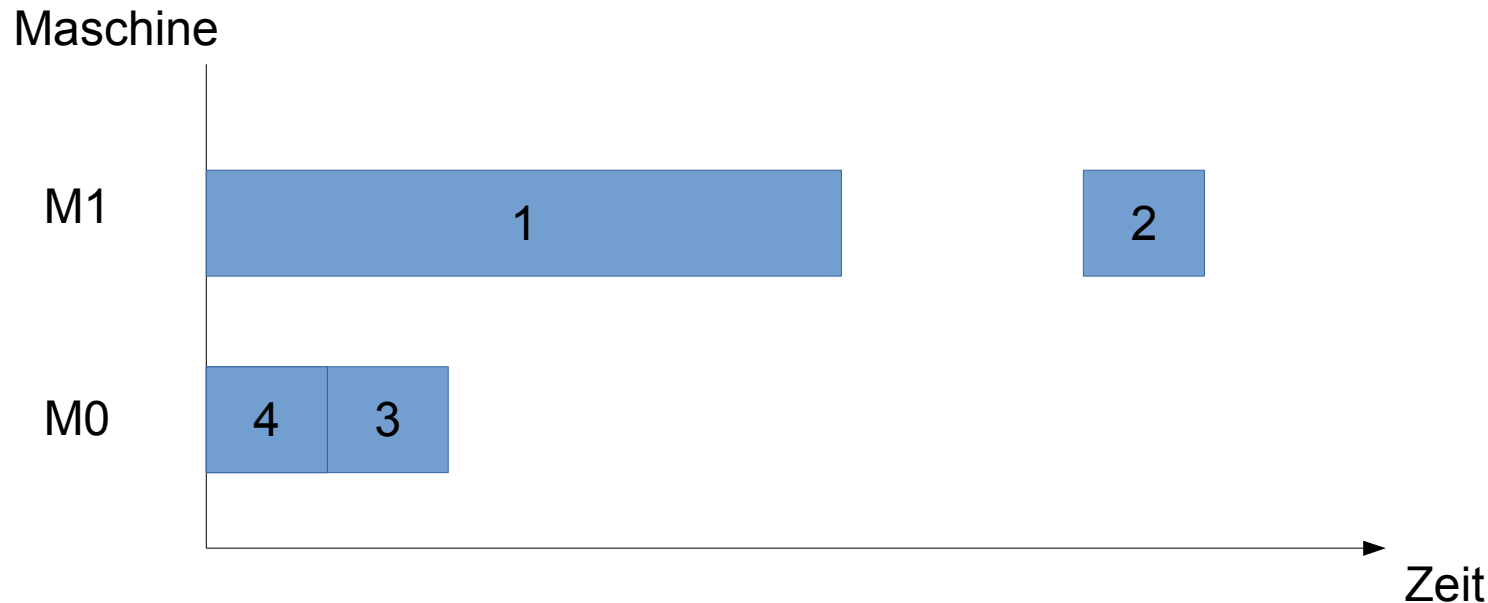
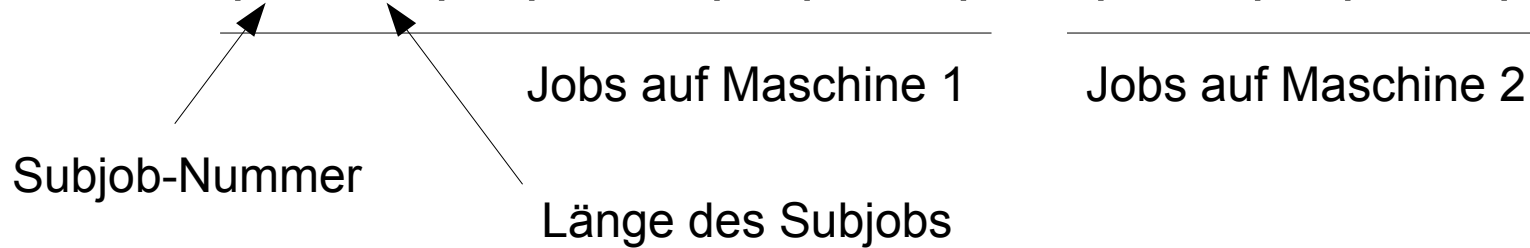
→ machine_job_i = [
 (begin, length, jobNr),
 (begin, length, jobNr),
 ...
]

ASP

- Modellieren der Problemstellung
- Subjobs in Jobs mit zusätzlichen Indizes
- Constraints:
 - nur ein Subjob zu einem Zeitpunkt
 - Subjob muss vor Nachfolger beendet werden
→ vorheriger Index < nachfolgender Index
 - Startzeitpunkt + Länge \geq Startzeitpunkt des Folgejob

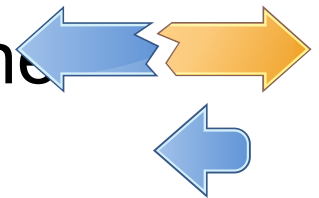
Evolutionary Algorithm

- $[[(1, 12), (-1, 4), (2, 4)], [(4, 2), (3, 2)]]$

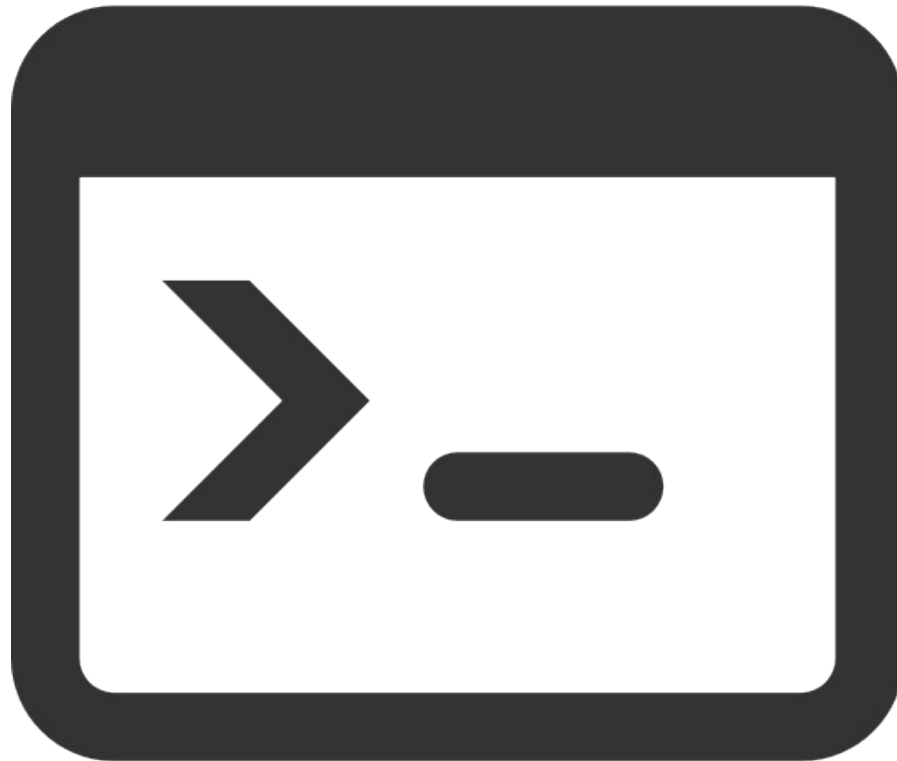


Evolutionary Algorithm

- **Startpopulation:** Mutationen aus sequentieller Lösung
- **Mutation:** Vertauschen zweier Subjobs oder Verschieben eines Jobs innerhalb der Maschine
- **Fitness-Funktion:**
 - Variante 1: Länge der Lösung
 - Variante 2: Aufsummierte Starts der Subjobs
→ bei bisherigen Tests gleich gut



Demo



Probleme

ASP

- Optimierung bzw. Minimierung von Lösungen
- **Evolutionärer Algorithmus**
- Performance
- Rekombination – Funktion