

# Proseminar: Parallele Algorithmen

Von Theorie zu Praxis

Peter Sanders, Jochen Speck, Daniel Funke | 19. April 2016

INSTITUT FÜR THEORETISCHE INFORMATIK



# Organisatorisches

Datum	Dauer	Ereignis
19.04.2016		Auftaktveranstaltung
xx.xx.2016	<i>TBD</i>	Referat: Wissenschaftliches Arbeiten
	<i>5 Wochen</i>	<i>Literaturarbeit &amp; Schreiben der Ausarbeitung</i>
		<i>Konsultation mit Betreuer</i>
24.05.2016		<b>Abgabe:</b> Gliederung der Ausarbeitung
	<i>2 Wochen</i>	<i>Fertigstellen der Ausarbeitung</i>
07.06.2016		<b>Abgabe:</b> Draft der Ausarbeitung
	<i>2 Woche</i>	<i>Peer Review von zwei Ausarbeitungen</i>
21.06.2016		<b>Abgabe:</b> Peer Review
	<i>1 Woche</i>	<i>Verbesserung der Ausarbeitung</i>
28.06.2016		<b>Abgabe:</b> Finale Ausarbeitung
	<i>2 Wochen</i>	<i>Anfertigen der Folien</i>
12.07.2016		<b>Abgabe:</b> Vortragsfolien
	<i>1 Woche</i>	<i>Feedback durch Betreuer</i>
19.07.2016	<i>TBD</i>	Präsentationen

## 1. Lernziel

Wie nutze ich wissenschaftliche Artikel für meine eigene Arbeit?

## 2. Lernziel

Wie stelle ich wissenschaftliche Ergebnisse dar?

## Inhalt:

- Darstellung des Algorithmus/Modells
- Einordnung in angewendeten Kontext
- Probleme/Schwierigkeiten des Modells/Algorithmus in der Praxis
- Beispiele

## Formalien:

- 8-10 Seiten + Referenzen
- in  $\text{\LaTeX}$  geschrieben  $\Rightarrow$  wir stellen eine Vorlage
- Deutsch oder Englisch

## Inhalt:

- Aspekte der Ausarbeitung
- Fokus auf Verständlichkeit/Nachvollziehbarkeit

## Formalien:

- Dauer: 15 Minuten Vortrag + 5 Minuten für Fragen
- Folien in  $\LaTeX$  Beamer, lpe, ...
- Abgabe der Folien eine Woche vor Präsentation  
⇒ wir geben euch Feedback

Gewichtung	Kriterium
45 %	Ausarbeitung
45 %	Vortrag
10 %	Peer Review

## Ausarbeitung:

- Inhalt
  - Auswahl relevanter Aspekte der geg. Literatur
  - Sinnvolle Darstellung des Themenkerns
  - Verbindung zu Einordnung/Anwendung/Beispielen
- Form
  - Sprachliche Korrektheit/Stil
  - Anschauliche Abbildungen
  - Korrekte Zitierweise

Gewichtung	Kriterium
45 %	Ausarbeitung
45 %	Vortrag
10 %	Peer Review

## Vortrag

- Folien
  - Inhalt, Korrektheit, Übersichtlichkeit
  - Passend zum Vortrag
- Vortragsform
  - Angemessene inhaltliche Darstellung
  - Vortragsstil
  - Beantwortung von Fragen



Gewichtung	Kriterium
45 %	Ausarbeitung
45 %	Vortrag
10 %	Peer Review

## Peer review

- Inhaltliche Korrektheit der Arbeit
- Keine Paraphrase der geg. Literatur
- Sinnvolle und angemessene Kritik

Wir geben euch ein Hinweisblatt mit Review-Kriterien an die Hand

## EasyChair:

[www.easychair.org](http://www.easychair.org)

*EasyChair is a free web-based conference management software system used, among other tasks, to organise paper submission and review. EasyChair is widely used in the scientific community, with reportedly more than one million users in 2014. [Wikipedia]*



## EasyChair Konferenz [ProSemParaAlg2016](#)

- Jeder Teilnehmer braucht EasyChair Account (kostenlos)
- EasyChair hat mehrere *Rollen*
  - *Authors* reichen Paper ein
  - *Program Committee Members* erstellen Reviews
  - *Chairs* führen Administration durch
- Teilnehmer sowohl Autoren als auch PCM
- Nach der Submission: „bidding“ um Reviews
- Wir als Chairs führen endgültige Verteilung der Reviews durch

# Themen

## Literatur:

Jaja, J., *Introduction to Parallel Algorithms*, Kapitel 1

Greenlaw R. et al., *Limits to Parallel Computation*, Kapitel 2

- Maschinenmodelle – PRAM, Boolean Circuit
- Speichermodelle – shared, distributed
- Netzwerkmodelle – ring, hypercube
- Performance-Metriken – speedup, efficiency

## 2. $P$ -completeness

### Literatur:

Greenlaw R. et al., *Limits to Parallel Computation*, Kapitel 3-5,8

- Komplexitätsklassen  $P$ ,  $NC$
- Reduktionen
- $P$ -completeness und  $NC \subseteq P$
- Beispielprobleme und -algorithmen

# 3. Parallelisierung „Inhärent“ Sequentieller Algorithmen

## Literatur:

Shun, J. et al., *Reducing contention through priority updates*  
id., *Sequential random permutation, list contraction and tree contraction*  
*are highly parallel*

- Abhängigkeitsgraph seq. Algorithmen
- Beispielalgorithmen
- Parallele Operationen in shared memory

# 4. Paralleles Branch-and-Bound

## Literatur:

Talbi, E.-G., *Parallel Combinatorial Optimization*, Kapitel 1

- Sequentielles Branch-and-Bound
- Parallelisierungsstrategien
- Beispiele



# 5. Paralleles Dynamic Programming

## Literatur:

Talbi, E.-G., *Parallel Combinatorial Optimization*, Kapitel 2

Lee, J. et al., *A Hypercube Algorithm for the 0/1 Knapsack Problem*

- Parallelisierungsstrategien für Dynamic Programming
- Beispiel: Knapsack Problem
- Einfluss des Kommunikationsmodells

### Literatur:

Aggarwal, A. et al., *Parallel Computational Geometry*

Miller, R. und Stout Q.F., *Efficient Parallel Convex Hull Algorithms*

- Sequentielle konvexe Hülle
- Parallelisierung des Algorithmus
- Einfluss des Kommunikationsmodells

## Literatur:

Dror Irony et al., *Communication lower bounds for distributed-memory matrix multiplication*

- Parallele Multiplikation großer Matrizen
- Kommunikation notwendig
- Es gibt untere Schranken abhängig vom Parallelitätsgrad und verfügbaren Speicher

### Literatur:

Edgar Solomonik et al., *Communication-Optimal Parallel 2.5D Matrix Multiplication and LU Factorization Algorithms*

Vipin Kumar, *Introduction to parallel computing : design and analysis of algorithms*

- Parallele Multiplikation großer Matrizen
- Speicher/Kommunikationsaufwand Tradeoff
- Praktische Implementierung mit Experimenten

## Literatur:

K. E. Batcher, *Sorting networks and their applications*

- Sortieren durch spezielle Netzwerke
- Hauptsächlich von theoretischem Interesse

## Literatur:

Guy E. Blelloch et al., *A Comparison of Sorting Algorithms for the Connection Machine CM-2*

- Sortieren auf Cluster-Computern
- Aufteilung der Arbeit zwischen den einzelnen Rechnern
- Für wirklich große Datenmengen

## Literatur:

Johannes Singler et al., *MCSTL: The Multi-Core Standard Template Library*

- Sortieren auf Mehrkernrechnern (Standard)
- Höchste praktische Relevanz
- Wie bringt man parallele Algorithmen „unters Volk“

# 12. Paralleles Optimieren per Metaheuristik

## Literatur:

Byung-Il Koh et al., *Parallel asynchronous particle swarm optimization*

- Beispiel einer Metaheuristik
- Synchrone/Asynchrone Parallelisierung
- Experimentelle Untersuchung



**Fragen?**