Technique 1: Round the LP solution.

We first solve the LP-relaxation and then we round the fractional values so that we obtain an integral solution.

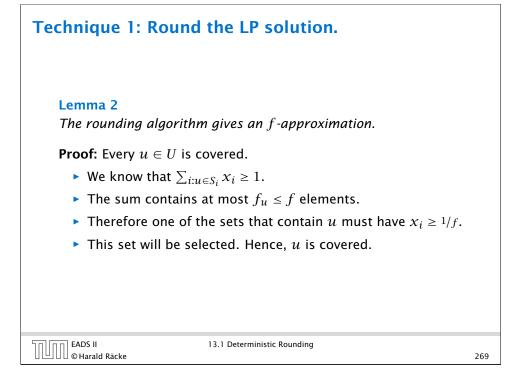
Set Cover relaxation:

min		$\sum_{i=1}^k w_i x_i$		
s.t.	$\forall u \in U$	$\sum_{i:u\in S_i} x_i$	\geq	1
	$\forall i \in \{1, \dots, k\}$	x_i	\in	[0,1]

Let f_u be the number of sets that the element u is contained in (the frequency of u). Let $f = \max_u \{f_u\}$ be the maximum frequency.

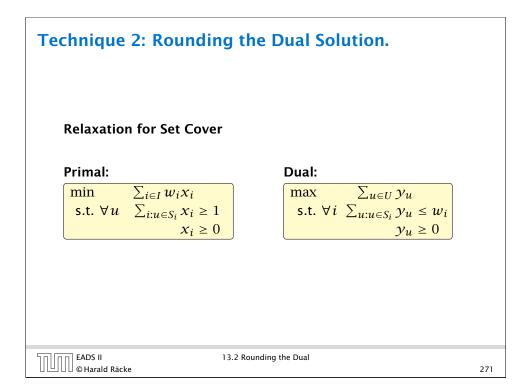
EADS II © Harald Räcke

267



EXAMPLIANCE IF A CONTACT OF A

Technique 1: Round t	he LP solution.	
$\sum r$	solution is at most $f \cdot \text{OPT}$. $w_i \leq \sum_{i=1}^k w_i (f \cdot x_i)$ $= f \cdot \text{cost}(x)$ $\leq f \cdot \text{OPT}$.	
EADS II 13 © Harald Räcke	3.1 Deterministic Rounding	270



Technique 2: Rounding the Dual Solution.

Lemma 3

The resulting index set is an f-approximation.

Proof:

Every $u \in U$ is covered.

- Suppose there is a *u* that is not covered.
- This means $\sum_{u:u\in S_i} y_u < w_i$ for all sets S_i that contain u.
- But then y_u could be increased in the dual solution without violating any constraint. This is a contradiction to the fact that the dual solution is optimal.

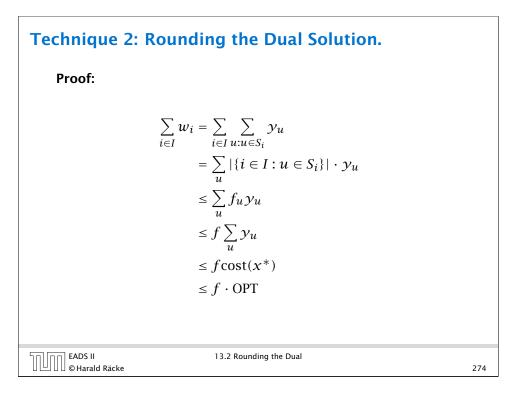
Technique 2: Rounding the Dual Solution.

Rounding Algorithm:

Let I denote the index set of sets for which the dual constraint is tight. This means for all $i \in I$

$$\sum_{u:u\in S_i} y_u = w_i$$

	13.2 Rounding the Dual	
UUU© Harald Räcke		272



273

Let I denote the solution obtained by the first rounding algorithm and I' be the solution returned by the second algorithm. Then

 $I\subseteq I'$.

This means I' is never better than I.

- Suppose that we take S_i in the first algorithm. I.e., $i \in I$.
- This means $x_i \ge \frac{1}{f}$.

EADS II

∐||||| © Harald Räcke

- Because of Complementary Slackness Conditions the corresponding constraint in the dual must be tight.
- Hence, the second algorithm will also choose S_i .

	13.2 Rounding the Dual	
🛛 💾 🔲 🖾 📽 Harald Räcke		275

Technique 3: The Primal Dual Method
Algorithm 1 PrimalDual
1: $y \leftarrow 0$
$2: I \leftarrow \emptyset$
3: while exists $u \notin \bigcup_{i \in I} S_i$ do
4: increase dual variable y_i until constraint for some
new set S_{ℓ} becomes tight
5: $I \leftarrow I \cup \{\ell\}$

13.3 Primal Dual Technique

Technique 3: The Primal Dual Method

The previous two rounding algorithms have the disadvantage that it is necessary to solve the LP. The following method also gives an f-approximation without solving the LP.

For estimating the cost of the solution we only required two properties.

1. The solution is dual feasible and, hence,

 $\sum_{u} y_{u} \le \operatorname{cost}(x^{*}) \le \operatorname{OPT}$

where x^* is an optimum solution to the primal LP.

2. The set *I* contains only sets for which the dual inequality is tight.

Of course, we also need that *I* is a cover.

הם הר	EADS II © Harald Räcke
	© Harald Räcke

13.3 Primal Dual Technique

echniqu	e 4: The Greedy Algorithm
-	rithm 1 Greedy
1: I	$\leftarrow \emptyset$
2: Ś	$j \leftarrow S_j$ for all j
3: V	vhile I not a set cover do
4:	$ \begin{split} \ell &\leftarrow \arg \min_{j: \hat{S}_j \neq 0} \frac{w_j}{ \hat{S}_j } \\ I &\leftarrow I \cup \{\ell\} \\ \hat{S}_j &\leftarrow \hat{S}_j - S_\ell \text{ for all } j \end{split} $
5:	$I \leftarrow I \cup \{\ell\}$
6:	$\hat{S}_j \leftarrow \hat{S}_j - S_\ell$ for all j

In every round the Greedy algorithm takes the set that covers remaining elements in the most cost-effective way.

We choose a set such that the ratio between cost and still uncovered elements in the set is minimized.

277

276

Technique 4: The Greedy Algorithm

Lemma 4

Given positive numbers a_1, \ldots, a_k and b_1, \ldots, b_k then

$$\min_{i} \frac{a_i}{b_i} \le \frac{\sum_{i} a_i}{\sum_{i} b_i} \le \max_{i} \frac{a_i}{b_i}$$

הח EADS II 13.4 Greedy	
UUU G Harald Räcke	279

Technique 4: The Greedy Algorithm Adding this set to our solution means $n_{\ell+1} = n_{\ell} - |\hat{S}_j|$. $w_j \leq \frac{|\hat{S}_j|\text{OPT}}{n_{\ell}} = \frac{n_{\ell} - n_{\ell+1}}{n_{\ell}} \cdot \text{OPT}$ Image: EADS II WHARAID Räcke

281

Technique 4: The Greedy Algorithm

Let n_{ℓ} denote the number of elements that remain at the beginning of iteration ℓ . $n_1 = n = |U|$ and $n_{s+1} = 0$ if we need s iterations.

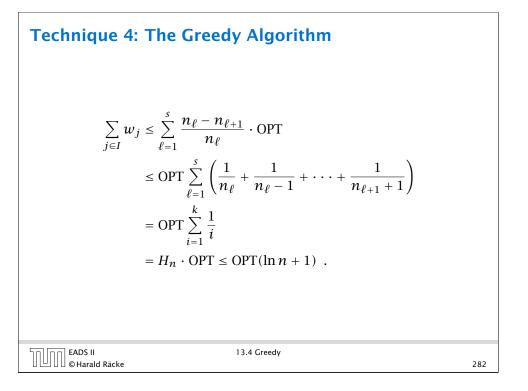
In the ℓ -th iteration

$$\min_{j} \frac{w_{j}}{|\hat{S}_{j}|} \leq \frac{\sum_{j \in \text{OPT}} w_{j}}{\sum_{j \in \text{OPT}} |\hat{S}_{j}|} = \frac{\text{OPT}}{\sum_{j \in \text{OPT}} |\hat{S}_{j}|} \leq \frac{\text{OPT}}{n_{\ell}}$$

since an optimal algorithm can cover the remaining n_ℓ elements with cost OPT.

Let \hat{S}_j be a subset that minimizes this ratio. Hence, $w_j/|\hat{S}_j| \leq \frac{\text{OPT}}{n_\ell}$.

	13.4 Greedy	
UUU GHarald Räcke		280



Technique 5: Randomized Rounding

One round of randomized rounding: Pick set S_i uniformly at random with probability $1 - x_i$ (for all *j*).

Version A: Repeat rounds until you have a cover.

Version B: Repeat for *s* rounds. If you have a cover STOP. Otherwise, repeat the whole algorithm.

EADS II © Harald Räcke EADS II 13.5 Randomized Rounding

 $\Pr[\exists u \in U \text{ not covered after } \ell \text{ round}]$

- = $\Pr[u_1 \text{ not covered} \lor u_2 \text{ not covered} \lor \dots \lor u_n \text{ not covered}]$
- $\leq \sum \Pr[u_i ext{ not covered after } \ell ext{ rounds}] \leq n e^{-\ell}$.

Lemma 5

EADS II

U]]] @ Harald Räcke

With high probability $O(\log n)$ rounds suffice.

With high probability:

For any constant α the number of rounds is at most $O(\log n)$ with probability at least $1 - n^{-\alpha}$.

13.5 Randomized Rounding

Pr[*u* not covered in one round] $= \prod_{j: u \in S_j} (1 - x_j) \le \prod_{j: u \in S_j} e^{-x_j}$ $=e^{-\sum_{j:u\in S_j} x_j} \le e^{-1}$.

Probability that $u \in U$ is not covered (after ℓ rounds):

Probability that $u \in U$ is not covered (in one round):

$$\Pr[u \text{ not covered after } \ell \text{ round}] \leq \frac{1}{e^{\ell}}$$

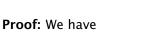
EADS II

©Harald Räcke

283

285

13.5 Randomized Rounding



 $\Pr[\#rounds \ge (\alpha + 1) \ln n] \le ne^{-(\alpha + 1) \ln n} = n^{-\alpha}$.

13.5 Randomized Rounding

284

Expected Cost

Version A. Repeat for $s = (\alpha + 1) \ln n$ rounds. If you don't have a cover simply take all sets.

 $E[\cos t] \le (\alpha + 1) \ln n \cdot \cot(LP) + (\sum_{i} w_{j}) n^{-\alpha} = \mathcal{O}(\ln n) \cdot \text{OPT}$

If the weights are polynomially bounded (smallest weight is 1), sufficiently large α and OPT at least 1.

	13.5 Randomized Rounding	
UUU GHarald Räcke	5	287

Randomized rounding gives an $O(\log n)$ approximation. The running time is polynomial with high probability.

Theorem 6 (without proof)

There is no approximation algorithm for set cover with approximation guarantee better than $\frac{1}{2}\log n$ unless NP has quasi-polynomial time algorithms (algorithms with running time $2^{\operatorname{poly}(\log n)}$).

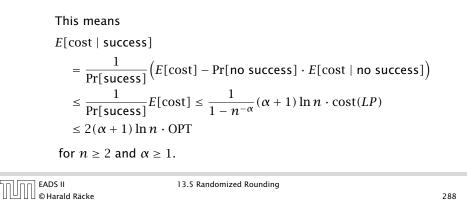
Expected Cost

Version B.

Repeat for $s = (\alpha + 1) \ln n$ rounds. If you don't have a cover simply repeat the whole process.

 $E[\text{cost}] = \Pr[\text{success}] \cdot E[\text{cost} | \text{success}]$

+ $\Pr[no success] \cdot E[cost | no success]$



Techniques:

- Deterministic Rounding
- Rounding of the Dual
- Primal Dual
- Greedv

EADS II

© Harald Räcke

- Randomized Rounding
- Local Search
- Rounding the Data + Dynamic Programming

EADS II UIII © Harald Räcke

13.5 Randomized Rounding

13.5 Randomized Rounding