Lecture: Approximation Algorithms

Jannik Matuschke

TUM

October 22, 2018

Schedule

Lectures (Jannik Matuschke)

- Mon 12:15-13:45 in MI 03.10.011
- Wed 12:15-13:45 in MI 03.10.011

from Oct 22 to Dec 5

Tutorials (Marcus Kaiser)

A Tue 16:00-17:30 in MI 03.08.011

B Wed 16:00-17:30 in MI 02.08.020

starting next week

registration from today 18:00
until tomorrow 23:59

Basic information

Problem sets

- published on Monday
- discussed the week after publication
- bonus for presenting solutions in tutorial

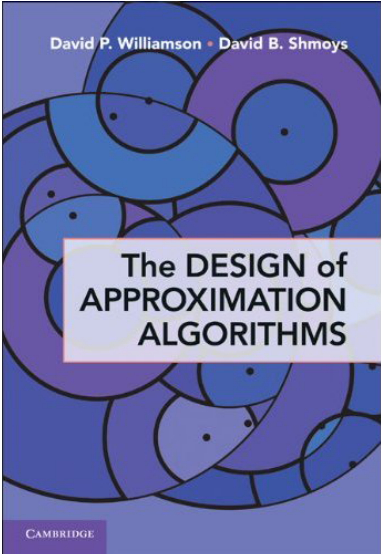
Exam

- first round in December
- retake in January

additional information & course materials:

`http://www.or.tum.de/en/teaching/winter2018/approxalgorithms/`

The book


David P. Williamson • David B. Shmoys
The DESIGN of APPROXIMATION ALGORITHMS
CAMBRIDGE

Introduction to Approximation Algorithms

Algorithmic wishlist

1. fast (run in polynomial time)
2. universal (work for any instance)
3. optimal (find best solution)

Algorithmic wishlist

1. fast (run in polynomial time)
2. universal (work for any instance)
3. optimal (find best solution)

Choose two.
(unless P = NP)

Algorithmic wishlist

1. fast (run in polynomial time)
2. universal (work for any instance)
3. approximately optimal (find provably good solution)

$\Downarrow$

Approximation Algorithms

Approximation Algorithms

Definition An α-approximation algorithm for an optimization problem is an algorithm that

- runs in polynomial time and
- computes for any instance of the problem a solution
- whose value is within a factor of α of the optimal solution.

Approximation Algorithms

Definition An α-approximation algorithm for an optimization problem is an algorithm that

- runs in polynomial time and
- computes for any instance of the problem a solution
- whose value is within a factor of α of the optimal solution.

ALG: value of solution computed by algorithm
OPT: value of optimal solution

for maximization problems:

$$\text{ALG} \geq \alpha \cdot \text{OPT}$$

$(\alpha \leq 1)$

for minimization problems:

$$\text{ALG} \leq \alpha \cdot \text{OPT}$$

$(\alpha \geq 1)$

Approximation Algorithms

Definition An α-approximation algorithm for an optimization problem is an algorithm that

- runs in polynomial time and
- computes for any instance of the problem a solution
- whose value is within a factor of α of the optimal solution.

ALG: value of solution computed by algorithm
OPT: value of optimal solution

for maximization problems:

$$\text{ALG} \geq \alpha \cdot \text{OPT}$$

$$(\alpha \leq 1)$$

for minimization problems:

$$\text{ALG} \leq \alpha \cdot \text{OPT}$$

$$(\alpha \geq 1)$$

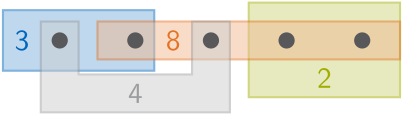
We call α approximation factor or performance guarantee.

Example: Set Cover

The SET COVER problem

Input: elements E, sets $\mathcal{F} \subseteq 2^E$, weights $w : \mathcal{F} \to \mathbb{R}_+$

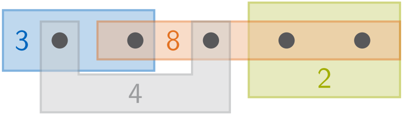
Task: find $\mathcal{F}' \subseteq \mathcal{F}$ with $\bigcup_{S \in \mathcal{F}'} S = E$
minimizing $\sum_{S \in \mathcal{F}'} w(S)$



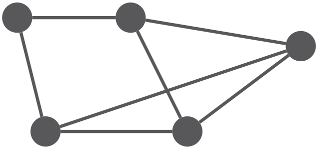
The Set Cover problem

Input: elements E, sets $\mathcal{F} \subseteq 2^E$, weights $w : \mathcal{F} \to \mathbb{R}_+$

Task: find $\mathcal{F}' \subseteq \mathcal{F}$ with $\bigcup_{S \in \mathcal{F}'} S = E$
minimizing $\sum_{S \in \mathcal{F}'} w(S)$



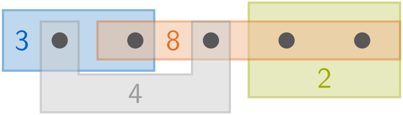
Special case: Vertex Cover



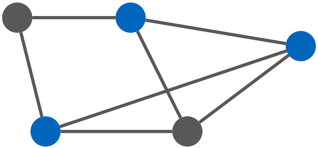
The Set Cover problem

Input: elements E, sets $\mathcal{F} \subseteq 2^E$, weights $w : \mathcal{F} \to \mathbb{R}_+$

Task: find $\mathcal{F}' \subseteq \mathcal{F}$ with $\bigcup_{S \in \mathcal{F}'} S = E$
minimizing $\sum_{S \in \mathcal{F}'} w(S)$



Special case: Vertex Cover



How to design an approximation algorithm?

How to design an approximation algorithm?

We don't know OPT,
but we can get lower bounds.

IP formulation

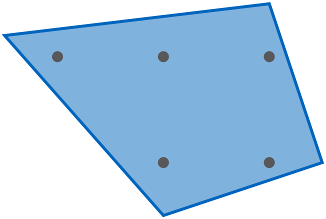
$$\begin{aligned} \text{min} \quad & \sum_{S \in \mathcal{F}} w(S)x(S) & \\ \text{s.t.} \quad & \sum_{S: e \in S} x(S) \geq 1 & \forall\, e \in E \\ & x(S) \in \{0, 1\} & \forall S \in \mathcal{F} \end{aligned}$$

LP relaxation

$$\begin{aligned} \min \quad & \sum_{S \in \mathcal{F}} w(S) x(S) & \\ \text{s.t.} \quad & \sum_{S \in \mathcal{F} : e \in S} x(S) \geq 1 & \forall\, e \in E \\ & x(S) \geq 0 & \forall S \in \mathcal{F} \end{aligned}$$

LP relaxation

$$
\begin{aligned}
\min \quad & \sum_{S \in \mathcal{F}} w(S)x(S) & \\
\text{s.t.} \quad & \sum_{S \in \mathcal{F} : e \in S} x(S) \geq 1 & \forall\, e \in E \\
& x(S) \geq 0 & \forall S \in \mathcal{F}
\end{aligned}
$$

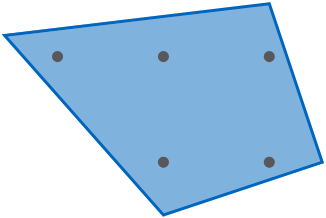


LP relaxation

$$
\begin{aligned}
Z^* \; := \; \min \quad & \sum_{S \in \mathcal{F}} w(S) x(S) & \\
\text{s.t.} \quad & \sum_{S \in \mathcal{F}: e \in S} x(S) \geq 1 & \forall\, e \in E \\
& x(S) \geq 0 & \forall S \in \mathcal{F}
\end{aligned}
$$

LP value is lower bound:

$$Z^* \leq \mathrm{OPT}$$



(Deterministic) LP Rounding

LP rounding

Idea: Select S if $x(S) \geq \frac{1}{f}$. $\qquad f := \max_{e \in E} |\{S \in \mathcal{F} : e \in S\}|$

LP rounding

Idea: Select S if $x(S) \geq \frac{1}{f}$.

$f := \max_{e \in E} |\{S \in \mathcal{F} : e \in S\}|$

Theorem 1.1

LP rounding is an f-approximation algorithm for SET COVER.

LP rounding

Idea: Select S if $x(S) \geq \frac{1}{f}$.

$f := \max_{e \in E} |\{S \in \mathcal{F} : e \in S\}|$

Theorem 1.1

LP rounding is an f-approximation algorithm for Set Cover.

Proof.

$$\mathcal{F}' := \left\{ S \in \mathcal{F} : x(S) \geq \frac{1}{f} \right\}$$

LP rounding

Idea: Select S if $x(S) \geq \frac{1}{f}$.

$f := \max_{e \in E} |\{S \in \mathcal{F} : e \in S\}|$

Theorem 1.1

LP rounding is an f-approximation algorithm for Set Cover.

Proof.

$$\mathcal{F}' := \left\{ S \in \mathcal{F} : x(S) \geq \frac{1}{f} \right\}$$

- Is every element covered?

LP rounding

Idea: Select S if $x(S) \geq \frac{1}{f}$.

$$f := \max_{e \in E} |\{S \in \mathcal{F} : e \in S\}|$$

Theorem 1.1

LP rounding is an f-approximation algorithm for SET COVER.

Proof.

$$\mathcal{F}' := \left\{ S \in \mathcal{F} : x(S) \geq \frac{1}{f} \right\}$$

- Is every element covered?

$$\sum_{S \in \mathcal{F} : e \in S} x(S) \geq 1 \quad \Rightarrow \quad \exists\, S \in \mathcal{F} :\ x(S) \geq \frac{1}{f}$$

LP rounding

Idea: Select S if $x(S) \geq \frac{1}{f}$. $f := \max_{e \in E} |\{S \in \mathcal{F} : e \in S\}|$

Theorem 1.1

LP rounding is an f-approximation algorithm for Set Cover.

Proof.

$$\mathcal{F}' := \left\{ S \in \mathcal{F} : x(S) \geq \frac{1}{f} \right\}$$

- Is every element covered?

  $$\sum_{S \in \mathcal{F} : e \in S} x(S) \geq 1 \quad \Rightarrow \quad \exists\, S \in \mathcal{F} : \; x(S) \geq \frac{1}{f}$$

- Is the approximation factor fulfilled?

LP rounding

Idea: Select S if $x(S) \geq \frac{1}{f}$. $\quad f := \max_{e \in E} |\{S \in \mathcal{F} : e \in S\}|$

Theorem 1.1

LP rounding is an f-approximation algorithm for SET COVER.

Proof. $\quad \mathcal{F}' := \left\{ S \in \mathcal{F} : x(S) \geq \frac{1}{f} \right\}$

- Is every element covered?

$$\sum_{S \in \mathcal{F} : e \in S} x(S) \geq 1 \quad \Rightarrow \quad \exists\, S \in \mathcal{F} :\ x(S) \geq \frac{1}{f}$$

- Is the approximation factor fulfilled?

$$\sum_{S \in \mathcal{F}'} w(S) \ \leq \ \sum_{S \in \mathcal{F}'} w(S) \cdot f \cdot x(S)$$

LP rounding

Idea: Select S if $x(S) \geq \frac{1}{f}$. $\qquad f := \max_{e \in E} |\{S \in \mathcal{F} : e \in S\}|$

Theorem 1.1

LP rounding is an f-approximation algorithm for SET COVER.

Proof. $\qquad \mathcal{F}' := \left\{ S \in \mathcal{F} : x(S) \geq \frac{1}{f} \right\}$

- Is every element covered?

$$\sum_{S \in \mathcal{F} : e \in S} x(S) \geq 1 \quad \Rightarrow \quad \exists\, S \in \mathcal{F} : \; x(S) \geq \frac{1}{f}$$

- Is the approximation factor fulfilled?

$$\sum_{S \in \mathcal{F}'} w(S) \;\leq\; \sum_{S \in \mathcal{F}'} w(S) \cdot f \cdot x(S) \;\leq\; f \cdot \sum_{S \in \mathcal{F}} w(S) x(S) \;=\; f \cdot Z^*$$

□

A fortiori guarantee

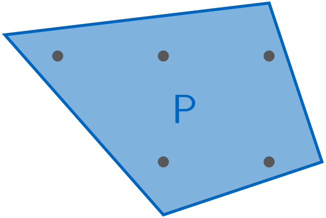
- The LP rounding analysis gives us an a priori guarantee: $\text{ALG} \leq f \cdot \text{OPT}$ for any instance of $\textsc{Set Cover}$.
- For a concrete run of the algorithm, we get an a fortiori guarantee from Z^*:

$$\frac{\text{ALG}}{\text{OPT}} \leq \frac{\text{ALG}}{Z^*}$$

(we know ALG and Z^*)

Integrality gap

$$\text{OPT} = \min\{c^t x : x \in P \cap \mathbb{Z}^E\}$$
$$Z^* = \min\{c^t x : x \in P\}$$



The integrality gap of an LP is the ratio $\dfrac{\text{OPT}}{Z^*}$.

The LP rounding algorithm implies that the integrality gap of the SET COVER LP is bounded by f:

$$\text{OPT} \leq \text{ALG} \leq f \cdot Z^*$$

The Primal-Dual Method

Primal-dual method

$$\begin{aligned} \max \quad & \sum_{e \in E} y(e) & \\ \text{s.t.} \quad & \sum_{e \in S} y(e) \leq w(S) & \forall\, S \in \mathcal{F} \\ & y(e) \geq 0 & \forall e \in E \end{aligned}$$

Primal-dual method

$$\begin{aligned} \max \quad & \sum_{e \in E} y(e) & \\ \text{s.t.} \quad & \sum_{e \in S} y(e) \leq w(S) & \forall\, S \in \mathcal{F} \\ & y(e) \geq 0 & \forall e \in E \end{aligned}$$

Algorithm:

1. Initialize $y(e) = 0$ for all $e \in E$.
2. while ($\exists$ uncovered element e)
   Increase $y(e)$ until a set S with $e \in S$ becomes tight.
   Add S to $\mathcal{F}'$. $\quad (\sum_{e \in S} y(e) = w(S))$
3. Return $\mathcal{F}'$.

Primal-dual method

$$\begin{aligned} \max \quad & \sum_{e \in E} y(e) & \\ \text{s.t.} \quad & \sum_{e \in S} y(e) \leq w(S) & \forall\, S \in \mathcal{F} \\ & y(e) \geq 0 & \forall e \in E \end{aligned}$$

Algorithm:

1. Initialize $y(e) = 0$ for all $e \in E$.
2. while ($\exists$ uncovered element e)
   Increase $y(e)$ until a set S with $e \in S$ becomes tight.
   Add S to $\mathcal{F}'$. $\qquad (\sum_{e \in S} y(e) = w(S))$
3. Return $\mathcal{F}'$.

Theorem 1.2

Primal-dual is an f-approximation algorithm for SET COVER.

Greedy algorithm

Algorithm:

1. while ($\exists$ uncovered element)
   Choose S' minimizing $\frac{w(S')}{|S' \setminus \bigcup_{S \in \mathcal{F}'} S|}$
   Add S' to $\mathcal{F}'$.
2. Return $\mathcal{F}'$.

Greedy algorithm

Algorithm:

1. while ($\exists$ uncovered element)
   Choose S' minimizing $\frac{w(S')}{|S' \setminus \bigcup_{S \in \mathcal{F}'} S|}$
   Add S' to $\mathcal{F}'$.
2. Return $\mathcal{F}'$.

$n := |E|$, $H_n := \sum_{i=1}^{n} \frac{1}{i}$

Theorem 1.3

The Greedy Algorithm is an H_n-approximation for Set Cover.

Greedy algorithm

Algorithm:

1. while ($\exists$ uncovered element)
   Choose S' minimizing $\frac{w(S')}{|S' \setminus \bigcup_{S \in \mathcal{F}'} S|}$
   Add S' to $\mathcal{F}'$.
2. Return $\mathcal{F}'$.

$n := |E|$, $H_n := \sum_{i=1}^{n} \frac{1}{i}$

Theorem 1.3

The Greedy Algorithm is an H_n-approximation for SET COVER.

S_i: set selected in iteration i

n_i: uncovered elements at start of iteration i

Greedy algorithm

Algorithm:

1. while ($\exists$ uncovered element)
   Choose S' minimizing $\frac{w(S')}{|S' \setminus \bigcup_{S \in \mathcal{F}'} S|}$
   Add S' to $\mathcal{F}'$.
2. Return $\mathcal{F}'$.

$n := |E|$, $H_n := \sum_{i=1}^{n} \frac{1}{i}$

Theorem 1.3

The Greedy Algorithm is an H_n-approximation for SET COVER.

S_i: set selected in iteration i

n_i: uncovered elements at start of iteration i

Lemma 1.4

For every iteration i: $$w(S_i) \leq \frac{n_i - n_{i+1}}{n_i} \mathrm{OPT}\,.$$

Greedy algorithm

Algorithm:

1. while ($\exists$ uncovered element)

   Choose S' minimizing $\frac{w(S')}{|S' \setminus \bigcup_{S \in \mathcal{F}'} S|}$

   Add S' to $\mathcal{F}'$.
2. Return $\mathcal{F}'$.

Theorem 1.5

$\sum_{S \in \mathcal{F}'} w(S) \leq H_g \cdot Z^*$, where $g := \max_{S \in \mathcal{F}} |S|$.