Can we do better?

In the following we show how to obtain a solution where the number of bins is only

$$OPT(I) + O(\log^2(SIZE(I)))$$
.

Note that this is usually better than a guarantee of

 $(1+\epsilon)$ OPT(I) + 1 .

	16.4 Advanced Rounding for Bin Packing	
UUU © Harald Räcke	5 5	347

Configuration LP

A possible packing of a bin can be described by an *m*-tuple (t_1, \ldots, t_m) , where t_i describes the number of pieces of size s_i . Clearly,

$$\sum_i t_i \cdot s_i \le 1$$

We call a vector that fulfills the above constraint a configuration.

Configuration LP

Change of Notation:

- Group pieces of identical size.
- Let s₁ denote the largest size, and let b₁ denote the number of pieces of size s₁.
- s_2 is second largest size and b_2 number of pieces of size s_2 ;

348

- ▶ ...
- s_m smallest size and b_m number of pieces of size s_m .

EADS II © Harald Räcke	16.4 Advanced Rounding for Bin Packing
🛛 🕒 🛛 🖉 🛛 🖓 🛛 🖓 🖉 🖓 🖓	

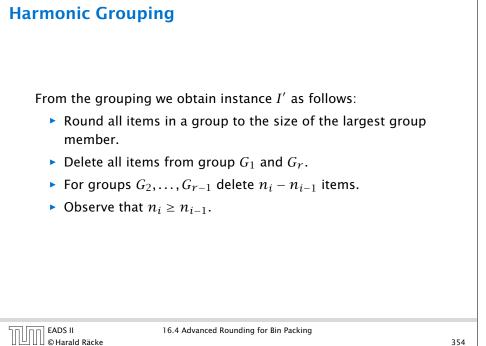
Configuration LPLet N be the number of configurations (exponential).Let T_1, \ldots, T_N be the sequence of all possible configurations (a configuration T_j has T_{ji} pieces of size s_i). $\min \qquad \sum_{j=1}^{N} x_j$
s.t. $\forall i \in \{1 \ldots m\} \qquad \sum_{j=1}^{N} T_{ji}x_j \geq b_i$
 $\forall j \in \{1, \ldots, N\} \qquad x_j \geq 0$
 $\forall j \in \{1, \ldots, N\} \qquad x_j$ integral $\forall I \in Advanced Rounding for Bin Packing$

16.4 Advanced Rounding for Bin Packing

How to solve this	LP?	
later		
EADS II © Harald Räcke	16.4 Advanced Rounding for Bin Packing	351
		221

Harmonic Grouping Harmonic Grouping Sort items according to size (monotonically decreasing). ▶ Process items in this order; close the current group if size of items in the group is at least 2 (or larger). Then open new member. • I.e., *G*¹ is the smallest cardinality set of largest items s.t. total size sums up to at least 2. Similarly, for G_2, \ldots, G_{r-1} . • Observe that $n_i \ge n_{i-1}$. • Only the size of items in the last group G_r may sum up to less than 2.

We can assume that each item has size at least 1/SIZE(I). EADS II © Harald Räcke 16.4 Advanced Rounding for Bin Packing 352



group.

16.4 Advanced Rounding for Bin Packing

Lemma 10

The number of different sizes in I' is at most SIZE(I)/2.

- Each group that survives (recall that G₁ and G_r are deleted) has total size at least 2.
- Hence, the number of surviving groups is at most SIZE(I)/2.
- All items in a group have the same size in I'.

	16.4 Advanced Rounding for Bin Packing	
🛛 💾 🛛 🖉 © Harald Räcke		355

Algorithm 1 BinPack

1: **if** SIZE(I) < 10 **then**

- 2: pack remaining items greedily
- 3: Apply harmonic grouping to create instance I'; pack discarded items in at most $O(\log(SIZE(I)))$ bins.
- 4: Let x be optimal solution to configuration LP
- 5: Pack $\lfloor x_j \rfloor$ bins in configuration T_j for all j; call the packed instance I_1 .
- 6: Let I_2 be remaining pieces from I'
- 7: Pack I_2 via BinPack (I_2)

Lemma 11

The total size of deleted items is at most $O(\log(SIZE(I)))$.

- ► The total size of items in G₁ and G_r is at most 6 as a group has total size at most 3.
- Consider a group G_i that has strictly more items than G_{i-1} .
- It discards $n_i n_{i-1}$ pieces of total size at most

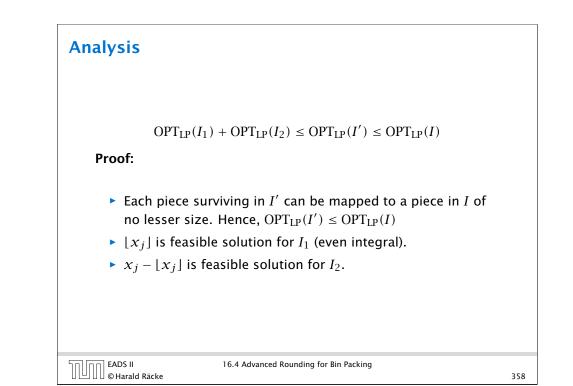
$$3\frac{n_i - n_{i-1}}{n_i} \le \sum_{j=n_{i-1}+1}^{n_i} \frac{3}{j}$$

since the smallest piece has size at most $3/n_i$.

Summing over all *i* that have n_i > n_{i-1} gives a bound of at most

$$\sum_{j=1}^{n_{r-1}} \frac{3}{j} \le \mathcal{O}(\log(\text{SIZE}(I))) \ .$$

(note that $n_r \leq \text{SIZE}(I)$ since we assume that the size of each item is at least 1/SIZE(I)).



Analysis

Each level of the recursion partitions pieces into three types

- 1. Pieces discarded at this level.
- **2.** Pieces scheduled because they are in I_1 .
- **3.** Pieces in I_2 are handed down to the next level.

Pieces of type 2 summed over all recursion levels are packed into at most $\mbox{OPT}_{\mbox{LP}}$ many bins.

Pieces of type 1 are packed into at most

 $\mathcal{O}(\log(\text{SIZE}(I))) \cdot L$

many bins where L is the number of recursion levels.

	16.4 Advanced Rounding for Bin Packing	
🛛 🕒 🛛 🖉 © Harald Räcke		359

How to solve the LP?

Let T_1, \ldots, T_N be the sequence of all possible configurations (a configuration T_j has T_{ji} pieces of size s_i). In total we have b_i pieces of size s_i .

Primal

min		$\sum_{j=1}^{N} x_j$		
s.t.	$\forall i \in \{1 \dots m\}$	$\sum_{j=1}^{N} T_{ji} x_j$	\geq	b_i
	$\forall j \in \{1, \dots, N\}$	x_j		~

Dual

s.t. $\forall j \in \{1, \dots, N\}$ $\sum_{i=1}^{m} T_{ji} \mathcal{Y}_i \leq$			$\sum_{i=1}^{m} \gamma_i b_i$	ax
	1	\leq	$\sum_{i=1}^{m} T_{ji} \gamma_i$	$ \text{ s.t. } \forall j \in \{1, \dots, N\} $
$\forall i \in \{1, \dots, m\} \qquad \qquad \mathcal{Y}_i \geq \mathcal{Y}_i$	0	\geq	${\mathcal Y}_i$	$\forall i \in \{1, \ldots, m\}$

ղվող	EADS II	
	© Harald	Räcke

Analysis

We can show that $SIZE(I_2) \le SIZE(I)/2$. Hence, the number of recursion levels is only $O(\log(SIZE(I_{original})))$ in total.

- ► The number of non-zero entries in the solution to the configuration LP for I' is at most the number of constraints, which is the number of different sizes (\leq SIZE(I)/2).
- ► The total size of items in I_2 can be at most $\sum_{j=1}^{N} x_j \lfloor x_j \rfloor$ which is at most the number of non-zero entries in the solution to the configuration LP.

EADS II © Harald Räcke 16.4 Advanced Rounding for Bin Packing

Separation Oracle Suppose that I am given variable assignment γ for the dual. How do I find a violated constraint? I have to find a configuration $T_j = (T_{j1}, ..., T_{jm})$ that • is feasible, i.e., $\sum_{i=1}^{m} T_{ji} \cdot s_i \le 1$, • and has a large profit $\sum_{i=1}^{m} T_{ji} \gamma_i > 1$

But this is the Knapsack problem.

361

16.4 Advanced Rounding for Bin Packing

360

Separation Oracle

We have FPTAS for Knapsack. This means if a constraint is violated with $1 + \epsilon' = 1 + \frac{\epsilon}{1-\epsilon}$ we find it, since we can obtain at least $(1 - \epsilon)$ of the optimal profit.

The solution we get is feasible for:

Dual'

max		$\sum_{i=1}^{m} y_i b_i$		
s.t.	$\forall j \in \{1, \dots, N\}$	$\sum_{i=1}^{m} T_{ji} \mathcal{Y}_i$	\leq	$1 + \epsilon'$
	$\forall i \in \{1, \ldots, m\}$	${\mathcal Y}_i$	\geq	0

Primal'

s.t. $\forall i \in \{1,m\}$ $\sum_{j=1}^{N} T_{ji} x_j \geq b_i$ $\forall j \in \{1,,N\}$ $x_j \geq 0$	min		$(1+\epsilon')\sum_{j=1}^N x_j$		
$\forall j \in \{1, \dots, N\} \qquad \qquad x_j \ge 0$	s.t.	$\forall i \in \{1 \dots m\}$	$\sum_{j=1}^{N} T_{ji} x_j$	\geq	b_i
		$\forall j \in \{1, \dots, N\}$	x_j	\geq	0

Separation Oracle

If the value of the computed dual solution (which may be infeasible) is z then

 $OPT \le z \le (1 + \epsilon')OPT$

How do we get good primal solution (not just the value)?

- The constraints used when computing z certify that the solution is feasible for DUAL'.
- Suppose that we drop all unused constraints in DUAL. We will compute the same solution feasible for DUAL'.
- \blacktriangleright Let $\mathrm{DUAL}^{\prime\prime}$ be DUAL without unused constraints.
- The dual to DUAL" is PRIMAL where we ignore variables for which the corresponding dual constraint has not been used.
- The optimum value for PRIMAL'' is at most $(1 + \epsilon')$ OPT.
- We can compute the corresponding solution in polytime.

This gives that overall we need at most

$$(1 + \epsilon')$$
OPT_{LP} $(I) + O(\log^2(SIZE(I)))$

bins.

We can choose $\epsilon' = \frac{1}{OPT}$ as $OPT \le \#$ items and since we have a fully polynomial time approximation scheme (FPTAS) for knapsack.

365