

# Themenvorschlag für eine Abschlussarbeit

## *Heuristiken für Parallel Task Scheduling*

Eignung: Bachelor

### **Gegeben:**

- Eine Menge  $J$  von  $n$  Jobs mit Ausführungszeiten  $p_j$  und Maschinenbedarf  $q_j$ .
- Eine Menge von  $m$  Maschinen

### **Gesucht:**

Ein Schedule (Ablaufplan), der jedem Job  $j$  einen Startpunkt  $s_j$  zuordnet. Die Jobs dürfen nicht unterbrochen werden (d.h. sie werden im Intervall  $[s_j, s_j + p_j)$  ausgeführt). Der Schedule ist zulässig falls zu jedem Zeitpunkt  $t$  die Anzahl der genutzten Maschinen durch  $m$  beschränkt ist; d.h.

$$\sum_{j \in J: t \in [s_j, s_j + p_j)} q_j \leq m.$$

In einem Algorithmus für ein anderes Problem, dem sogenannten Multiple Cluster Scheduling, tritt dieses Problem als Teilproblem auf. Hier wird ein Schedule benötigt den man in zwei Abschnitte aufteilen kann einen Abschnitt mit Ausführungszeit  $(5/4)\text{OPT}$  und einen weiteren mit Ausführungszeit  $p_{\max}$ .

Für das Parallel Task Scheduling Problem gibt es bereits Algorithmen. Für diese Arbeit interessant sind zum einen eine 2-Approximation von Turek et al. [1] zum anderen eine  $(3/2)\text{OPT} + p_{\max}$  aus dieser Arbeitsgruppe.

## **Ziele der Arbeit**

Das Hauptziel der Arbeit ist es einen schnellen Algorithmus mit Approximationsgarantie  $(5/4)\text{OPT} + p_{\max}$  oder  $(4/3)\text{OPT} + p_{\max}$  zu finden. Da dies jedoch nicht sicher möglich ist, sollen mindestens die folgenden Punkte erfüllt werden:

- Implementierung des Algorithmus von Turek et al. [1]
- Implementierung des  $(3/2)\text{OPT} + p_{\max}$  Algorithmus.

- Entwicklung von **mindestens** einer eigenen heuristischen Idee und deren Implementierung + Analyse der Ausführungszeit und Gütegarantie
- Vergleich der Implementierungen der Heuristiken (Laufzeit, Speicherverbrauch, Nähe zum Optimum, etc)
- Visualisierung der Packungen (Optional aber empfohlen)
- Generieren von Instanzen, die schlecht möglichste Ergebnisse für die ersten beiden Algorithmen liefern (Optional)

## Literatur

- [1] John Turek, Joel L. Wolf, and Philip S. Yu. Approximate algorithms scheduling parallelizable tasks. In *4th annual ACM symposium on Parallel algorithms and architectures (SPAA)*, pages 323–332, 1992. doi:10.1145/140901.141909.

## Kontakt und Informationen

Max Deppert  
CAP4, R.1003  
made@informatik.uni-kiel.de