

Lernziele – Kompetenzen

- ▶ **eigenständiges Einarbeiten** in einen Themenbereich der theoretischen Informatik
- ▶ eine Einführung in das Thema in einem 5-minütigen **Kurzvortrag** geben
- ▶ das Thema anschaulich und gut aufbereitet in einem 35-minütigen **wissenschaftlichen Vortrag** vermitteln
- ▶ Themen der anderen Teilnehmer **aktiv diskutieren**
 - Grundfähigkeiten des wissenschaftlichen Arbeitens
 - Vorbereitung auf Präsentation der Bachelorarbeit

Anforderungen

- ▷ **eigenständiges** Einarbeiten
- ▷ Präsentation des Themas in **Kurzvortrag** (5 min) und **Hauptvortrag** (35 min)
- ▷ **Anwesenheit** an allen Terminen und Diskussionsbeteiligung
- ▷ Einhalten der gesetzten **Fristen**
- ▷ **schriftliche Ausarbeitung** von ca. 10 Seiten
 - **Benotung:** Qualität der Vorträge (Inhalt und Form),
schriftliche Ausarbeitung
Diskussionsbeteiligung

Betreuung

- ▷ Ihr Betreuer ist Ihr **Ansprechpartner** bei allen Fragen, sowohl inhaltlich als auch zum Vortrag.
- ▷ Es liegt in **Ihrer Verantwortung** auf ihn/sie zuzugehen.
- ▷ Sie erhalten **Feedback und Hilfestellungen** auf Vorabversionen des Vortrags und der Ausarbeitung.

→ **Verbindliche Treffen:**

mind. 1 Woche vor dem Kurzvortrag

mind. 2 Wochen vor dem Hauptvortrag

Folien mind. 1 Woche vor dem Hauptvortrag

Ausarbeitung 11.7. für Feedback, **22.7. finale Abgabe**

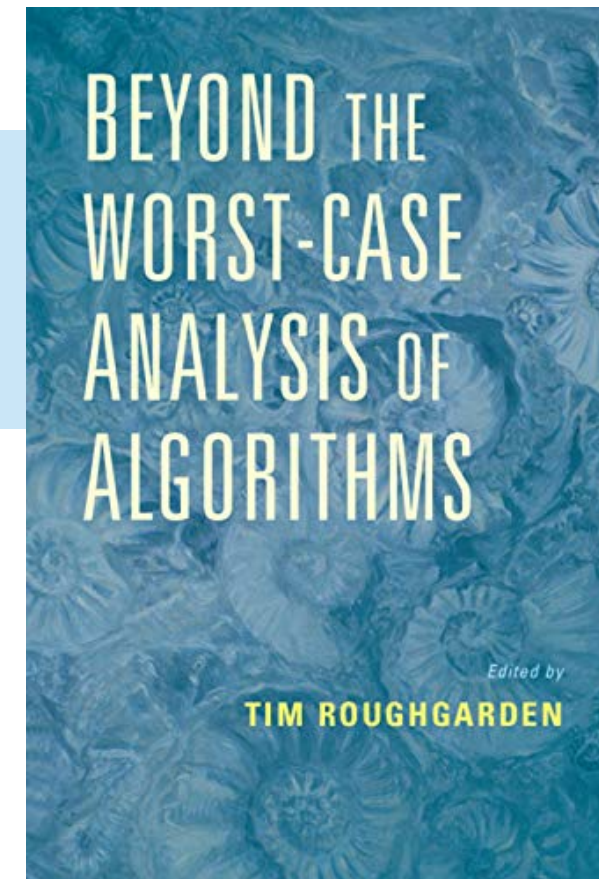
Quelle

- ▷ Wir behandeln das Buch

**“Beyond the Worst-Case Analysis
of Algorithms”**

von *Tim Roughgarden* (Hrsgb.).

- ▷ Verschiedene Aspekte der Laufzeitbetrachtung
- ▷ meist erklärt anhand von konkreten Beispielen
- ▷ ca. 10 Seiten pro Thema



→ verfügbar als E-Ressource:

<https://primo.bibliothek.kit.edu/permalink/f/coi3a3/KITSRC1745081291>

Themen

(1) Parameterized Algorithms I

- ▷ Motivation, Einführung
- ▷ formale Definitionen
- ▷ FPT Algorithmen, XP Algorithmen
- ▷ Beispiele und Anschauung!

Problem	Good news
VERTEX COVER	$\mathcal{O}(2^k \cdot k \cdot n)$ -time algorithm
CLIQUE WITH Δ	$\mathcal{O}(2^\Delta \cdot \Delta^2 \cdot n)$ -time algorithm
CLIQUE WITH k	$n^{\mathcal{O}(k)}$ -time algorithm
VERTEX COLORING WITH k	

Material: Kapitel 2.1, 2.2, Seiten 27–34

Betreuerin: Laura Merker

Themen

(2) Parameterized Algorithms II

- ▷ strukturelle Parametrisierung
- ▷ Kernelisierung
- ▷ W[1]-Schwere
- ▷ Exponential Time Hypothesis

Problem	Good news
VERTEX COLORING WITH VC	$\mathcal{O}(t^t \cdot (n + m))$ -time algorithm
VERTEX COVER WITH k	kernel of size k^2
MATRIX RANK WITH r, k	kernel of size $\mathcal{O}((n \cdot k)^2)$

Material: Kapitel 2.3–2.5, Seiten 35–42

Betreuer: Paul Jungeblut

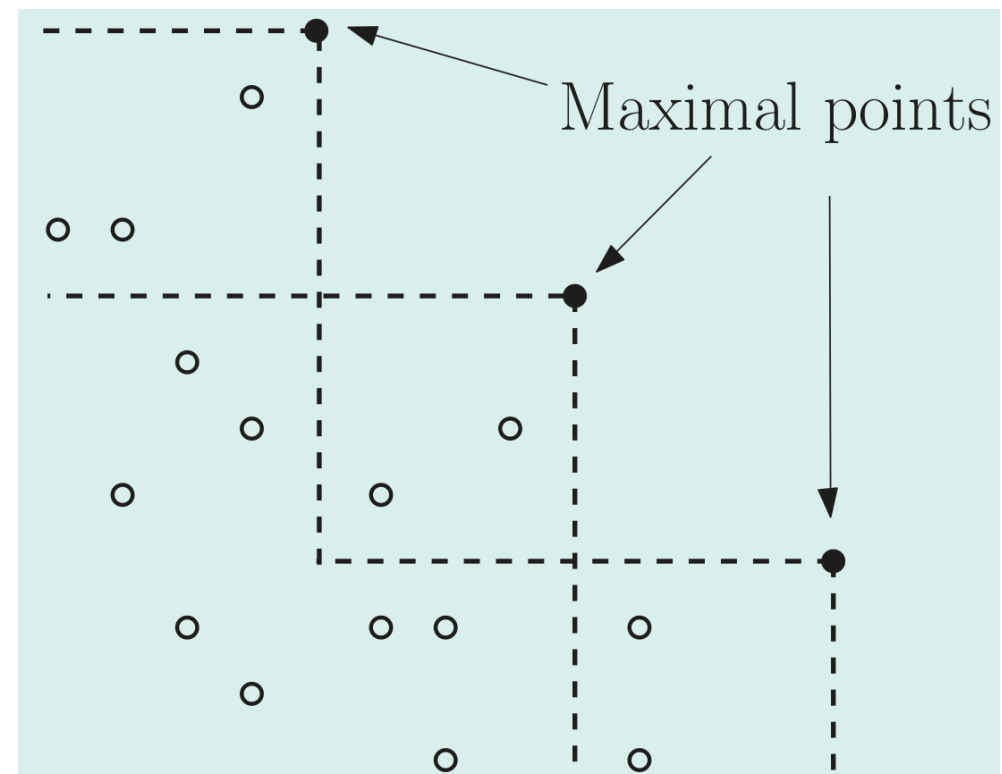
Themen

(3) Adaptive Analysis to Instance Optimality

▷ Maxima Set Problem

Gift Wrapping
Graham's Scan

▷ keiner von beiden auf allen Instanzen besser



Material: Kapitel 3.1, 3.2.1, Seiten 52–61

Betreuer: Michael Zündorf

Themen

(4) Resource Augmentation

- ▷ Online vs Offline Algorithmen
- ▷ Competitive Ratio
- ▷ unfairen Vergleich
- Handicap für Onlinealgo

In *resource augmentation* the performance of an algorithm is compared to the best possible solution that is handicapped by fewer resources. [...]
Resource augmentation bounds also imply “loosely competitive” bounds, which show that an algorithm’s performance is near optimal for most resource levels

Material: Kapitel 4.1, 4.2, 4.5, Seiten 72–76, 86–89

Betreuer: Max Göttlicher

Themen

(5) Perturbation Resilience

- ▷ Wie sehr können wir die Instanz ändern, bis sich die optimale Lösung ändert?
- ▷ Definition kombinatorisches Optimierungsproblem
- ▷ Zertifizierte Algorithmen

Loosely speaking, an instance is *perturbation resilient* if the optimal solution remains the same when we perturb the instance. [*Certified algorithms*] bridge the gap between the worst-case and structured instances: on one hand, they always find an approximate solution; on the other hand, they exactly solve perturbation-resilient instances.

Material: Kapitel 5.2, 5.2, 5.4, Seiten 95–100, 106–107

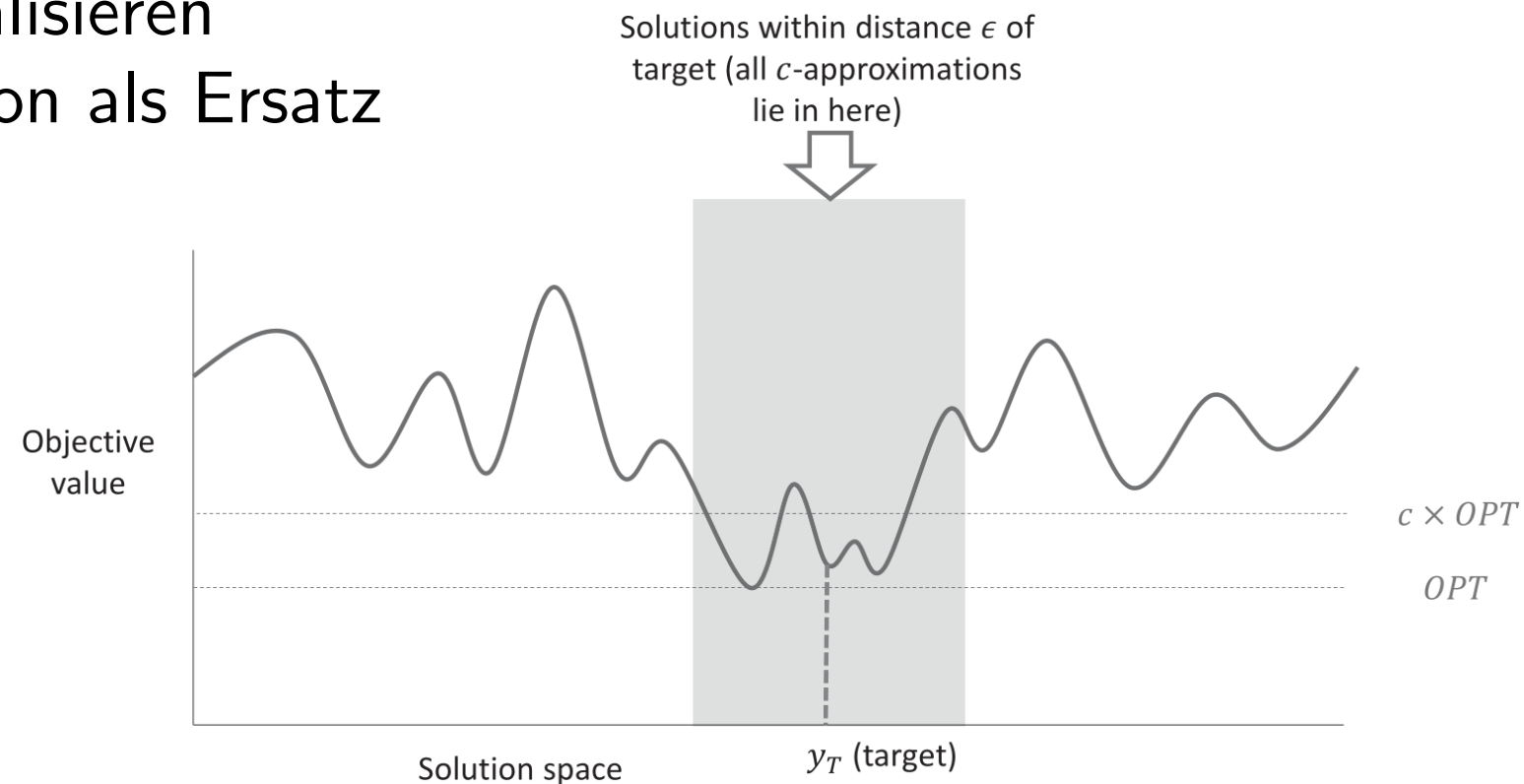
Betreuer: Adrian Feilhauer

Themen

(6) Approximation Stability

- ▶ Optimierungsfunktion manchmal schwer zu formalisieren
- Einfache Funktion als Ersatz

- ▶ Wie misst man die Qualität im eigentlichen Wert?



Material: Kapitel 6.1–6.3.2, 6.7, Seiten 120–126, 135f

Betreuer: Torsten Ueckerdt

Themen

(7) Distributional Analysis I

- ▷ Onlinealgorithmus
Optimal Stopping
- ▷ Average Case
QuickSort
Linear Probing
Bin Packing

In *distributional* or *average case analysis*, the goal is to design an algorithm with good-on-average performance with respect to a specific probability distribution.

Material: Kapitel 8.1, 8.2, Seiten 167–174

Betreuer: Thomas Bläsius

Themen

(8) Distributional Analysis II

▷ 2D-Probleme

Konvexe Hülle

Traveling Salesman

▷ Zufallsgraphen

Erdős-Rényi

Planted Bisection

Problem	Good news
CONVEX HULL	expected $O(n)$
TRAVELING SALESMAN	poly-time algo with expected good result on random input

Material: Kapitel 8.3, 8.4, Seiten 175–182

Betreuer: Marcus Wilhelm

Themen

(9) Random Order Models

- ▷ Wieder Onlinealgorithmen
 - Secretary Problem
 - Multiple Secretary Problem
- ▷ Gegenspieler darf Eingabe wählen
 - Abschwächung durch eingeschränkte Randomisierung

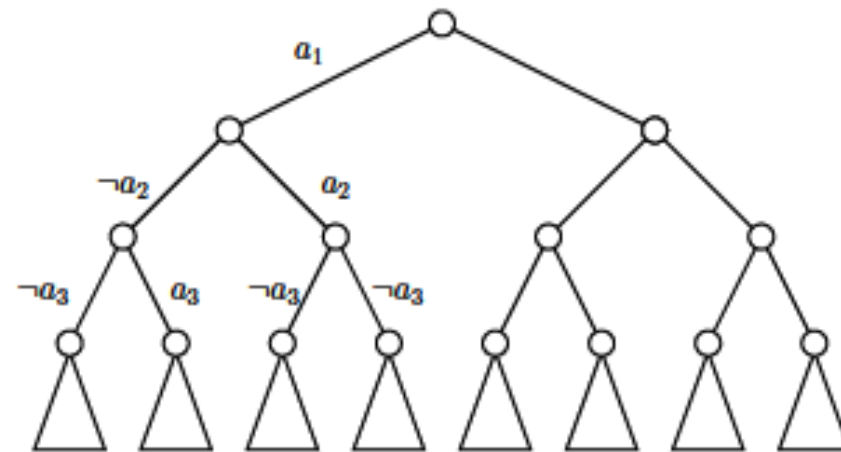
Material: Kapitel 11.1–11.3.1, Seiten 234–240

Betreuer: Christopher Weyand

Themen

(10) SAT Solver

- ▷ BOOLEAN SATISFIABILITY ist das erste NP-vollständige Problem
- ▷ Kann in der Praxis trotzdem oft schnell gelöst werden
- ▷ Techniken: Wie funktionieren SAT Solver?



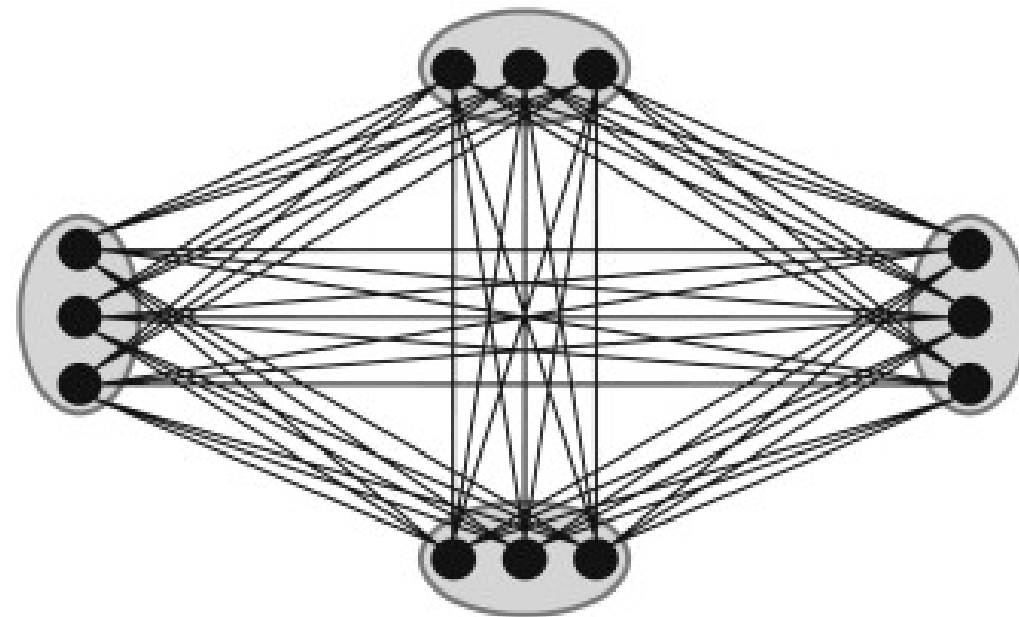
Material: Kapitel 25.1, 25.2, Seiten 547–554

Betreuer: Jonas Sauer

Themen

(11) Distribution-Free Models of Social Networks I

- ▷ Soziale Netzwerke haben besondere Struktur
- ▷ Freunde von Freunden sind oft auch miteinander befreundet
- ▷ Wie findet man schnell maximale Cliques



Material: Kapitel 28.1, 28.2, Seiten 606–612

Betreuer: Maximilian Katzmann

Themen

(12) Distribution-Free Models of Social Networks II

- ▷ Freunde von Freunden sind oft auch miteinander befreundet
- ▷ weitere Eigenschaft: Knotengrade
- ▷ *Power-Law* Verteilung
- ▷ Inspiriert davon: *Triangle Dense Graphs*



Material: Kapitel 28.3, 28.4, Seiten 612–619

Betreuer: Max Göttlicher

(1)
**Parametrisierte
Algorithmen I**
Laura

(2)
**Parametrisierte
Algorithmen II**
Paul

(3)
**Adaptive
Analysis**
Michael

(4)
**Resource
Augmentation**
Max

(5)
**Perturbation
Resilience**
Adrian

(6)
**Approximation
Stability**
Torsten

(7)
**Distributional
Analysis I**
Thomas

(8)
**Distributional
Analysis II**
Marcus

(9)
**Random-
Order Models**
Chris

(10)
SAT Solver
Jonas

(11)
**Distribution-
Free Models
of Social
Networks I**
Maximilian

(12)
**Distribution-
Free Models
of Social
Networks II**
Max

