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

#### Set Cover relaxation:

$$\begin{array}{|c|c|c|c|c|}\hline \min & & \sum_{i=1}^{k} w_i x_i \\ \text{s.t.} & \forall u \in U & \sum_{i:u \in S_i} x_i \geq 1 \\ & \forall i \in \{1, \dots, k\} & x_i \in [0, 1] \end{array}$$

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.



#### Rounding Algorithm:

Set all  $x_i$ -values with  $x_i \ge \frac{1}{f}$  to 1. Set all other  $x_i$ -values to 0.



#### Lemma 2

The rounding algorithm gives an f-approximation.

**Proof:** Every  $u \in U$  is covered.

- We know that  $\sum_{i:u\in S_i} x_i \ge 1$ .
- The sum contains at most  $f_u \leq f$  elements.
- Therefore one of the sets that contain u must have  $x_i \ge 1/f$ .
- ► This set will be selected. Hence, *u* is covered.



The cost of the rounded solution is at most  $f \cdot \text{OPT}$ .

$$\sum_{i \in I} w_i \leq \sum_{i=1}^k w_i (f \cdot x_i)$$
$$= f \cdot \operatorname{cost}(x)$$
$$\leq f \cdot \operatorname{OPT} .$$



#### **Relaxation for Set Cover**

#### Primal:

 $\begin{array}{|c|c|c|} \min & \sum_{i \in I} w_i x_i \\ \text{s.t. } \forall u & \sum_{i: u \in S_i} x_i \ge 1 \\ & x_i \ge 0 \end{array}$ 

#### Dual:

$$\begin{array}{c|c}
\max & \sum_{u \in U} \mathcal{Y}_{u} \\
\text{s.t. } \forall i & \sum_{u:u \in S_{i}} \mathcal{Y}_{u} \leq w_{i} \\
\mathcal{Y}_{u} \geq 0
\end{array}$$



#### **Rounding Algorithm:**

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

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



#### 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.



#### **Proof:**

$$\sum_{i \in I} w_i = \sum_{i \in I} \sum_{u: u \in S_i} y_u$$
$$= \sum_u |\{i \in I : u \in S_i\}| \cdot y_u$$
$$\leq \sum_u f_u y_u$$
$$\leq f \sum_u y_u$$
$$\leq f \operatorname{cost}(x^*)$$
$$\leq f \cdot \operatorname{OPT}$$



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  $x_i \ge \frac{1}{f}$ .
- Because of Complementary Slackness Conditions the corresponding constraint in the dual must be tight.
- ► Hence, the second algorithm will also choose *S*<sub>*i*</sub>.



# **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.

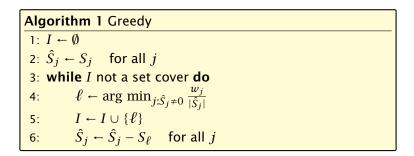
Of course, we also need that *I* is a cover.



# **Technique 3: The Primal Dual Method**

Algorithm 1 PrimalDual
$\begin{array}{c} 1: \ \mathcal{Y} \leftarrow 0\\ 2: \ I \leftarrow \emptyset \end{array}$
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 $S_\ell$ becomes tight
5: $I \leftarrow I \cup \{\ell\}$





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.



#### Lemma 4

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

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



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

In the  $\ell$ -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  $n_\ell$  elements with cost OPT.

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

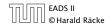


Adding this set to our solution means  $n_{\ell+1} = n_{\ell} - |\hat{S}_j|$ .

$$w_j \leq \frac{|\hat{S}_j|\text{OPT}}{n_\ell} = \frac{n_\ell - n_{\ell+1}}{n_\ell} \cdot \text{OPT}$$



$$\sum_{j \in I} w_j \le \sum_{\ell=1}^s \frac{n_\ell - n_{\ell+1}}{n_\ell} \cdot \text{OPT}$$
$$\le \text{OPT} \sum_{\ell=1}^s \left( \frac{1}{n_\ell} + \frac{1}{n_\ell - 1} + \dots + \frac{1}{n_{\ell+1} + 1} \right)$$
$$= \text{OPT} \sum_{i=1}^k \frac{1}{i}$$
$$= H_n \cdot \text{OPT} \le \text{OPT}(\ln n + 1) \quad .$$



### **Technique 5: Randomized Rounding**

One round of randomized rounding:

Pick set  $S_j$  uniformly at random with probability  $1 - x_j$  (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.



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

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  $\ell$  rounds):

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



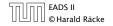
 $\Pr[\exists u \in U \text{ not covered after } \ell \text{ round}]$ 

$$= \Pr[u_1 \text{ not covered } \lor u_2 \text{ not covered } \lor \ldots \lor u_n \text{ not covered}]$$
  
$$\leq \sum_i \Pr[u_i \text{ not covered after } \ell \text{ rounds}] \leq ne^{-\ell} .$$

#### **Lemma 5** With high probability $O(\log n)$ rounds suffice.

#### With high probability:

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



Proof: We have

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



### **Expected Cost**

Version A.

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

$$E[\operatorname{cost}] \le (\alpha + 1) \ln n \cdot \operatorname{cost}(LP) + (\sum_{j} w_{j}) n^{-\alpha} = \mathcal{O}(\ln n) \cdot \operatorname{OPT}$$

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



### **Expected Cost**

Version B.

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

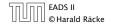
```
E[\text{cost}] = \Pr[\text{success}] \cdot E[\text{cost} | \text{success}]
+ \Pr[\text{no success}] \cdot E[\text{cost} | \text{no success}]
```

This means

E[cost | success]

$$= \frac{1}{\Pr[\mathsf{sucess}]} \left( E[\cos t] - \Pr[\mathsf{no success}] \cdot E[\cos t | \mathsf{no success}] \right)$$
  
$$\leq \frac{1}{\Pr[\mathsf{sucess}]} E[\cos t] \leq \frac{1}{1 - n^{-\alpha}} (\alpha + 1) \ln n \cdot \operatorname{cost}(LP)$$
  
$$\leq 2(\alpha + 1) \ln n \cdot \operatorname{OPT}$$
  
or  $n > 2$  and  $\alpha > 1$ 

for  $n \ge 2$  and  $\alpha \ge 1$ .



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  $2^{\operatorname{poly}(\log n)}$ ).



#### Techniques:

- Deterministic Rounding
- Rounding of the Dual
- Primal Dual
- Greedy
- Randomized Rounding
- Local Search
- Rounding the Data + Dynamic Programming

