HTWD
Hochschule für Technik und
Wirtschaft Dresden
University of Applied Sciences

Scheduling mit Answer Set Programming

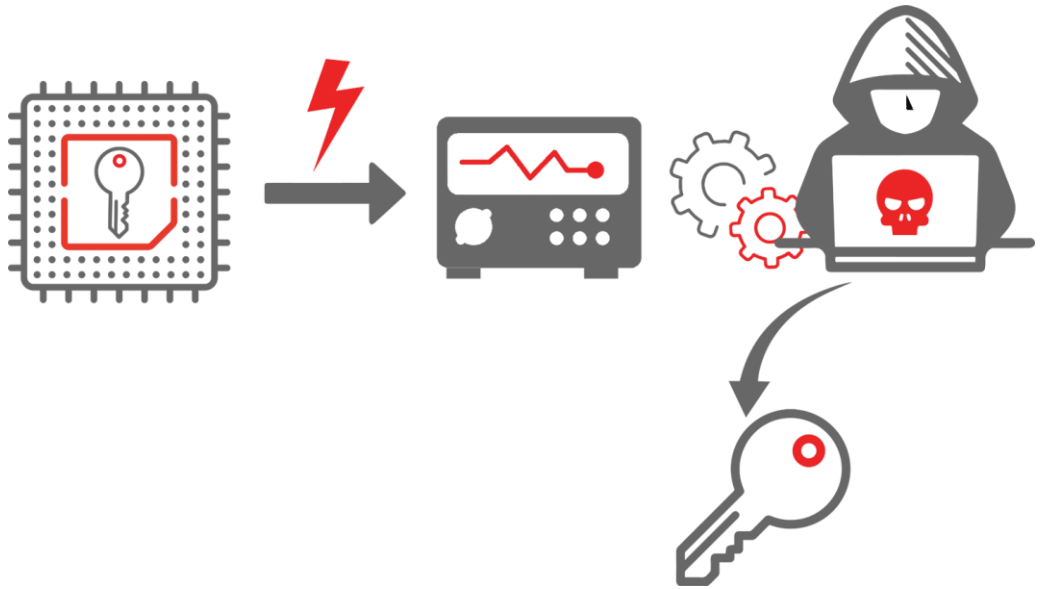
Tom Marinovic

Betreut durch: Prof. Dr. Westfeld

19.06.2024

Motivation

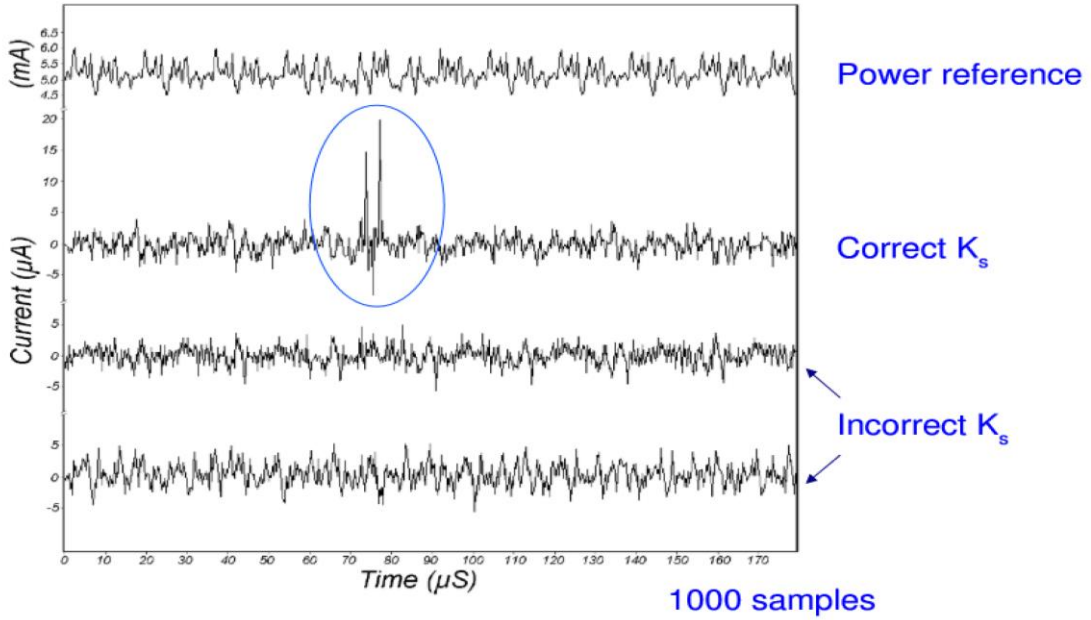
- Viele Probleme lassen sich vollständig beschreiben ohne, dass eine Lösung entsteht
- Answer Set Programming ist eine deklarative Programmiersprache mit Fokus auf NP-harte Suchprobleme
- Sehr effiziente Lösungssuche mittels SAT-Solver
- Statisches Planen (Scheduling) von Taskmengen unter speziellen Bedingungen



https://b3h9h6w4.rocketcdn.me/wp-content/uploads/2023/01/SCA_Diagram.png

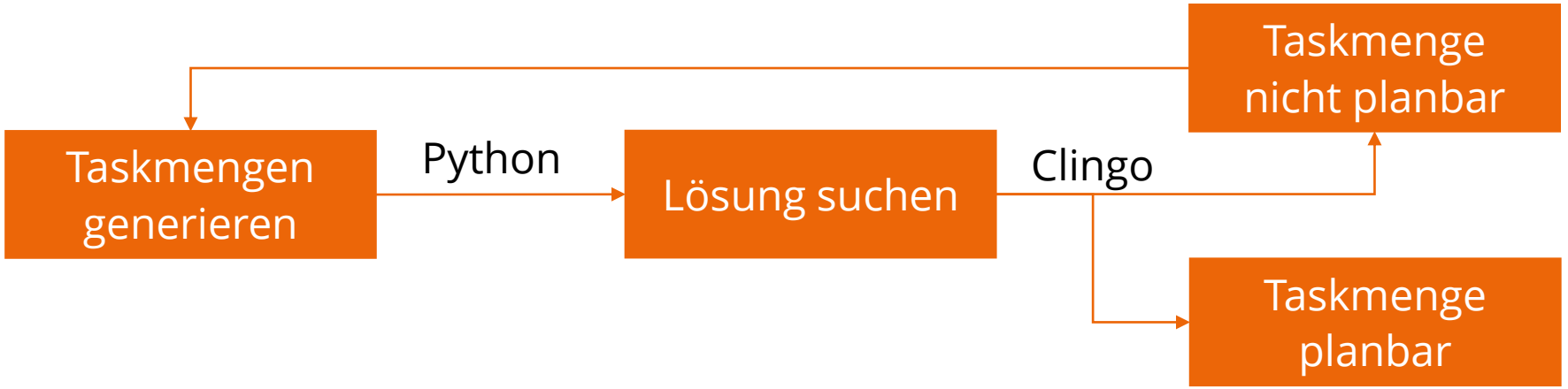
Sicherheitskritisch

- Problem des Seitenkanalangriffs über die Ausführungszeit
- Ausführungszeit eines Tasks ist eine Sicherheitskritische Information, welche nicht ersichtlich sein sollte
- Ein unkritischer Prozess kann sehen, wie lange ein kritischer Prozess arbeitet anhand der Blockierung der CPU
- Lösung über statisches Scheduling



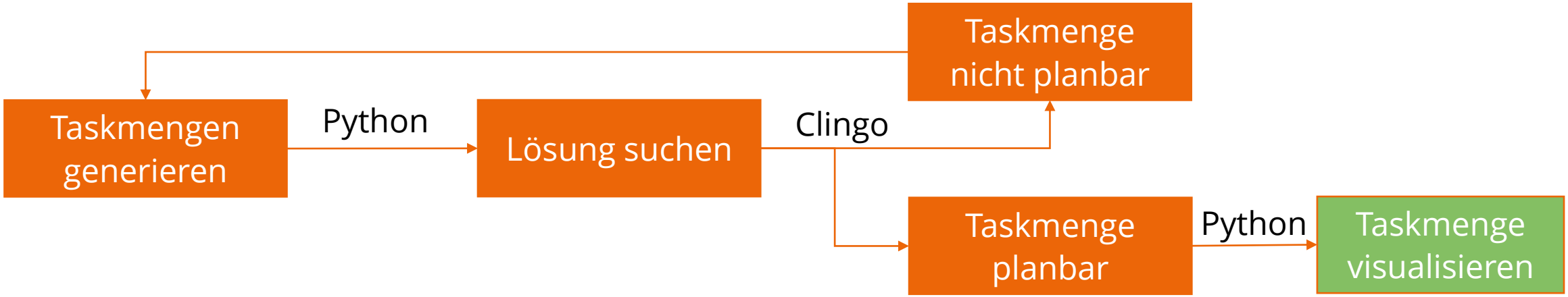
IST Stand

- Programme für folgende Funktionen existieren
  - Testplänen generieren
  - Überprüfen ob eine gegebene Taskmenge unter speziellen Anforderungen planbar ist
  - Testpläne automatisch erstellen und validieren

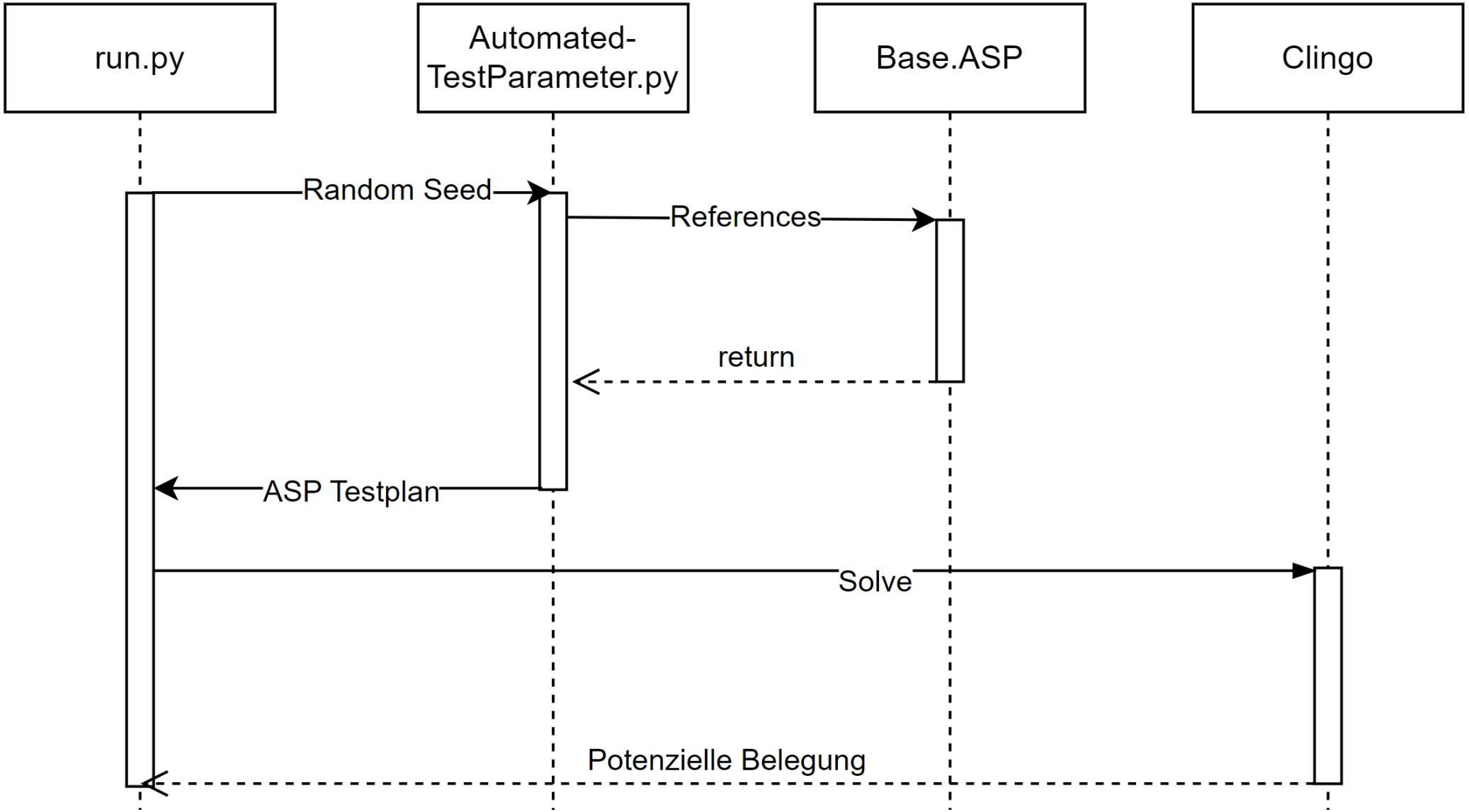


Aufgaben

- Prüfen des Planverhaltens bei verschiedenen Parametern (z.B. CPU-Anzahl, Anzahl der Tasks)
- Analyse, warum manche Pläne längere Validierungszeiten haben als andere
- Erweitern der aktuellen Pipeline um Visualisierung von validen Plänen



Ablauf Testplanerstellung



ASP-Beispiel

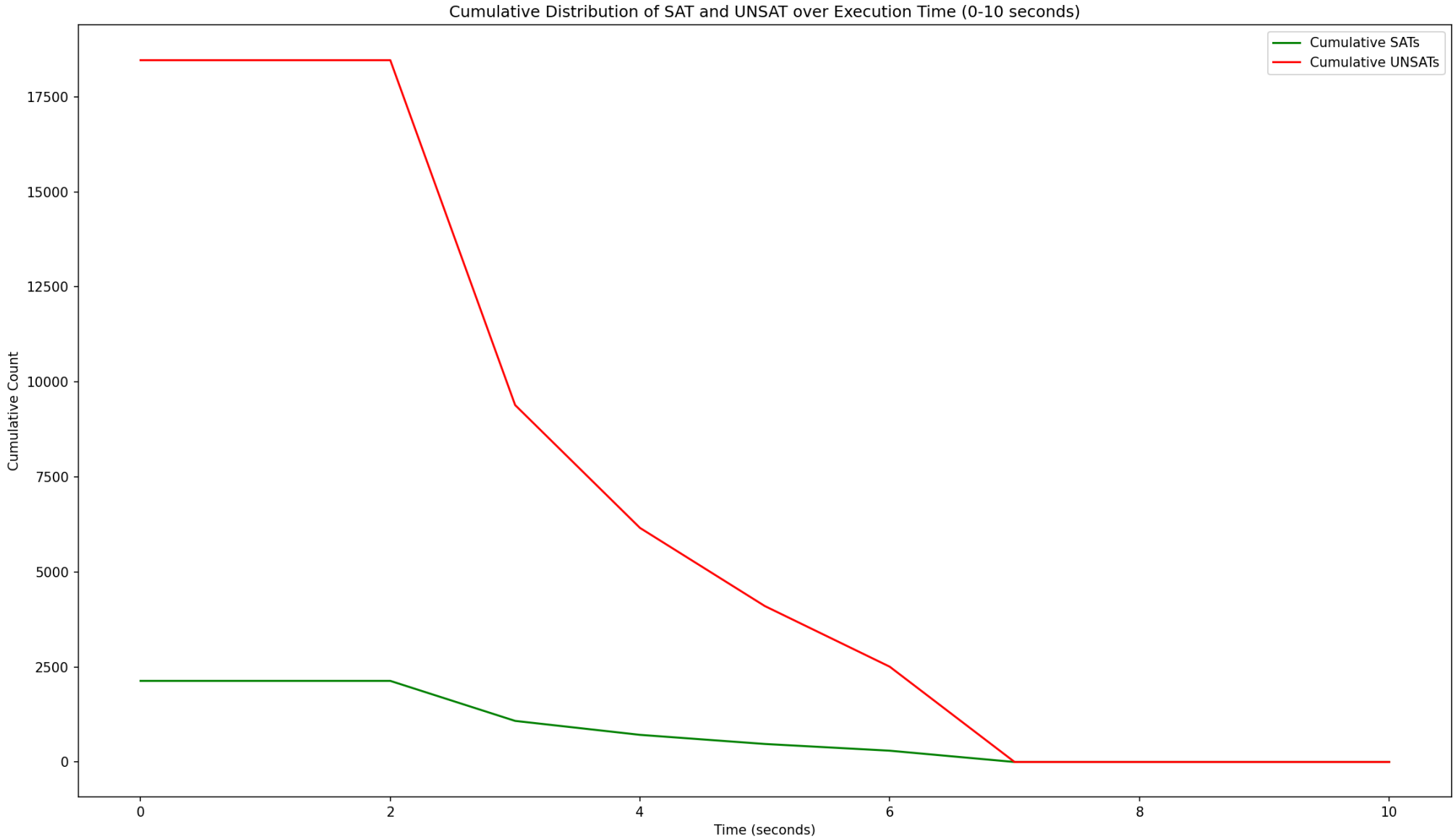
```
partition_possible_configuration(PlanId, S, (MF, F, X)) :-
    plan(PlanId),
    partition(S),
    configuration(PlanId, (MF, F, X)),
    partition_constraints(PlanId, S, Min_Freq, Min_Length, Max_Freq, Max_Length),
    configuration(PlanId, (MF, F, X)),
    F >= Min_Freq,
    F <= Max_Freq,
    F * MF >= Min_Length,
    F * MF <= Max_Length.
```

- Definition einer Tabelle
- Setzt sich aus Subtabellen zusammen

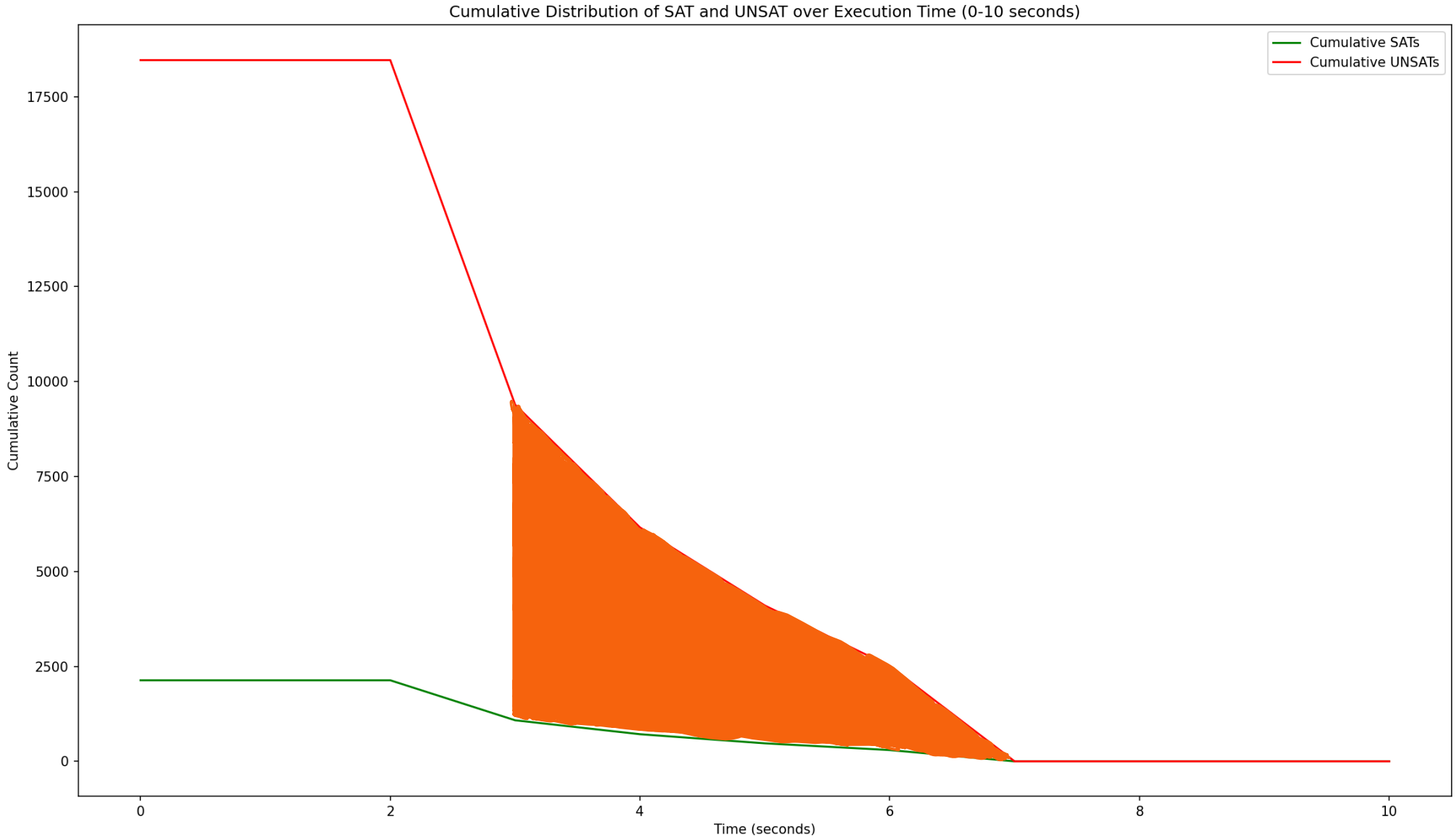
Herausforderungen

- ASP-Programmierung unterscheidet sich stark von „regulärer“ Programmierung (C / Python)
- Es werden keine Abläufe Programmiert
- Das Scheduling Beispiel ist komplex
- SAT-Solver Optimierung notwendig für korrekte Ergebnisse
- Manche Taskmengen dauern länger zur Prüfung ob valide Pläne existieren

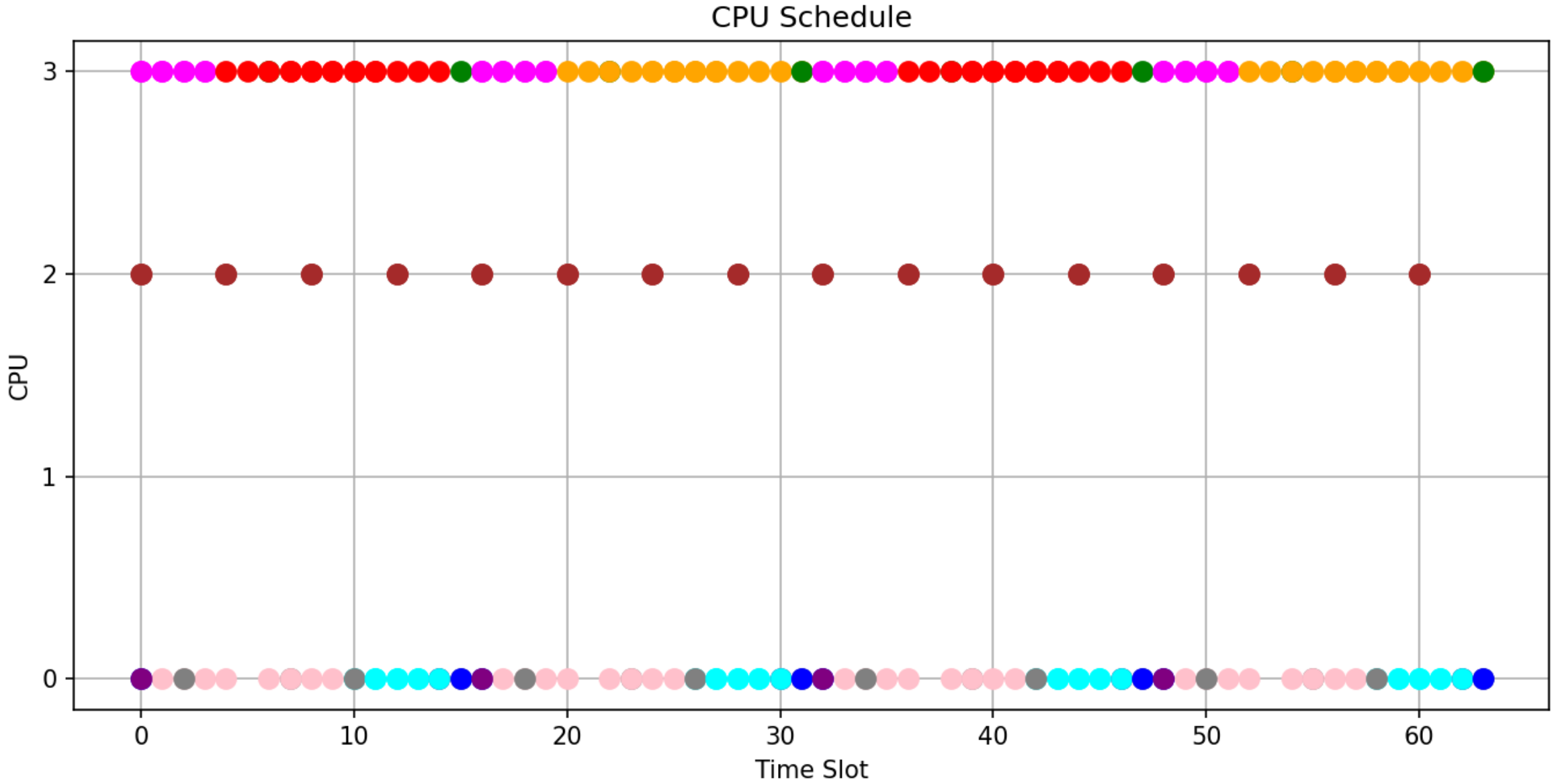
Problem der unterschiedlichen Bearbeitungszeit



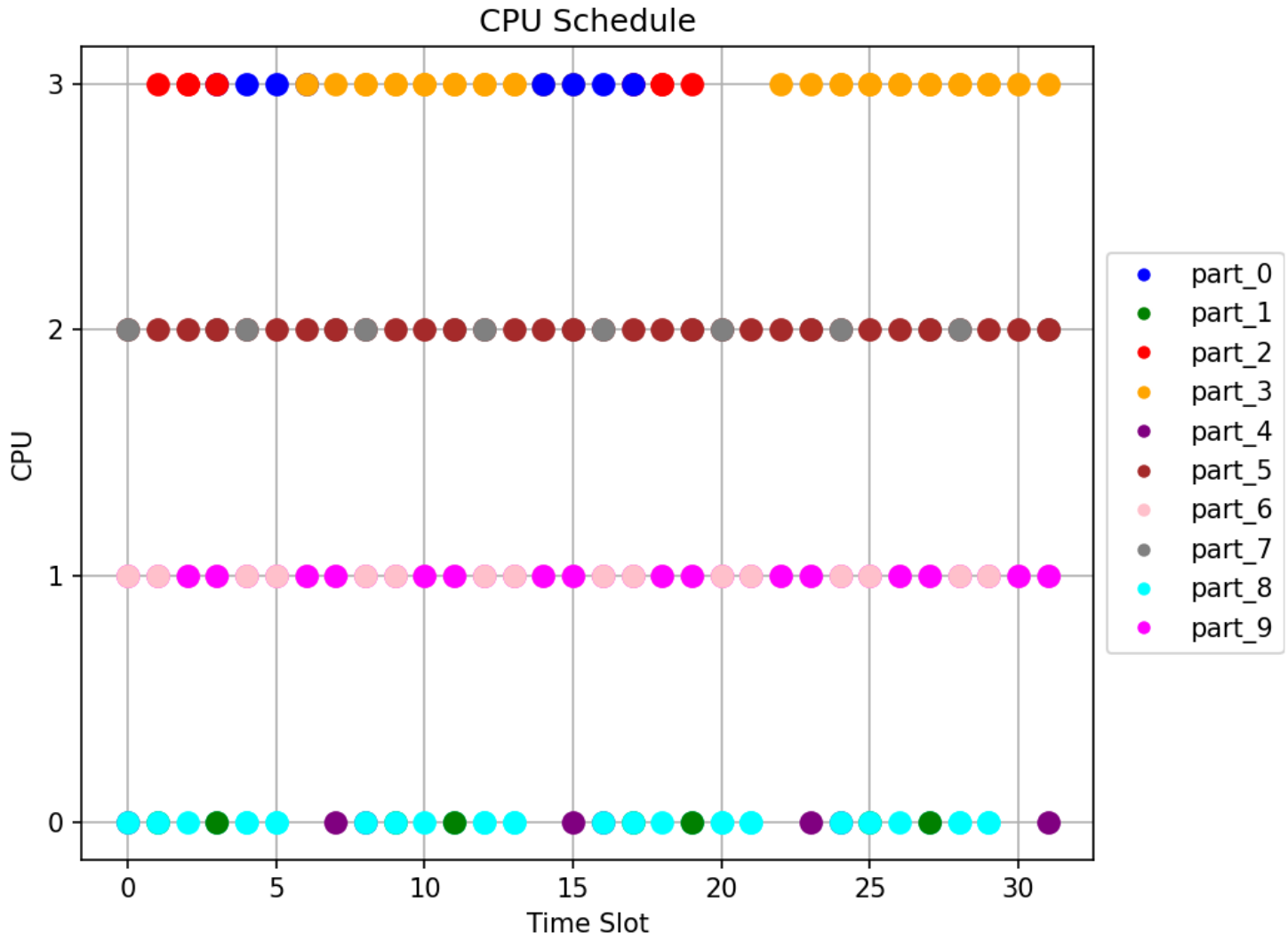
Problem der unterschiedlichen Bearbeitungszeit



Geplant aber nicht optimal

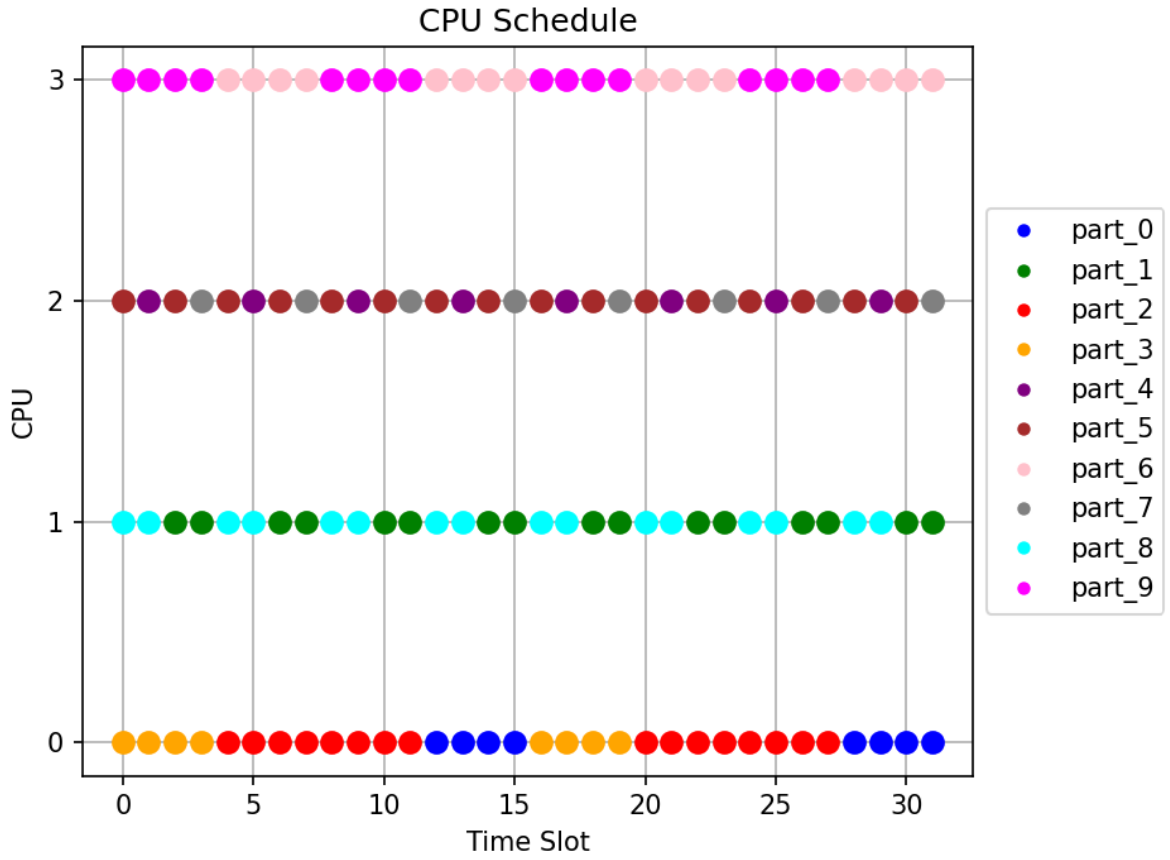


Geplant und Optimal



Projektergebnisse

- Visualisierungstool in Python, welches einen gegeben Plan darstellt
- Auswertung des Schedulers
- Anpassen der Plangenerierung
- Statistische Auswertung der Planparameter



Quellen

- Standaert, FX. (2010). Introduction to Side-Channel Attacks. In: Verbauwhede, I. (eds) Secure Integrated Circuits and Systems. Integrated Circuits and Systems. Springer, Boston, MA. https://doi.org/10.1007/978-0-387-71829-3_2
- Philipp, T., Roland, V., & Schweizer, L. (2021). Smoke test planning using answer set programming. *International Journal of Interactive Multimedia and Artificial Intelligence*, *6*(5), 57–65. https://doi.org/10.9781/ijimai.2021.02.003
- Gebser, M., Kaminski, R., Kaufmann, B., Ostrowski, M., Schaub, T., & Thiele, S. (2010). A User's Guide to gringo, clasp, clingo, and iclingo *.
- Lifschitz, V. (2019). Answer Set programming. In Springer eBooks. https://doi.org/10.1007/978-3-030-24658-7

Vielen Dank für Ihre Aufmerksamkeit.