# Approximation Algorithms

## Lecture 1:
## Introduction and Vertex Cover

### Part I:
### Organizational

Alexander Wolff                    Winter 2022/23

# Organizational

Lectures:   on site (in German :-)

Fri, 10:15–11:45 (ÜR I)

possibly some lectures via inverted classroom

Tutorials:   roughly one exercise sheet per lecture
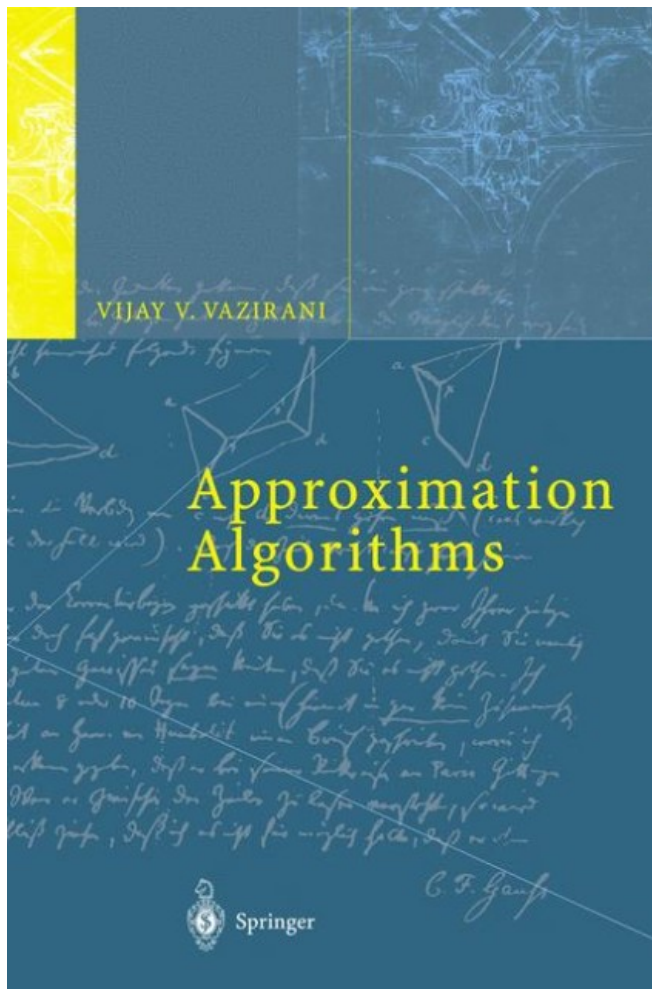
discussing old solutions and solving new tasks

Tue, 10:15–11:45 (SE I)

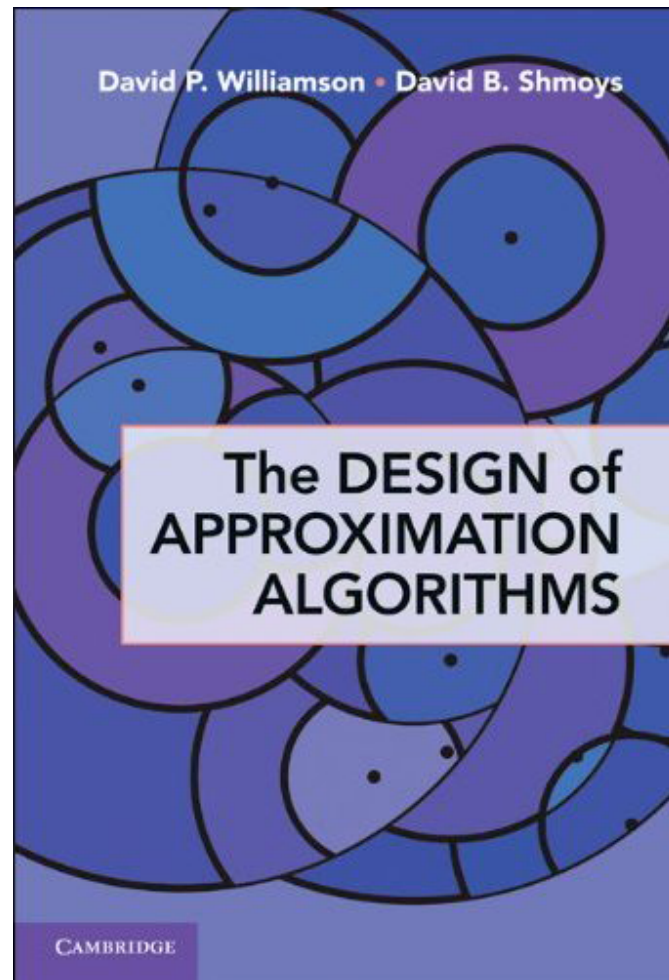Bonus ($+0.3$ on final grade) for $\geq 50\%$ points

Questions/Tasks during the lecture

Most slides are due to Joachim Spoerhase. Thanks!

# Textbooks

Vijay V. Vazirani:
Approximation Algorithms
Springer-Verlag, 2003.

D. P. Williamson & D. B. Shmoys:
The Design of Approximation Algorithms
Cambridge-Verlag, 2011.

http://www.designofapproxalgs.com/

# Approximation Algorithms

„All exact science is dominated by the idea of approximation.“

— Bertrand Russell
(1872 − 1970)

# Approximation Algorithms

- Many optimization problems are NP-hard!
  (For example, the traveling salesperson problem.)

- $\rightsquigarrow$ an optimal solution cannot be efficiently computed unless P=NP.

- However, good approximate solutions can often be found efficiently!

- **Techniques** for the design and analysis of approximation algorithms arise from studying specific optimization problems.

# Overview

## Combinatorial algorithms

- Introduction (Vertex Cover)
- Set Cover via Greedy
- Shortest Superstring via reduction to SC
- Steiner Tree via MST
- Multiway Cut via Greedy
- $k$-Center via Parametrized Pruning
- Min-Degree Spanning Tree and local search
- Knapsack via DP and Scaling
- Euclidean TSP via Quadtrees

## LP-based algorithms

- introduction to LP-Duality
- Set Cover via LP Rounding
- Set Cover via Primal–Dual Schema
- Maximum Satisfiability
- Scheduling und Extreme Point Solutions
- Steiner Forest via Primal–Dual
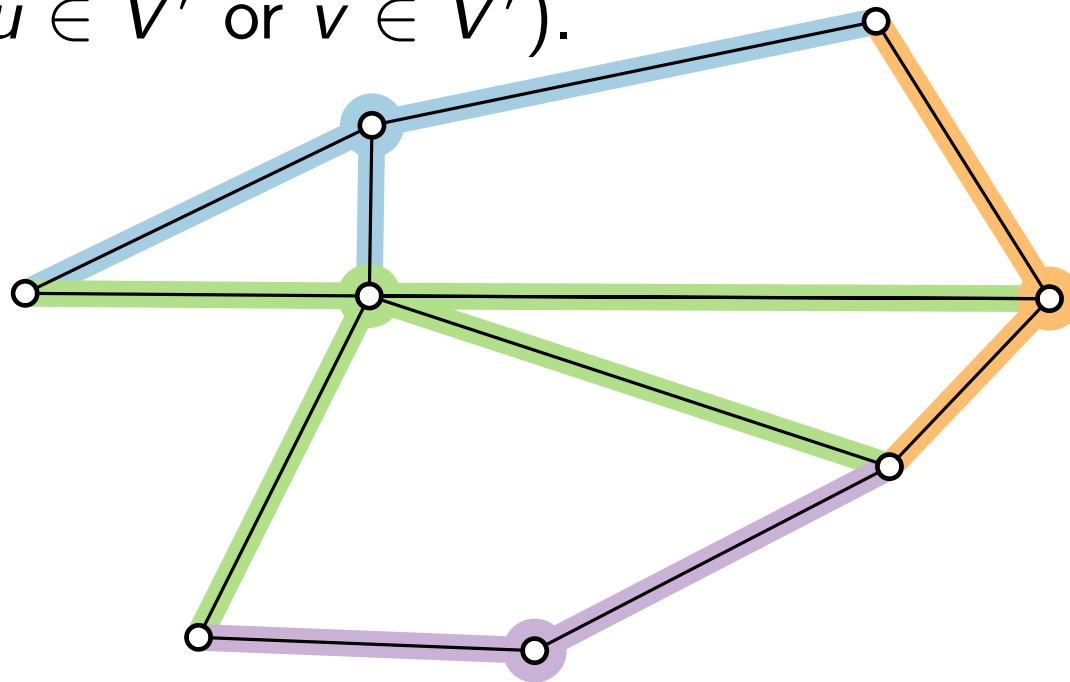
# Approximation Algorithms

## Lecture 1:
## Introduction and Vertex Cover

### Part II:
### (Cardinality) Vertex Cover

Alexander Wolff                    Winter 2022/23

# VERTEXCOVER (card.)

**Input:** Graph $G = (V, E)$

**Output:** a minimum **vertex cover**, that is,
a minimum-cardinality vertex set $V' \subseteq V$ such that
every edge is **covered** (i.e., for every $uv \in E$, it holds
that $u \in V'$ or $v \in V'$).



**Optimum** (OPT $= 4$) − but in general NP-hard to find :-(

# Approximation Algorithms

## Lecture 1:
## Introduction and Vertex Cover

### Part III:
### NP-Optimization Problem

Alexander Wolff                                    Winter 2022/23

# NP-Optimization Problem

An **NP-optimization problem** $\Pi$ is given by:

- A set $D_\Pi$ of **instances**.
  We denote the size of an instance $I \in D_\Pi$ by $|I|$.

- For each instance $I \in D_\Pi$,
  a set $S_\Pi(I) \neq \emptyset$ of **feasible solutions** for $I$ such that:
  - for each solution $s \in S_\Pi(I)$,
    its size $|s|$ is polynomially bounded in $|I|$, and
  - there is a polynomial-time algorithm that decides,
    for each pair $(s, I)$, whether $s \in S_\Pi(I)$.

- A polynomial time computable **objective function** $\mathrm{obj}_\Pi$
  which assigns a positive objective value $\mathrm{obj}_\Pi(I, s) \geq 0$ to
  any given pair $(s, I)$ with $s \in S_\Pi(I)$.

- $\Pi$ is either a minimization or maximization problem.

# VERTEXCOVER: NP-Optimization Problem

Task: Fill in the gaps for $\Pi = $ VERTEX COVER.

$D_\Pi = $ Set of all graphs

For $I \in D_\Pi$: $\qquad |I| = $ Number of vertices $|V|$

$G = (V, E)$ $\qquad S_\Pi(I) = $ Set of all vertex covers of $G$

- Why is $|s| \in \text{poly}(|I|)$ for every $s \in S_\Pi(I)$?

$$s \subseteq V \Rightarrow |s| \leq |V| = |I|$$

- For a given pair $(s, I)$, how can we efficiently decide whether $s \in S_\Pi(I)$? Test whether all edges are covered.

$\text{obj}_\Pi(I, s) = |s|$

$\Pi$ is a minimization problem.

# Optimum and Optimal Objective Value

Let $\Pi$ be a ~~minimization problem~~ maximization problem and $I \in D_\Pi$ an instance of $\Pi$.

A feasible solution $s^* \in S_\Pi(I)$ is **optimal** if $\text{obj}_\Pi(I, s^*)$ is ~~minimal~~ maximal among the objective values attained by the feasible solutions of $I$.

The optimal value $\text{obj}_\Pi(I, s^*)$ of the objective function is denoted by $\text{OPT}_\Pi(I)$ or simply by $\text{OPT}$ in context.

# Approximation Algorithms

Let $\Pi$ be a ~~minimization problem~~ and ~~$\alpha \in \mathbb{Q}^+$~~.

maximization problem $\qquad \alpha \colon \mathbb{N} \to \mathbb{Q}$

A factor-$\alpha$ approximation algorithm for $\Pi$ is an efficient algorithm that provides, for **any** instance $I \in D_\Pi$, a feasible solution $s \in S_\Pi(I)$ such that

$$\frac{\mathrm{obj}_\Pi(I, s)}{\mathrm{OPT}_\Pi(I)} \overset{\geq}{\underset{\leq}{\phantom{=}}} ~~\alpha~~. \quad \alpha(|I|)$$

# Approximation Algorithms

Lecture 1:
Introduction and Vertex Cover

Part IV:
Approximation Algorithm for VertexCover

Alexander Wolff                                        Winter 2022/23

# Approximation Alg. for VERTEXCOVER

Ideas?

- Edge-Greedy
- Vertex-Greedy
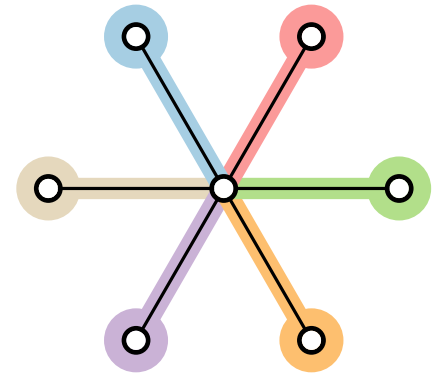


Quality?

**Problem:** How can we estimate $\mathsf{obj}_\Pi(I, s)/\mathsf{OPT}$, when it is hard to compute $\mathsf{OPT}$?

**Idea:** Find a "good" lower bound $L \leq \mathsf{OPT}$ for $\mathsf{OPT}$ and compare it to our approximate solution.

$$\frac{\mathsf{obj}_\Pi(I, s)}{\mathsf{OPT}} \leq \frac{\mathsf{obj}_\Pi(I, s)}{L}$$
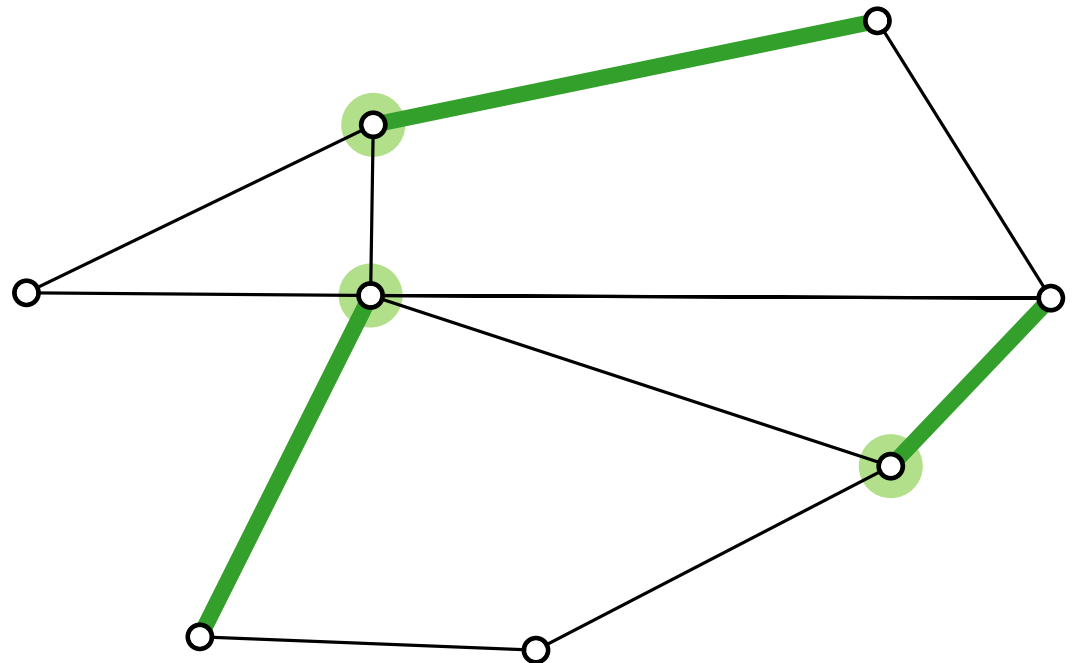
# Lower Bound by Matchings

An edge set $M \subseteq E$ of a graph $G = (V, E)$ is a **matching** if no two edges of $M$ are adjacent (i.e., share an end vertex).

$M$ is **maximal** if there is no matching $M'$ with $M' \supsetneq M$.

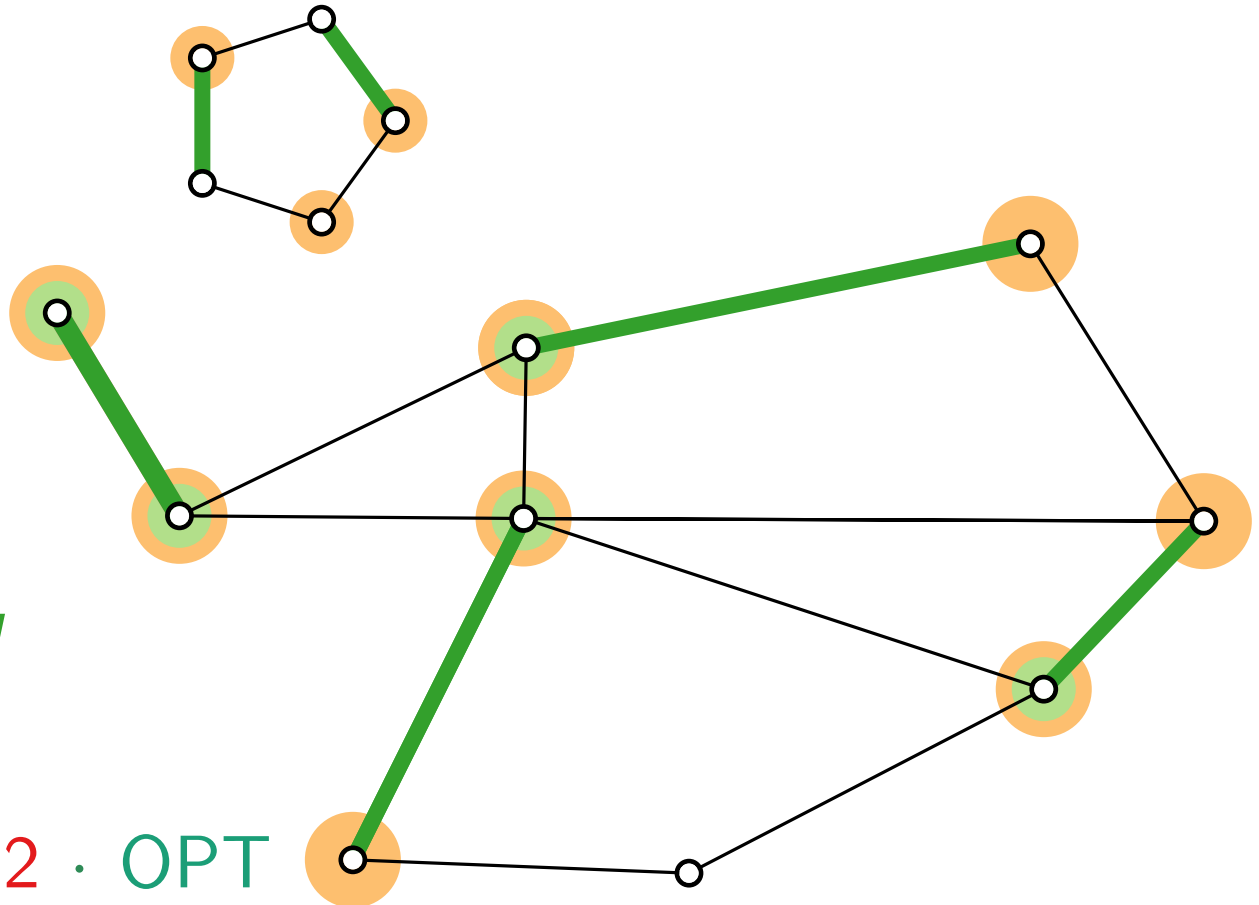OPT $\geq |M|$

Vertex cover of $M$

# Lower Bound by Matchings

Given a graph $G$, a set $M$ of edges of $G$ is a **matching**
if no two edges of $M$ are adjacent (i.e., share an end vertex).

$M$ is **maximal** if there is no matching $M'$ with $M' \supsetneq M$.



$\text{OPT} \geq |M|$

~~$\text{OPT} = |M|$~~?

Vertex cover of $M$
Vertex cover of $E$

$\text{ALG} = 2 \cdot |M| \leq 2 \cdot \text{OPT}$

# Approximation Alg. for VERTEXCOVER

Algorithm VertexCover($G$)

  $M \leftarrow \emptyset$

  **foreach** $e \in E(G)$ **do**

     **if** $e$ is not adjacent to any edge in $M$ **then**

       $M \leftarrow M \cup \{e\}$

  **return** $\{\, u, v \mid uv \in M \,\}$

**Theorem.** The above algorithm is a factor-2 approximation algorithm for VERTEXCOVER.

# Approximability of Vertex Cover

The best known approximation factor for VERTEXCOVER is
$$2 - \Theta(1/\sqrt{\log n}).$$

If P $\neq$ NP, VERTEXCOVER cannot be approximated within a factor of 1.3606.

VERTEXCOVER cannot be approximated within a factor of $2 - \Theta(1)$ – if the *Unique Games Conjecture* holds.