Cardinality Reasoning for bin-packing constraint.

Application to a tank allocation problem

Pierre Schaus, Jean-Charles Régin,
Rowan Van Schaeren, Wout Dullaert, and Birger Raa

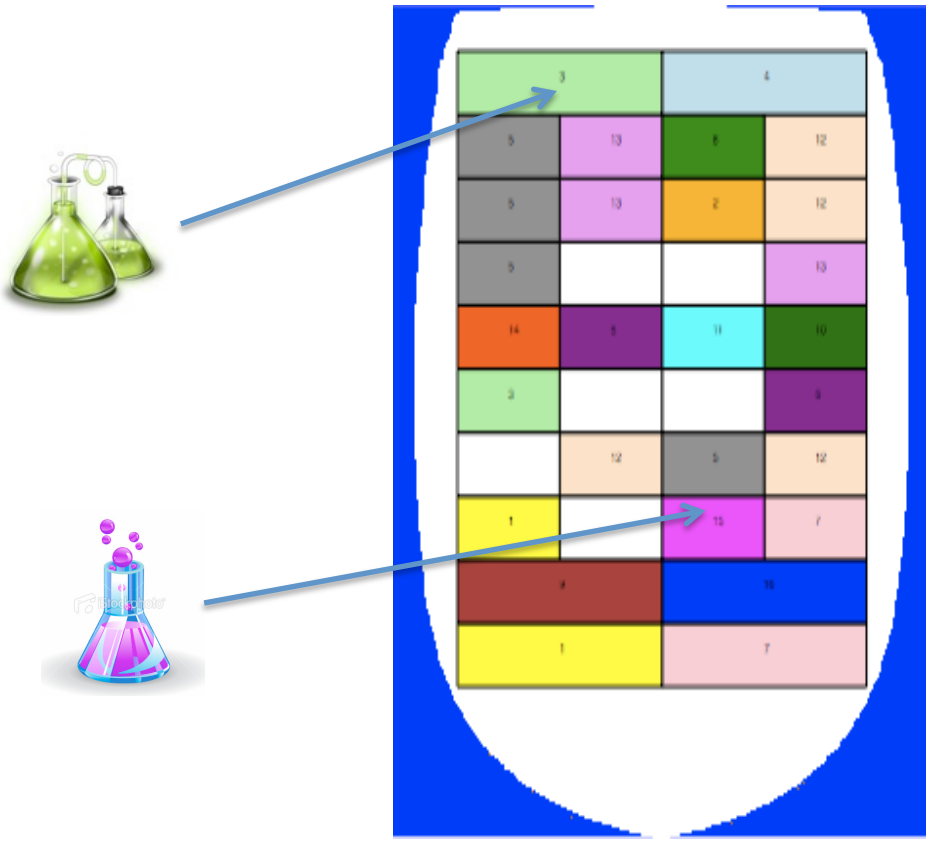
Chemical Tanker

Problem Description

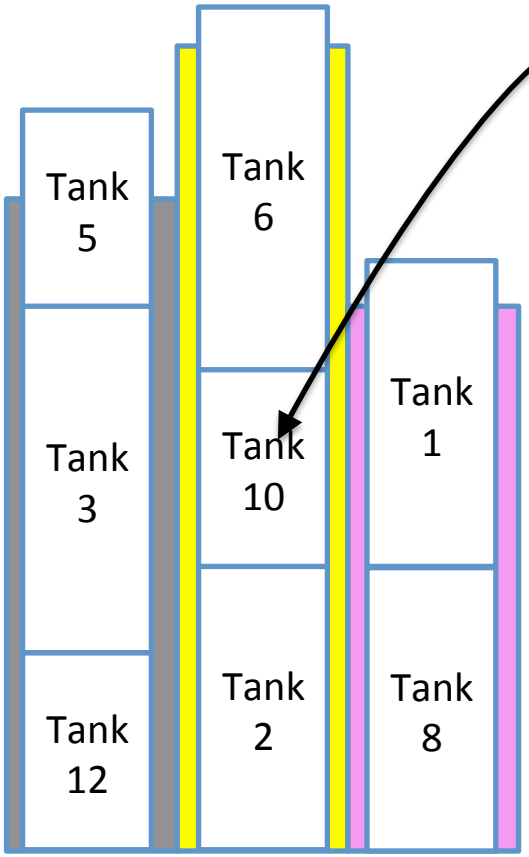
- A set of chemical products must be placed into tanks on a tanker
- For each product a volume (liters) to place is given
- Each tank has a maximum capacity

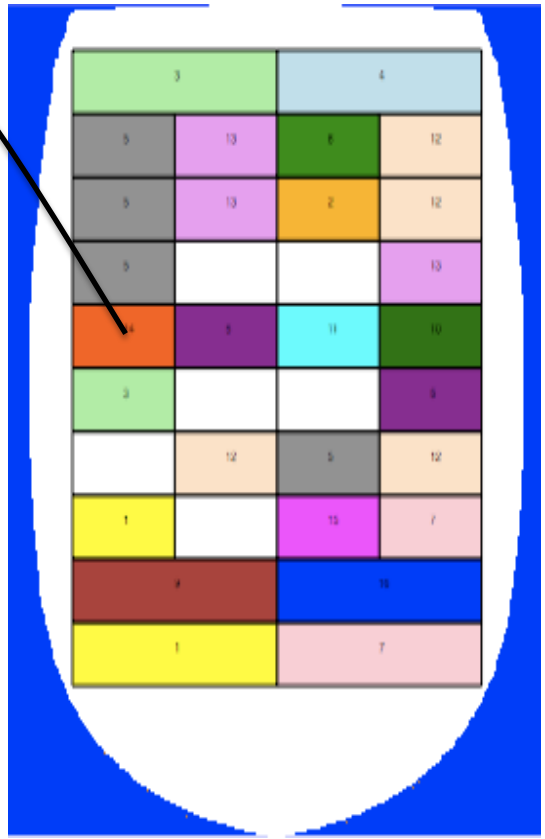
Objective:

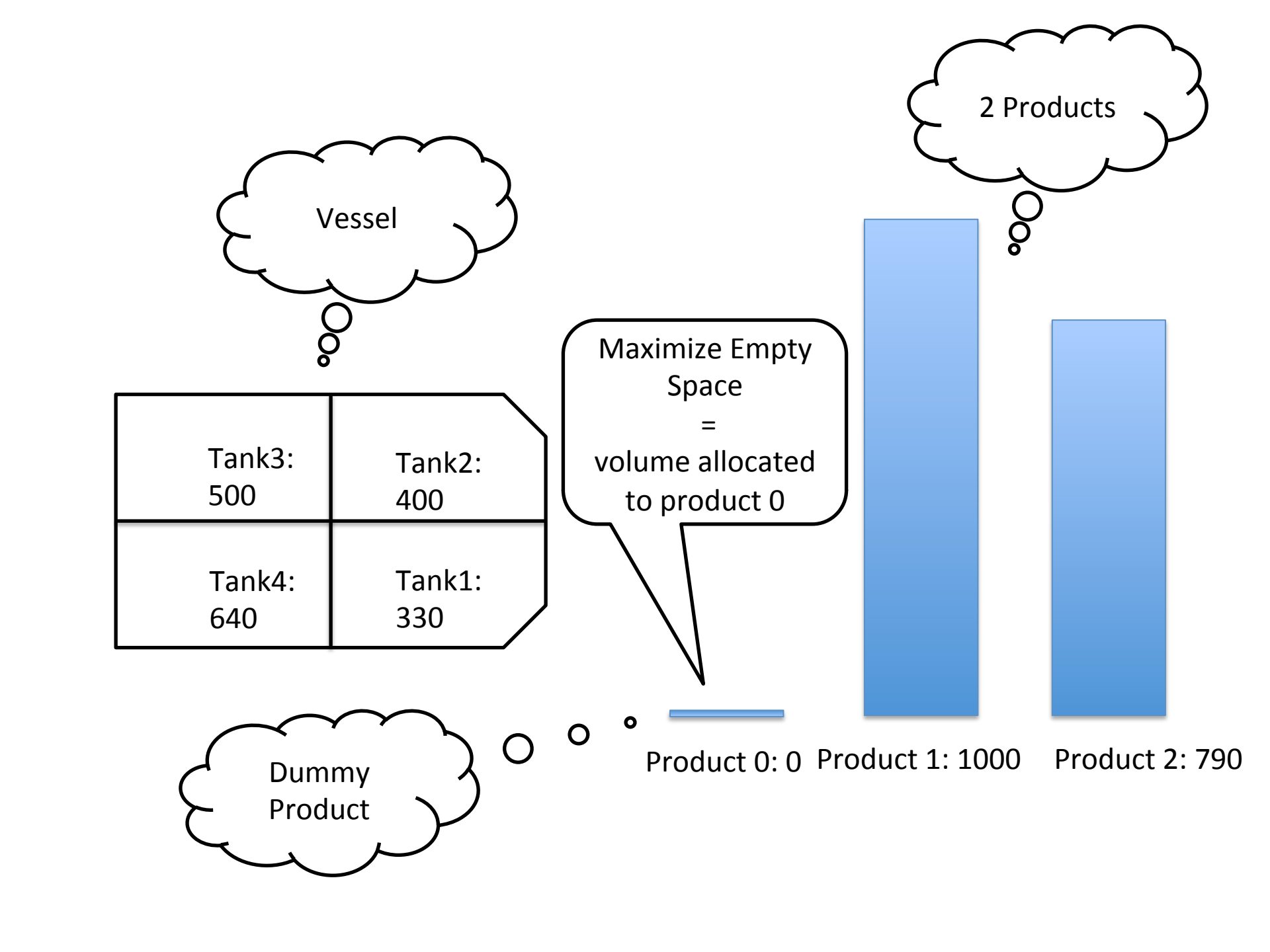
- Maximize the empty/ unused space while placing all the products



Intuition of the model: Tanks added to the products






2 Products
Vessel
Maximize Empty Space = volume allocated to product 0
Tank3: 500
Tank2: 400
Tank4: 640
Tank1: 330
Dummy Product
Product 0: 0
Product 1: 1000
Product 2: 790

Bin-Packing Model

Variables:

Decision Variables

- x1,x2,x3,x4 = is the product placed into each tank.
- load0,load1,load2 = the total tank volume allocated to each product.

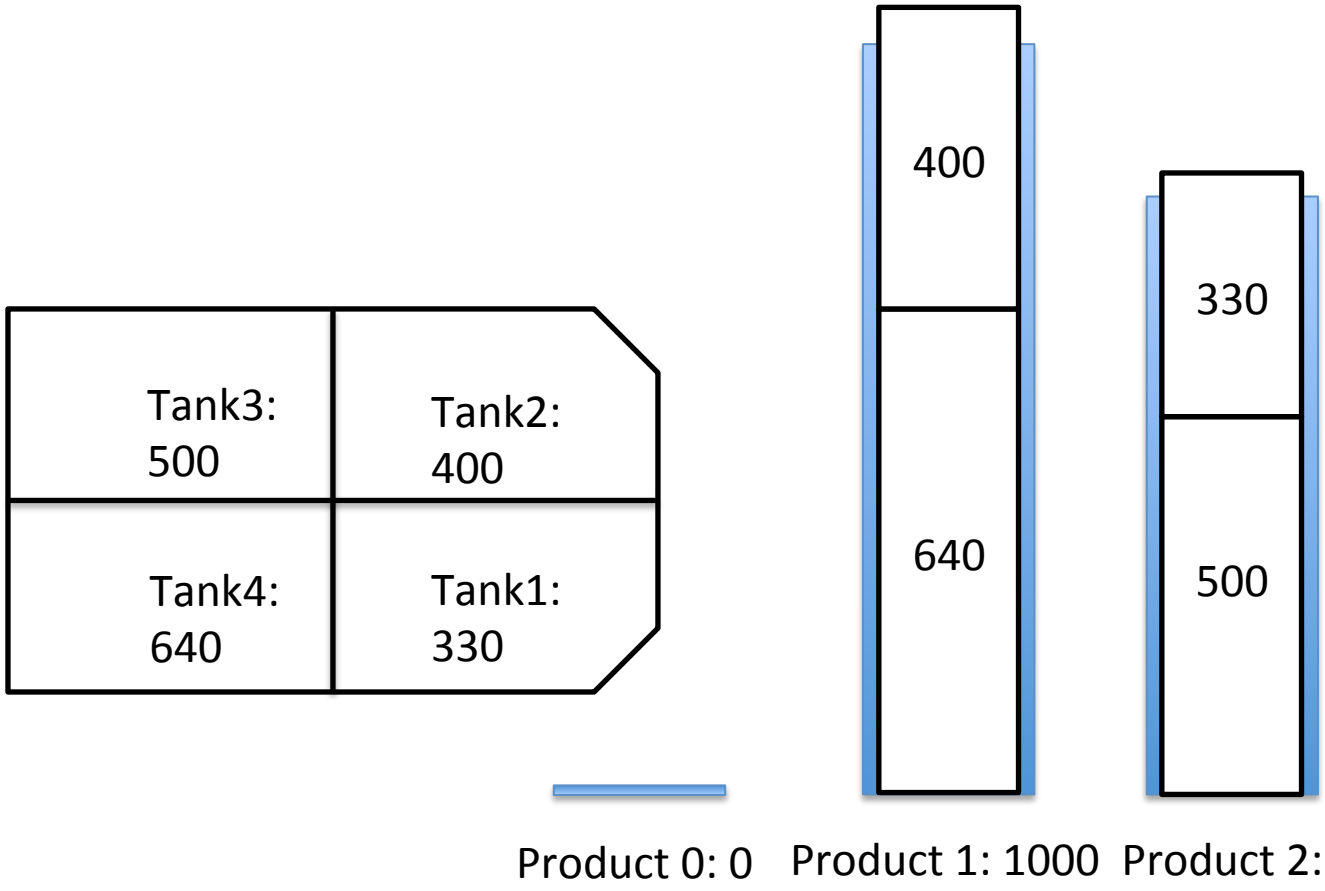
Constraint:

BinPacking([x1,x2,x3,x4],[330,400,500,640],[load0,load1,load2])

Dom(x1) = {0,1,2}

Dom(load1) = [1000, +inf]

A constraint for bin packing, P Shaw, CP 2004



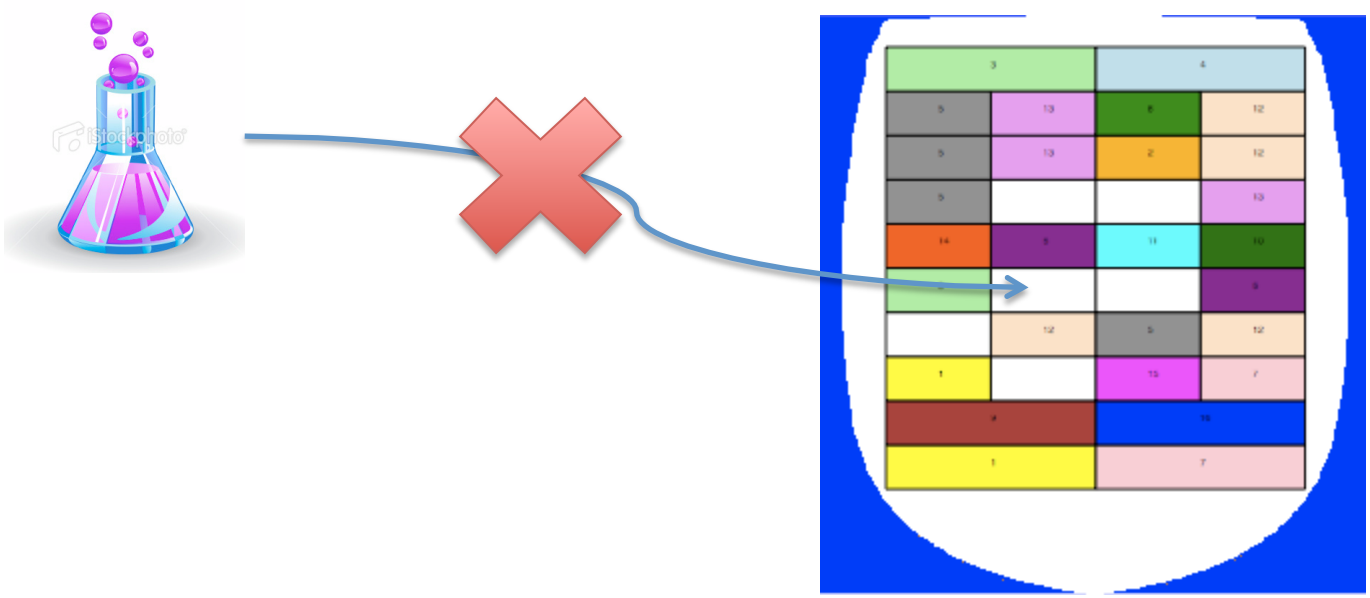
BinPacking([x1=2,x2=1,x3=2,x4=1],
[330, 400, 500, 640],
[load0=0,load1=1040,load2=830])

Remark

- We **care** there is enough volume to accommodate each product
- We **don't care** about the exact volume placed in each tank (this should take equilibrium constraints into considerations).

More Constraints

- Some products cannot be placed in some tanks

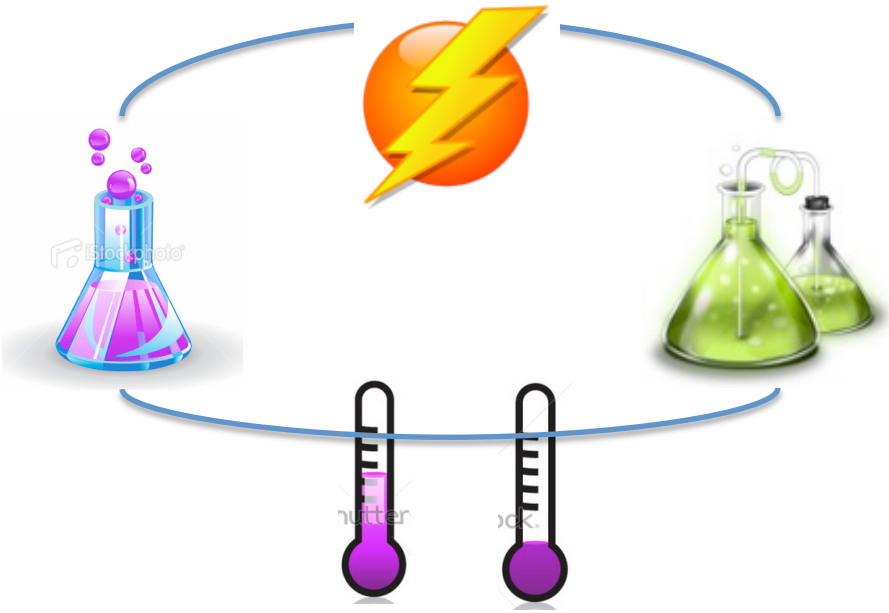


→ Removal of values from domain of x’s

More Constraints

- Some products cannot be placed in neighboring tanks because of

Dangerous reaction



Heating requirements too different

Why … ?

Some products cannot be placed in neighboring tanks

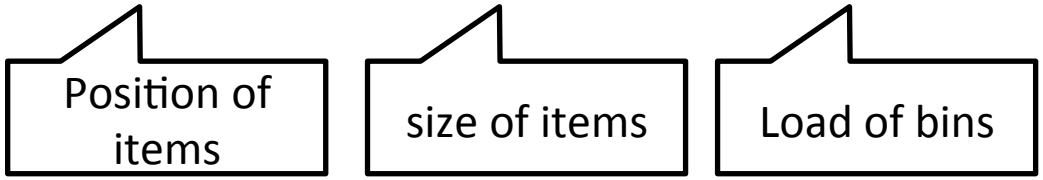
- Table of possible adjacency combinations

Product A	Product B
1	4
1	6
2	3
3	5
4	5
4	6

➔ Table constraint

Improved Filtering for Bin-Packing

BinPacking([x1,…,xn],[s1,…,sn],[l1,…,lm])

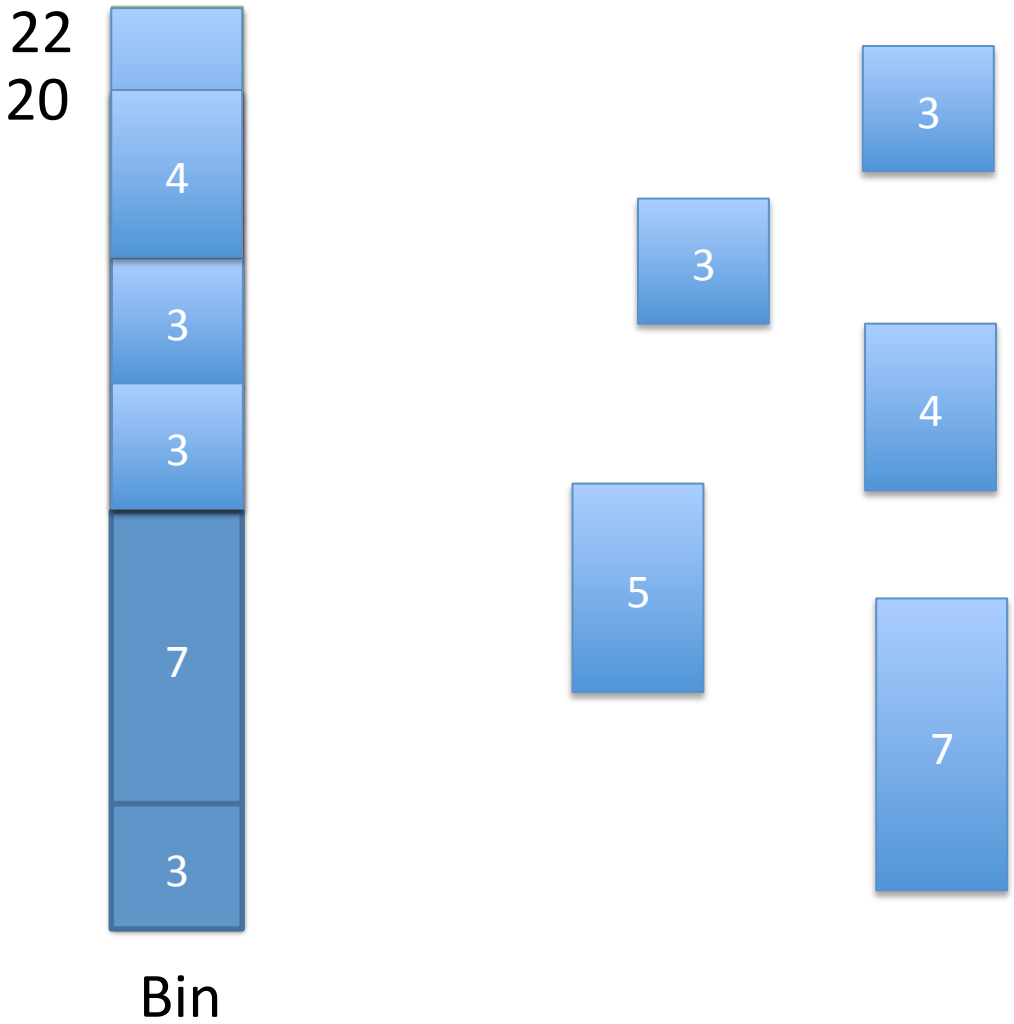


Idea: add gcc([x1,…,xn],[c1,…,cn])

with cj = number of items in bin j

Question: How to filter cj's?

Improved Filtering for Bin-Packing



Minimum cardinality = 2+2

Maximum cardinality = 2+3

Implem

BinPacking([x1,…,xn],[s1,…,sn],[l1,…,lm])

+

gcc([x1,…,xn],[c1,…,cn])

+

Forall j: BinPackingCardinality([x1,…,xn],[s1,…,sn],lj,cj)

Propagator in charge of updating cardinalities
O(n) to update the cardinality (items sorted by size)

Experiment

Data set publicly available here:

http://becool.info.ucl.ac.be/resources/tank-allocation-problem

- Basic model : NO Solution
- Basic model + redundant gcc: empty space = 1811 (5 minutes and 1,594,159 fails)
- Basic model + redundant gcc + LNS: empty space = 2296 within 3 seconds and after a dozen of restarts.
- Cplex: 3 seconds and can prove optimality

→ CP should improve to reach cplex on this problem (bin-packing?)

Cardinality Reasoning for bin-packing constraint.

Application to a tank allocation problem

Pierre Schaus, Jean-Charles Régin,
Rowan Van Schaeren, Wout Dullaert, and Birger Raa