FACULDADE DE CIÊNCIAS
UNIVERSIDADE DO PORTO

Arc-flow Formulation for Vector Packing

Filipe Brandão and João Pedro Pedroso

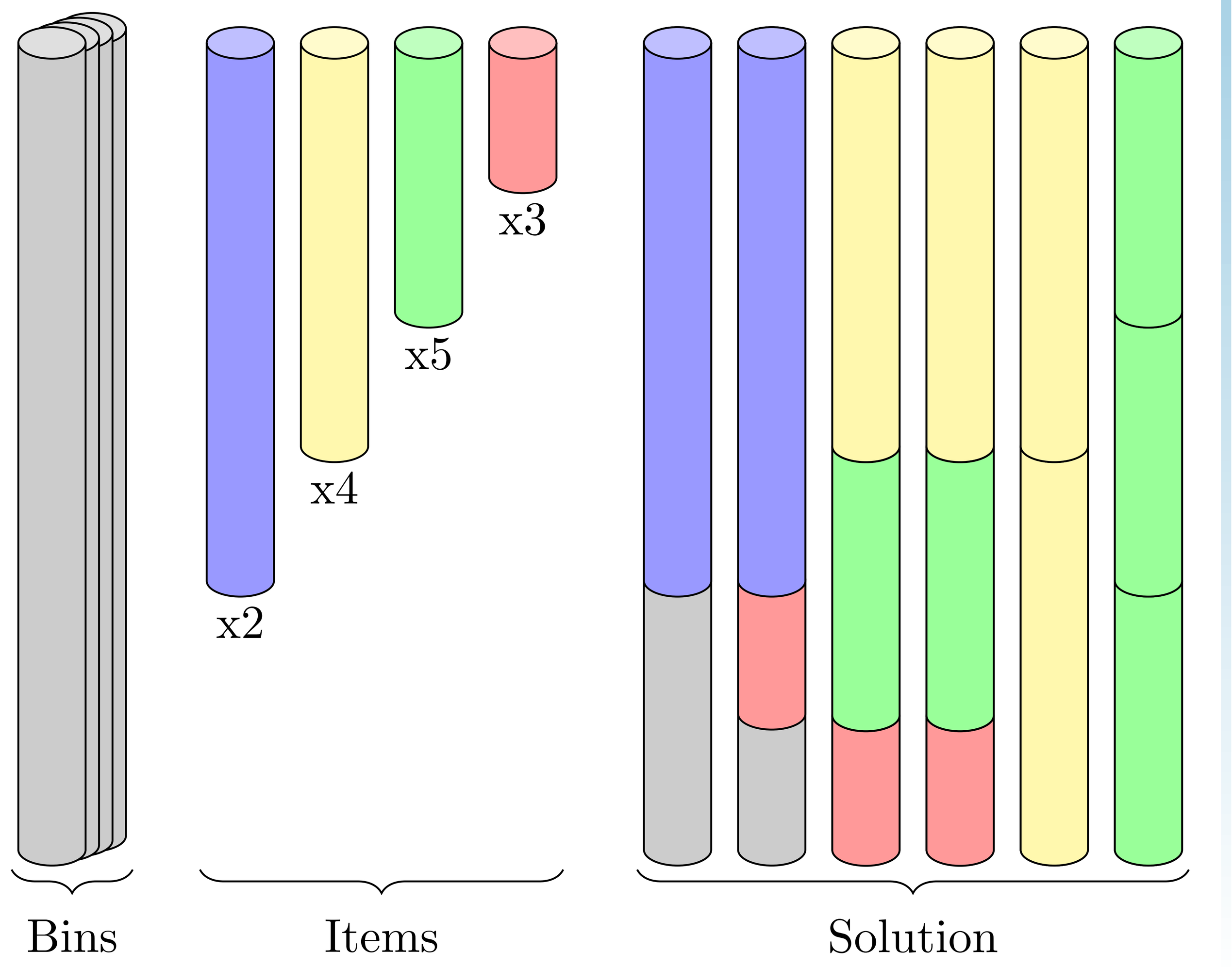
U.PORTO

I. Contribution

Exact method, based on an arc-flow formulation, for solving bin packing and cutting stock problems including multi-constraint variants.

II. Bin Packing/Cutting Stock

Objective: Pack a set of items into as few bins as possible



III. p-dimensional Vector Packing

- **Bin packing with multiple constraints**
- Pack n items of m different types, represented by p-dimensional vectors, into as few bins as possible.

IV. Assignment-based model

$$\min \sum_{j=1}^{n} y_j$$

$$\text{s.t.} \quad \sum_{j=1}^{n} x_{ij} \geq b_i, \qquad i = 1..m,$$

$$\sum_{i=1}^{m} w_i^k x_{ij} \leq y_j W^k, \qquad j = 1..n,\ k = 1..p,$$

$$y_j \in \{0,1\}, \qquad j = 1..n,$$

$$x_{ij} \geq 0, \text{ integer}, \qquad i = 1..m,\ j = 1..n,$$

where w_i and b_i are the weight vector and demand of items of type i, and W is the capacity vector. The variables are y_j, which is 1 if bin j is used and 0 otherwise, and x_{ij}, the number of times item i is assigned to bin j.

- Highly symmetric
- Very weak linear relaxation

V. Gilmore-Gomory's model

Let column vectors $a^j = (a_1^j, \ldots, a_m^j)^\top$ represent all possible cutting patterns j. The element a_i^j represents the number of items of type i in pattern j. Let x_j be a decision variable for the number of times pattern j is used.

$$\min \sum_{j \in J} x_j$$

$$\text{s.t.} \quad \sum_{j \in J} a_i^j x_j \geq b_i, \qquad i = 1..m,$$

$$x_j \geq 0, \text{ integer}, \qquad \forall j \in J,$$

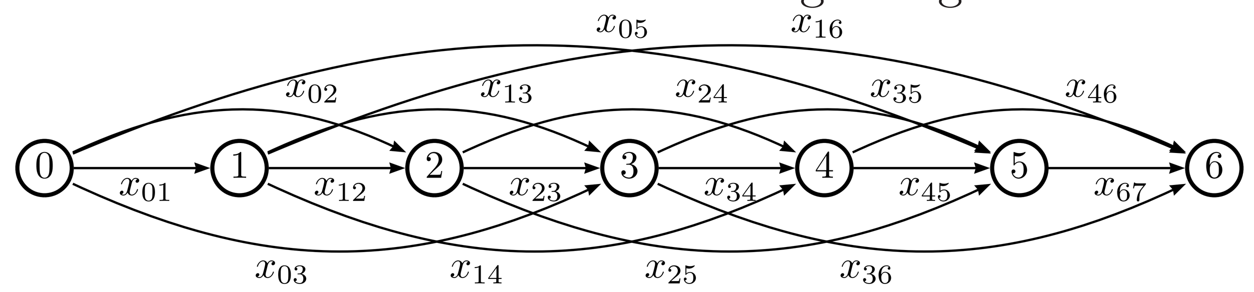
where J is the set of valid cutting patterns that satisfy:

$$\sum_{i=1}^{m} a_i^j w_i^k \leq W^k,\ k = 1..p,\ a_i^j \in \mathbb{N}_0.$$

- Very flexible
- Strong linear relaxation
- Exponential number of variables

VI. Valério de Carvalho's model

Consider decision variables x_{ij} corresponding to the number of items of size $j - i$ placed in any bin at a distance of i units from the beginning of the bin.

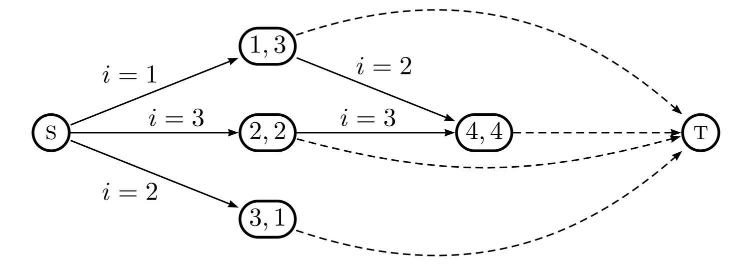


One-dimensional packing problems can be solved as a minimum flow between vertex 0 and vertex W with demand constraints.

- Strong linear relaxation
- Only models one-dimensional problems
- Large number of variables and constraints

VII. Vector Packing Graph

- For modeling p-dimensional problems, we use graphs with p-dimensional node labels.
- Every valid packing pattern is represented as a path from the source S to the target T.
- We only need to consider paths that respect a fixed order (permutations of items are redundant).



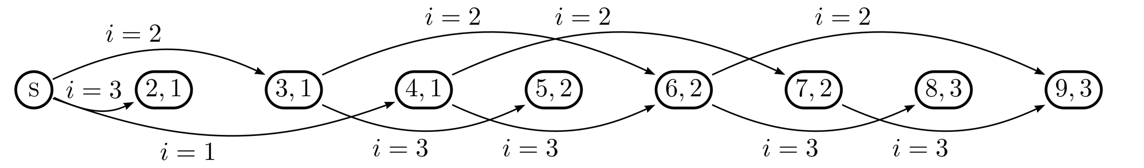
i	w_i^1	w_i^2	b_i
1	1	3	1
2	3	1	1
3	2	2	2
$W^1 = 7$		$W^2 = 3$	

The dashed arcs are loss arcs that represent unoccupied portions of the patterns.

VIII. Graph compression

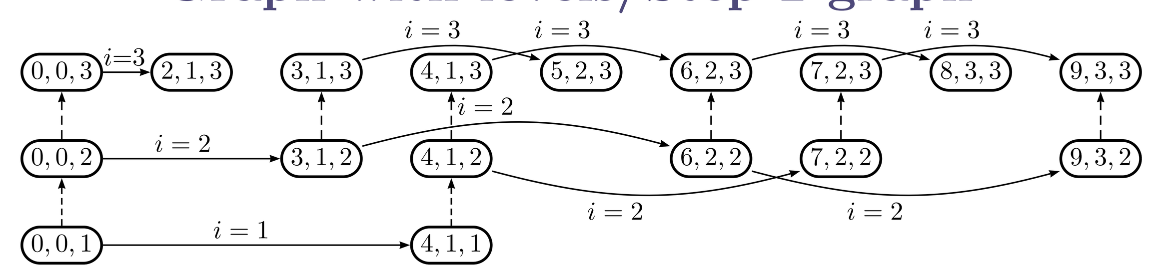
Consider an instance with bins of capacity $W = (9,3)$ and items of sizes (4,1), (3,1), (2,1) with demands 1, 3, 1, respectively.

Initial graph/Step-1 graph*



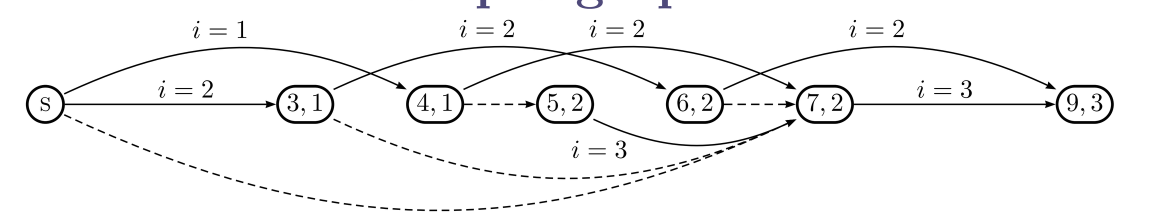
1) Break symmetry: we divide the graph into levels, one level for each different item.

Graph with levels/Step-2 graph*



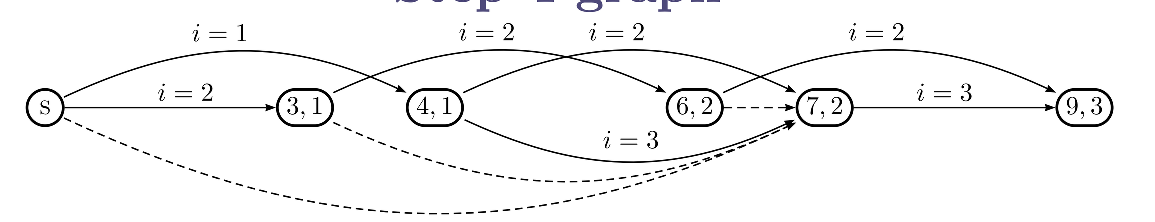
2) Main compression phase: we use the longest paths to the target in each dimension to relabel the nodes.

Step-3 graph*



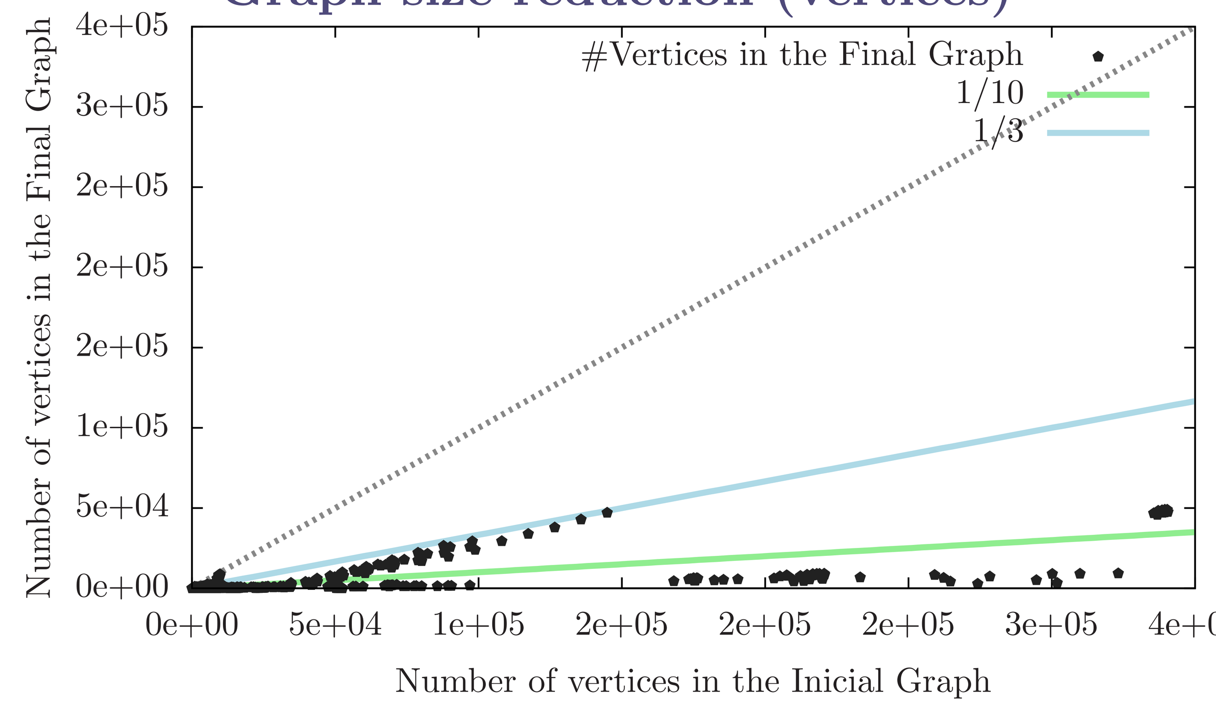
3) Final compression phase: we use the longest paths from the source in each dimension to relabel the nodes.

Step-4 graph*

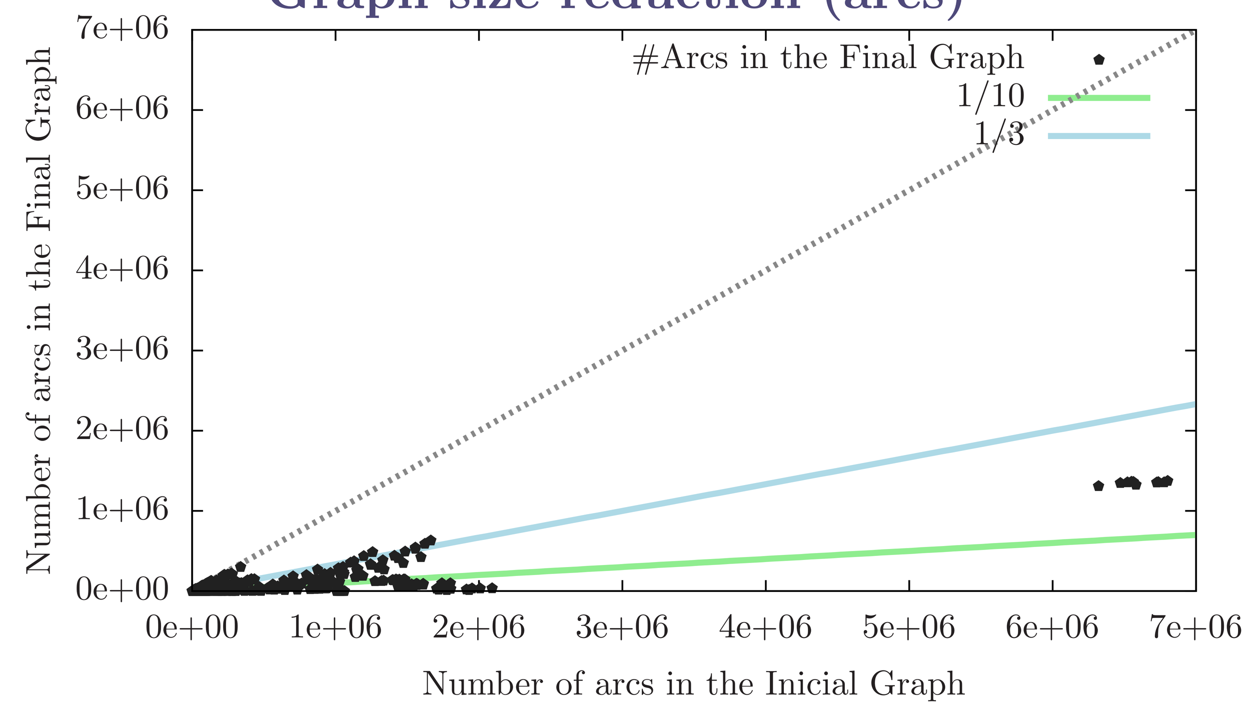


$*$ - the target T and the loss arcs connecting every internal node to it were omitted for simplicity.

Graph size reduction (vertices)



Graph size reduction (arcs)



IX. General Arc-flow model

Our arc-flow model only requires a directed acyclic multigraph $G = (V, A)$ containing every valid packing pattern represented as a path from the source to the target in order to solve the corresponding cutting/packing problem.

$$\min z$$

$$\text{s.t.} \quad \sum_{(u,v,i) \in A: v=k} f_{uvi} - \sum_{(v,r,i) \in A: v=k} f_{vri} = \begin{cases} -z & \text{if } k = \text{S}, \\ z & \text{if } k = \text{T}, \\ 0 & \text{for } k \in V \setminus \{\text{S}, \text{T}\}, \end{cases}$$

$$\sum_{(u,v,i) \in A: i=j} f_{uvi} \geq b_j, \qquad j = 1..m,$$

$$f_{uvi} \geq 0, \text{ integer}, \qquad \forall (u,v,i) \in A,$$

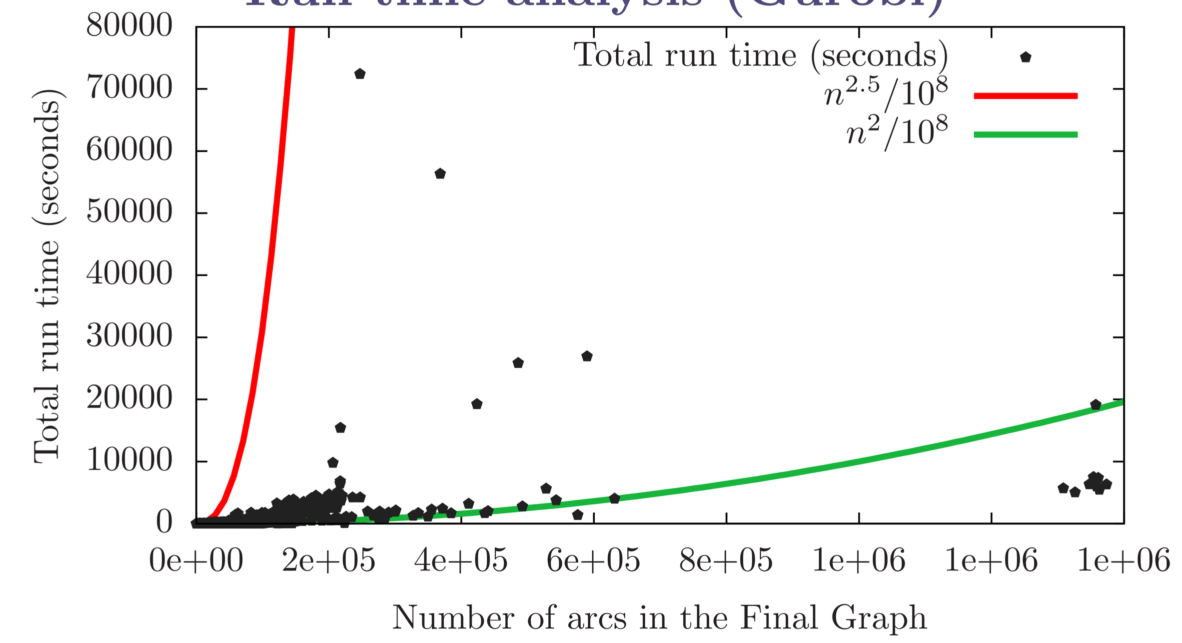
where (u,v,i) denotes an arc between u and v associated with items of type i, arcs $(u,v,i=0)$ are loss arcs, and f_{uvi} is the amount of flow along the arc (u,v,i).

- Very flexible
- Strong linear relaxation
- Reasonably small models (graph compression!)

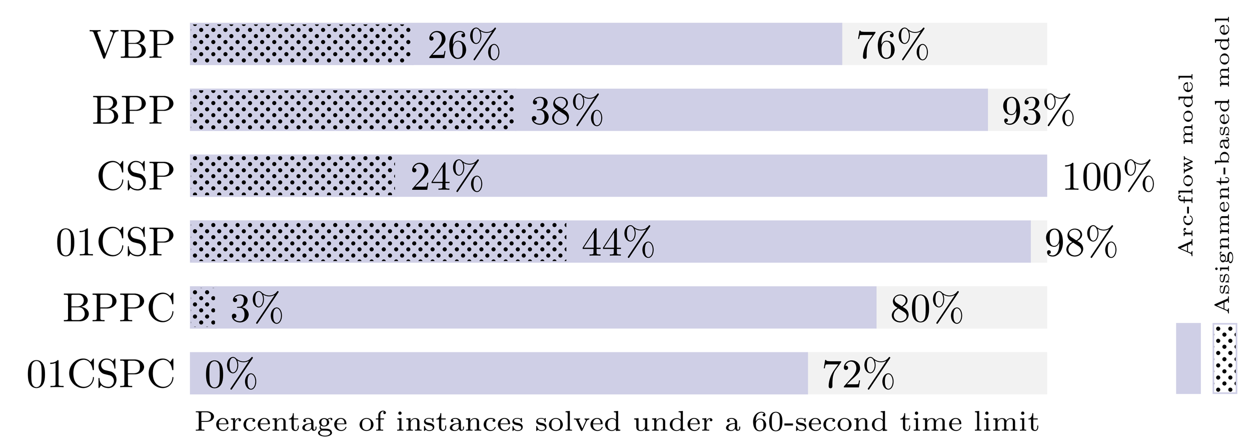
X. Results

Using the proposed method, we solved 23,153 benchmark instances on a desktop computer, spending 33 seconds per instance, on average.

Run time analysis (Gurobi)



These benchmark instances belong to several strongly NP-hard problems such as vector packing (VBP), bin packing (BPP), cutting stock (CSP), CSP with binary patterns (01CSP), BPP with conflicts (BPPC), and 01CSP with forbidden pairs (01CSPC).



- We solved benchmark instances with up to millions of items of 1,000 different types and 1,000 dimensions.
- **Despite its simplicity and generality, the proposed method outperforms complex problem-specific approaches such as branch-and-price algorithms.**

References

[1] Brandão, F. (2013). Arc-flow results. http://www.dcc.fc.up.pt/~fdabrandao/research/vpsolver/results/

[2] Brandão, F. and Pedroso, J. P. (2013). Bin Packing and Related Problems: General Arc-flow Formulation with Graph Compression. Technical Report DCC-2013-08, Faculdade de Ciências da Universidade do Porto, Portugal.