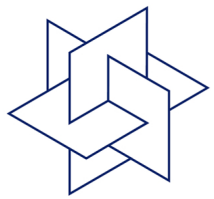
Binpacking

CO@Work Berlin

Stefan Heinz

Zuse Institute Berlin

09/30/2009

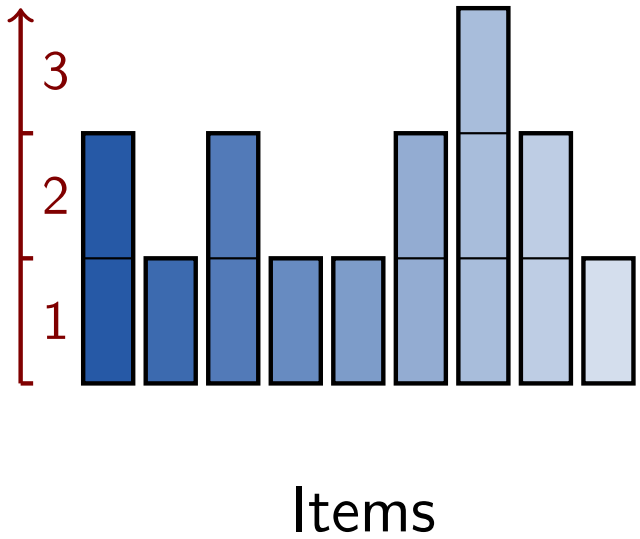


DFG Research Center MATHEON
Mathematics for key technologies

Problem Description

Definition

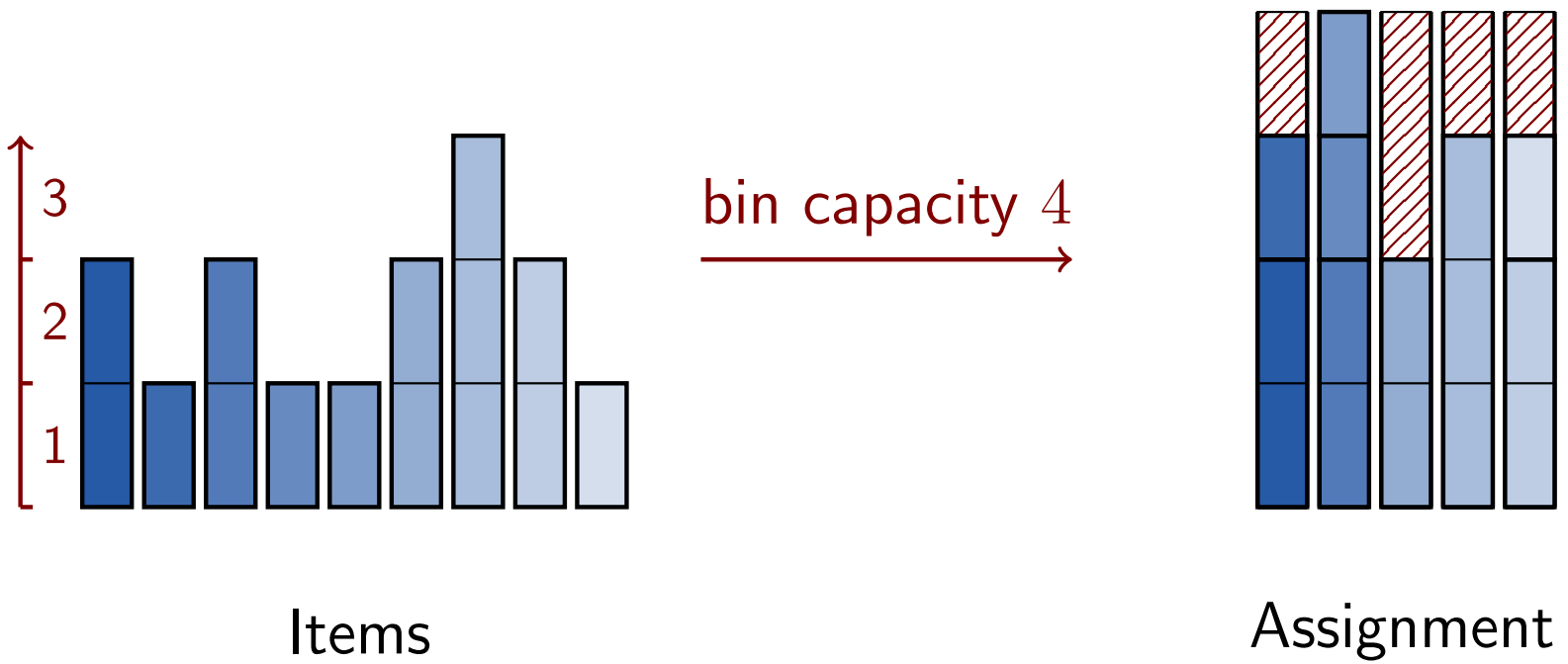
The binpacking problem consists of assigning sized items to bins of given capacities such that the total number of used bins is minimized.



Problem Description

Definition

The binpacking problem consists of assigning sized items to bins of given capacities such that the total number of used bins is minimized.



Assignment Formulation

Given

- bin capacity κ
- $\mathcal{I}$ set of items, $|\mathcal{I}| = n$
  - s_i size of order i

Binary variables

- $x_j = 1$ if bin j is used
- $y_{ij} = 1$ if order i is assigned to bin j

Set Covering Formulation

Packing

- A packing s is an assignment vector $\lambda_s \in \{0,1\}^n$.
- Defining which items belong to packing s.
- Item $i \in \{1,\ldots,n\}$ belongs to packing s if $(\lambda_s)_i$ is one.

Set Covering Formulation

Packing

- A packing s is an assignment vector $\lambda_s \in \{0,1\}^n$.
- Defining which items belong to packing s.
- Item $i \in \{1,\ldots,n\}$ belongs to packing s if $(\lambda_s)_i$ is one.

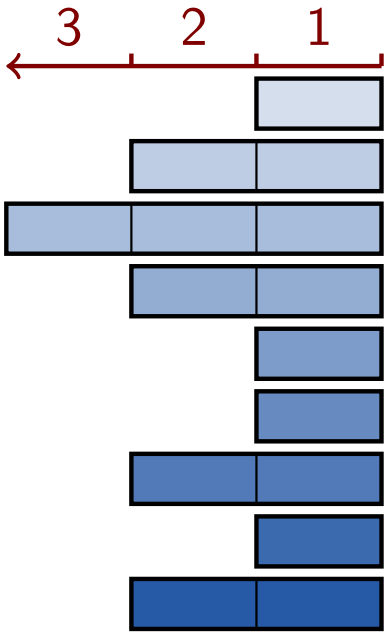
Feasible Packing

- Capacity restriction

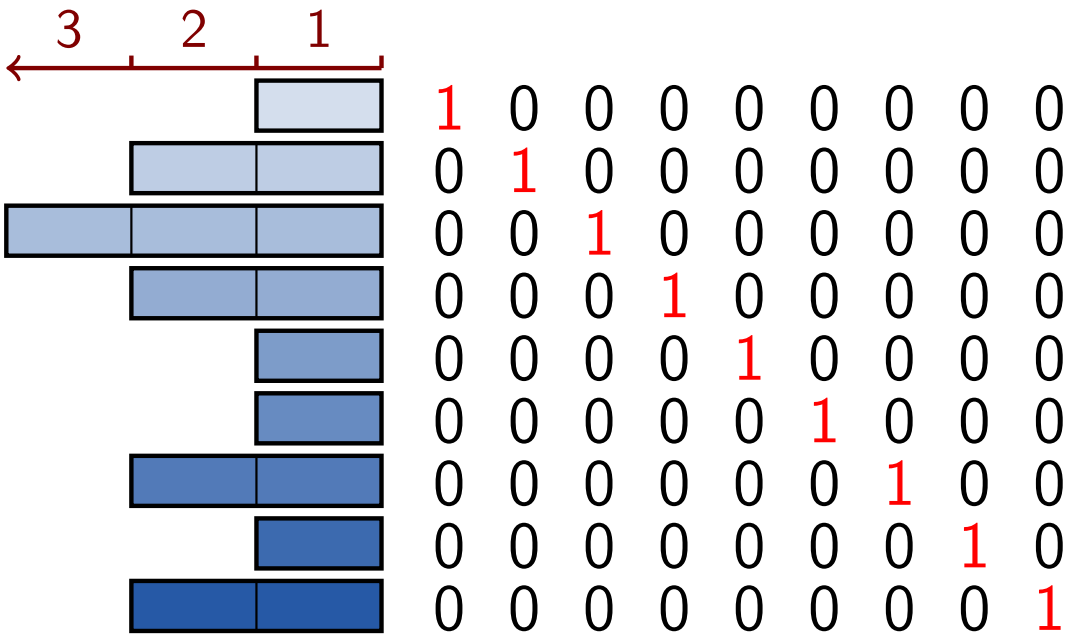
Binary variables

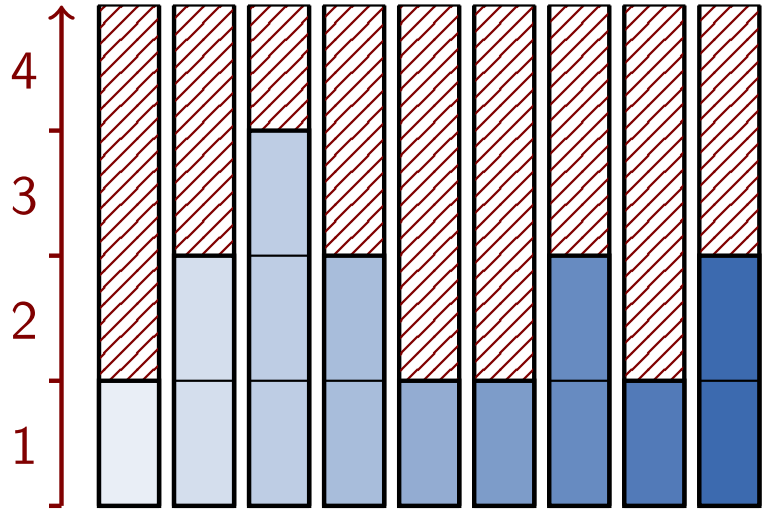
- $x_s = 1$ if packing s is used

Example

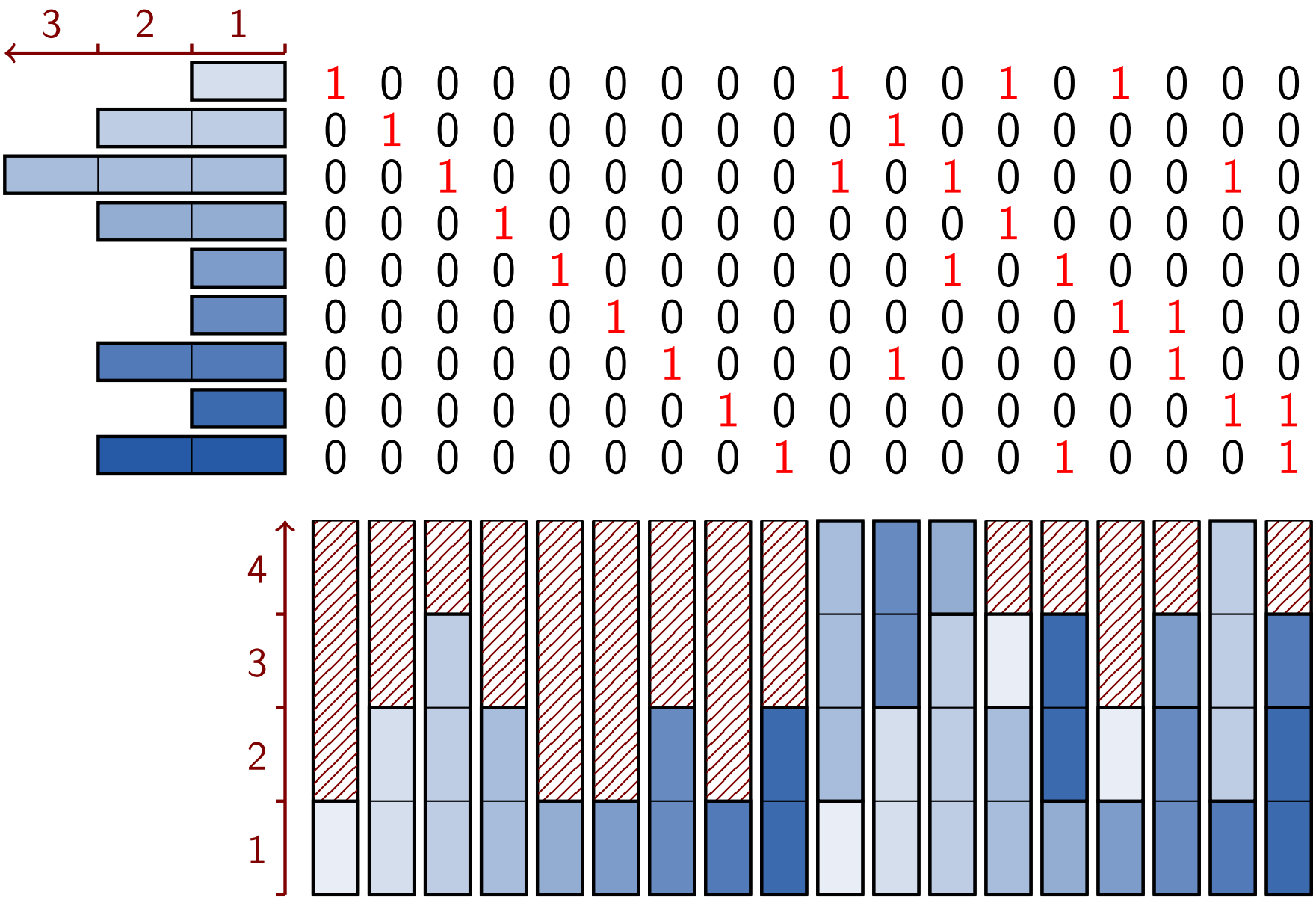


Example


3
2
1
1 0 0 0 0 0 0 0 0
0 1 0 0 0 0 0 0 0
0 0 1 0 0 0 0 0 0
0 0 0 1 0 0 0 0 0
0 0 0 0 1 0 0 0 0
0 0 0 0 0 1 0 0 0
0 0 0 0 0 0 1 0 0
0 0 0 0 0 0 0 1 0
0 0 0 0 0 0 0 0 1


4
3
2
1

Example


3
2
1
1 0 0 0 0 0 0 0 0 1 0 0 1 0 1 0 0 0
0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0
0 0 1 0 0 0 0 0 0 1 0 1 0 0 0 0 1 0
0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0
0 0 0 0 1 0 0 0 0 0 0 1 0 1 0 0 0 0
0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 1 0 0
0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 1 0 0
0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 1
0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 1
4
3
2
1

Exercise 1

1. Create an Assignment and a Set Covering Formulation for the binpacking problem
2. Model these with ZIMPL
3. Run automated tests for both model
4. Compare results

ZIMPL Model

```
######################
#
# data parsing
#
######################

# problem name
param name := read DATAFILE as "1s" use 1;
do print "name = ", name;

# capacity
param capacity := read DATAFILE as "1n" use 1 skip 1;
do print "capacity = ", capacity;

# number of items
param nitems := read DATAFILE as "2n" use 1 skip 1;
do print "number of items = ", nitems;

# index set for the items
set I := {1..nitems};

# get item sizes
set sizes[<i> in I] := {read DATAFILE as "<1n>" skip 1+i use 1};
do forall <i> in I do check card(sizes[i]) == 1;
```

IP Formulations

Assignment

$$\begin{aligned} \min \quad & \sum_{j \in \mathcal{B}} x_j & & \\ \text{subject to} \quad & \sum_{j \in \mathcal{B}} y_{ij} = 1 & & \forall i \in \mathcal{I} \\ & \sum_{i \in \mathcal{I}} s_i \, y_{ij} \leq c & & \forall j \in \mathcal{B} \\ & y_{ij} \leq x_j & & \forall i \in \mathcal{I} \; \forall j \in \mathcal{B} \\ & x_j, y_{ij} \in \{0,1\} & & \forall i \in \mathcal{I} \; \forall j \in \mathcal{B} \end{aligned}$$

Set Covering

$$\begin{aligned} \min \quad & \sum_{s \in \mathcal{S}} x_s & & \\ \text{subject to} \quad & \sum_{s \in \mathcal{S}} (\lambda_s)_i \, x_s \geq 1 & & \forall i \in \mathcal{I} \\ & x_s \in \{0,1\} & & \forall s \in \mathcal{S} \end{aligned}$$