Integer programming models and exact methods for the two-dimensional two-staged knapsack problem

강수호/Kang Suho

Seoul National Univ.

kangsuho0301@snu.ac.kr

Nov 13, 2020

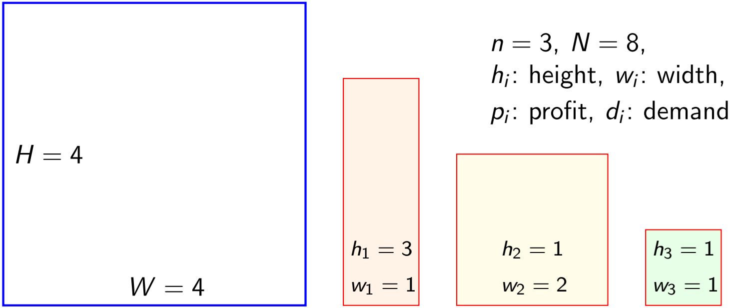
Contents

Contents

Introduction

Two-dimensional Two-staged Knapsack Problem

$n = 3,\ N = 8,$
h_i: height, w_i: width,
p_i: profit, d_i: demand



Only one plate is given

$d_1 = 3$, $p_1 = 7$

$d_2 = 2$, $p_2 = 10$

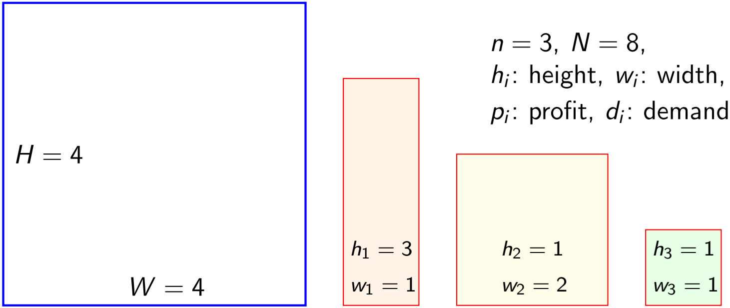
$d_3 = 3$, $p_3 = 2$

A General Two-dimensional Knapsack Problem

What is the maximum profit obtained by cutting small items from the large plate?

Two-dimensional Two-staged Knapsack Problem

$n = 3,\ N = 8,$
h_i: height, w_i: width,
p_i: profit, d_i: demand



Only one plate is given

$d_1 = 3$, $p_1 = 7$

$d_2 = 2$, $p_2 = 10$

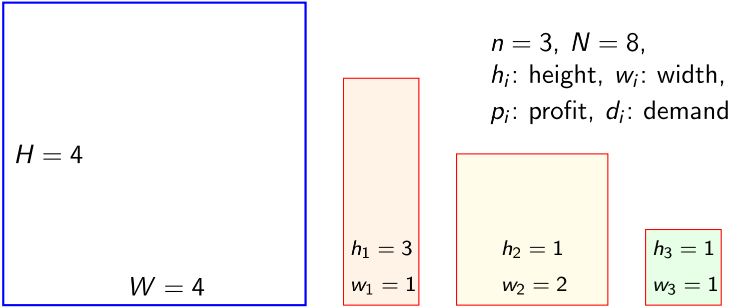
$d_3 = 3$, $p_3 = 2$

A **Two-dimensional Two-staged** Knapsack Problem

What is the maximum profit obtained by cutting small items from the large plate using **two-stage guillotine cuts**?

Examples of Two-staged Cutting

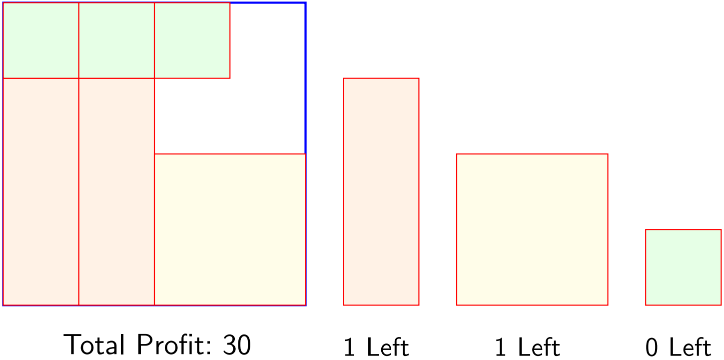
$n = 3, \; N = 8,$
h_i: height, w_i: width,
p_i: profit, d_i: demand



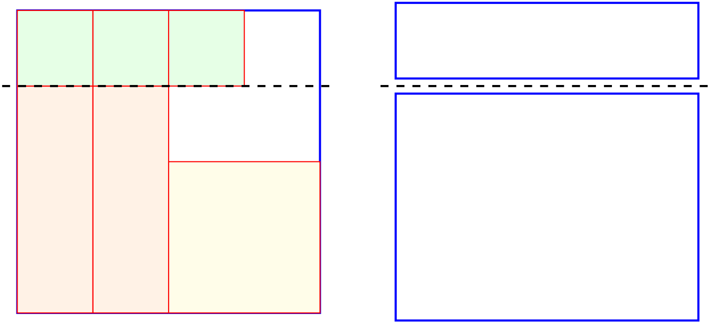
Only one plate is given

$d_1 = 3$, $p_1 = 7$ | $d_2 = 2$, $p_2 = 10$ | $d_3 = 3$, $p_3 = 2$

Examples of Two-staged Cutting



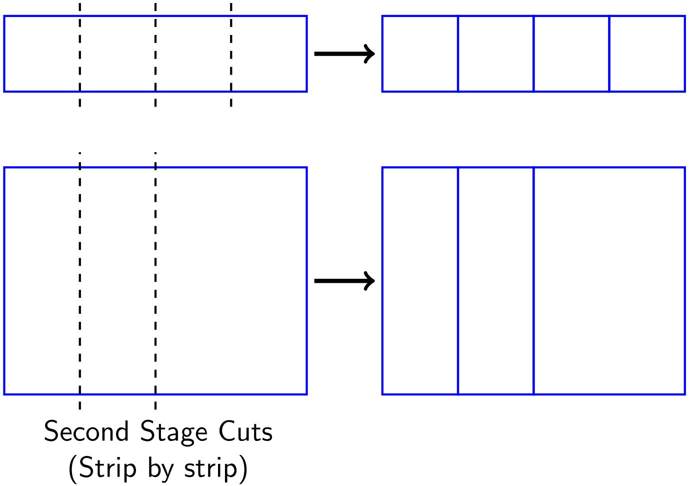
Examples of Two-staged Cutting



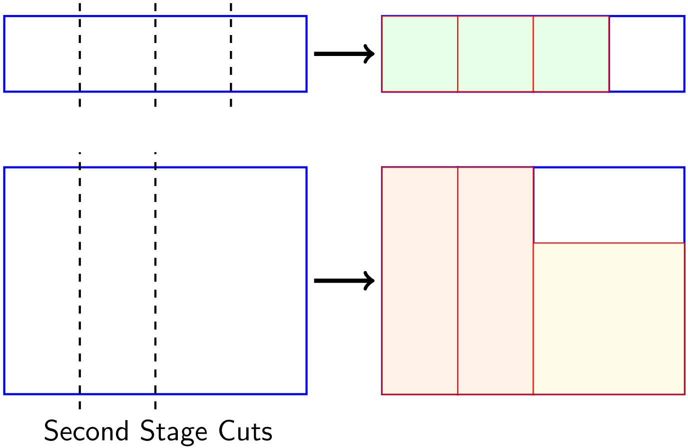
First Stage Cuts

Strips (Levels)

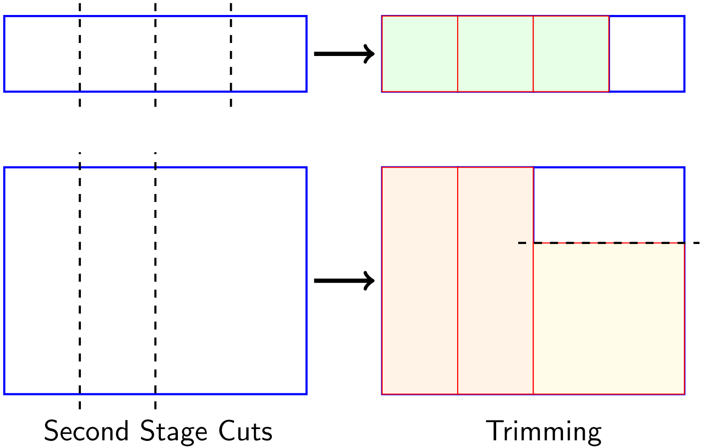
Examples of Two-staged Cutting



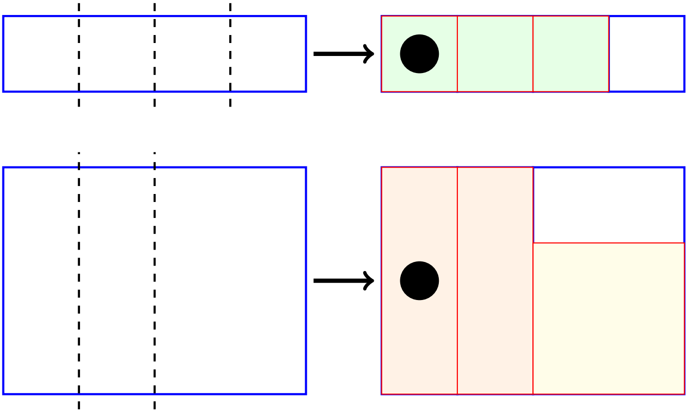
Examples of Two-staged Cutting



Examples of Two-staged Cutting



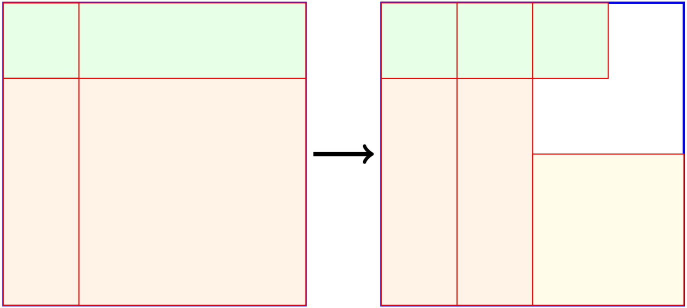
Examples of Two-staged Cutting



Black: Strip Defining Items

Examples of Two-staged Cutting

1. Set strip-defining items first.
2. Locate the rest of the items (with lower heights) into each strip.



Second Stage Cuts → Trimming

Two-dimensional Two-staged Knapsack Problem

- Industrial guillotine cutters: **efficiency** and **accuracy**.
  - One of the restrictions is two-staged guillotine cutting.

Two-dimensional Two-staged Knapsack Problem

- Industrial guillotine cutters: **efficiency** and **accuracy**.
  - One of the restrictions is two-staged guillotine cutting.
- Close relationship with the two-dimensional two-staged guillotine cutting stock problem
  - Sharing the same two-staged guillotine cutting constraint
  - Knapsack problem: a **slave problem**

Two-dimensional Two-staged Knapsack Problem

- Industrial guillotine cutters: **efficiency** and **accuracy**.
    - One of the restrictions is two-staged guillotine cutting.
- Close relationship with the two-dimensional two-staged guillotine cutting stock problem
    - Sharing the same two-staged guillotine cutting constraint
    - Knapsack problem: a **slave problem**
- NP-hard in a strong sense
    - Reduction from the 3-PARTITION problem
    - Unanswered research issues exist yet.

Literature Review

1. Integer programming models:
   - Gilmore and Gomory (1965) [12]: Strip Packing Model
   - Lodi and Monaci (2004) [18]: Level Packing Model

Literature Review

1. Integer programming models:
   - Gilmore and Gomory (1965) [12]: Strip Packing Model
   - Lodi and Monaci (2004) [18]: Level Packing Model
2. Exact Methods:
   - Hifi (2001) [13]: Dynamic Programming
   - Belov and Scheithauer (2006) [2]: Branch-and-cut-and-price Algorithm

Literature Review

1. Integer programming models:
   - Gilmore and Gomory (1965) [12]: Strip Packing Model
   - Lodi and Monaci (2004) [18]: Level Packing Model
2. Exact Methods:
   - Hifi (2001) [13]: Dynamic Programming
   - Belov and Scheithauer (2006) [2]: Branch-and-cut-and-price Algorithm
3. Heuristics:
   - Hifi and M'Hallah (2006) [14]: Strip Generation Algorithm
   - Alvarez-Valdes *et al.* (2007) [1]: Path Relinking Methods.

Contributions

1 Introduced new formulations for the problem based on concepts proposed for the two-dimensional two-staged cutting stock problem

Contributions

1. Introduced new formulations for the problem based on concepts proposed for the two-dimensional two-staged cutting stock problem
2. Proved the existence of polynomial-size formulation

Contributions

1. Introduced new formulations for the problem based on concepts proposed for the two-dimensional two-staged cutting stock problem
2. Proved the existence of polynomial-size formulation
3. Established a hierarchy of the LP-relaxation values

Contributions

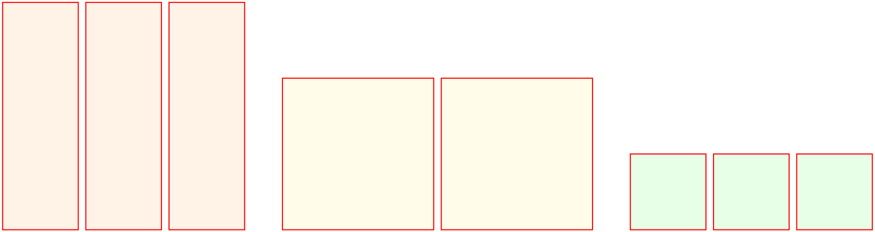
1. Introduced new formulations for the problem based on concepts proposed for the two-dimensional two-staged cutting stock problem
2. Proved the existence of polynomial-size formulation
3. Established a hierarchy of the LP-relaxation values
4. Developed some exact methods and tested them computationally.

Integer Programming Models

Existing Models: (1) A Level Packing Model (LM)

- Lodi and Monaci (2004) [18]

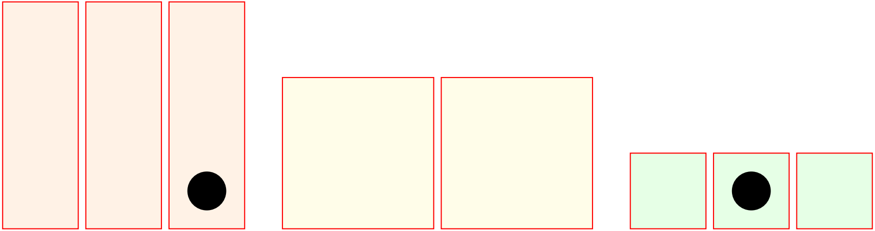
1. Determine which items will be used as strip-defining items.
2. Then, pack the rest of the items.



Existing Models: (1) A Level Packing Model (LM)

- Lodi and Monaci (2004) [18]

1. Determine which items will be used as strip-defining items.
2. Then, pack the rest of the items.



Existing Models: (1) A Level Packing Model (LM)

$$
\begin{aligned}
\text{LM}: \quad \text{maximize} \quad & \sum_{j=1}^{N} p_{\beta_k} \sum_{k=1}^{j} x_{jk} \\
\text{subject to} \quad & \sum_{k=1}^{j} x_{jk} \leq 1, \quad \forall j \in \{1, \ldots, N\} \\
& \sum_{j=k+1}^{N} w_{\beta_j} x_{jk} \leq (W - w_{\beta_k}) x_{kk}, \quad \forall k \in \{1, \ldots, N\} \\
& \sum_{k=1}^{N} h_{\beta_k} x_{kk} \leq H, \\
& x_{jk} \in \{0, 1\}, \quad \forall k \in \{1, \ldots, N\}, \quad \forall j \in \{k, \ldots, N\}.
\end{aligned}
$$

- The size of the formulation depends on the total numbers of items N, not the number of item types n. (Pseudo-polynomial-size model)

Modification of a Level Packing Model

We add the following set of valid inequalities to LM:

$$x_{jk} \leq x_{kk}, \quad \forall k \in \{1, \dots, N\}, \quad \forall j \in \{k+1, \dots, N\}.$$

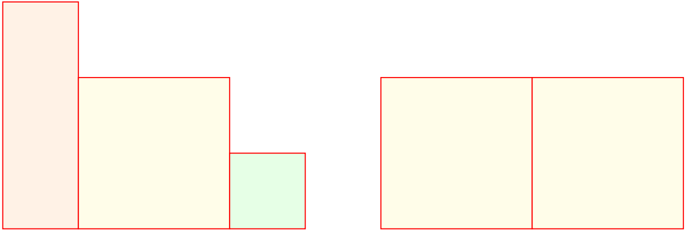
(In a strip, each item should not be used more than the strip-defining item.)

- Improve the quality of the LP-relaxation value
- Ease analyzing the relationship between other models.

Existing Models: (2) A Strip Packing Model (PM)

- Gilmore and Gomory (1965) [12]

1. Width patterns: the combination of items whose total width is below W.
2. One width pattern $\rightarrow$ One strip
3. Pack these width patterns with their total height below H.



Width Pattern $(1, 1, 1)$, Height: 3

Width Pattern $(0, 2, 0)$, Height: 2

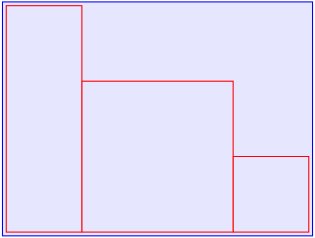
Existing Models: (2) A Strip Packing Model (PM)

- Gilmore and Gomory (1965) [12]

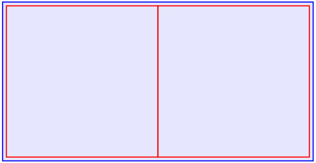
1 Width patterns: the combination of items whose total width is below W.

2 One width pattern $\rightarrow$ One strip

3 Pack these width patterns with their total height below H.



Width Pattern $(1, 1, 1)$,
Height: 3



Width Pattern $(0, 2, 0)$,
Height: 2

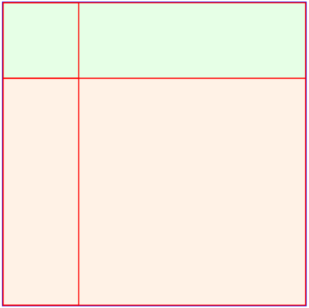
Existing Models: (2) A Strip Packing Model (PM)

$$
\begin{aligned}
\text{PM}: \quad & \text{maximize} && \sum_{q \in P_W(d)} \sum_{i \in I_n} p_i a_{qi} x_q \\
& \text{subject to} && \sum_{q \in P_W(d)} a_{qi} x_q \le d_i, \quad \forall i \in I_n, \\
& && \sum_{q \in P_W(d)} h_{t(q)} x_q \le H, \\
& && x_q \in \mathbf{Z}_+, \quad \forall q \in P_W(d).
\end{aligned}
$$

- Exponentially many width patterns exist. (Exponential-size model)

Proposed Models: (3) A Staged-pattern Model (SM)

1. Mrad *et al.* (2013) [22]
   - Propose the concept of height patterns at the two-staged two-dimensional cutting stock problem
2. Height patterns: the combination of strip-defining items whose total height is below H.
3. Choose adequate width patterns to complete the cutting.



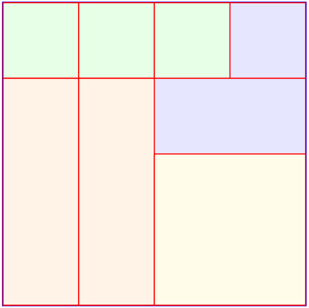
Height Pattern $(1, 0, 1)$

Proposed Models: (3) A Staged-pattern Model (SM)

1 Mrad *et al.* (2013) [22]
- Propose the concept of height patterns at the two-staged two-dimensional cutting stock problem

2 Height patterns: the combination of strip-defining items whose total height is below H.

3 Choose adequate width patterns to complete the cutting.



Width Pattern $(0, 0, 3), (2, 1, 0)$

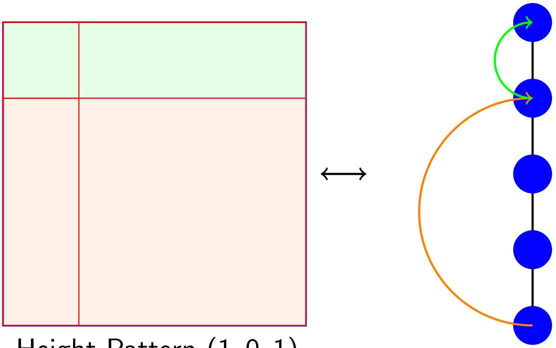
New Models: (3) A Staged-pattern Model (SM)

$$
\begin{aligned}
\text{SM}: \quad & \text{maximize} && \sum_{q \in P_W(d)} \sum_{i \in I_n} p_i a_{qi} x_q \\
& \text{subject to} && \sum_{q \in P_W(d)} a_{qi} x_q \le d_i, && \forall i \in I_n, \\
& && \sum_{r \in P_H(d)} b_{ri} y_r \ge \sum_{q \in P_W(d), t(q)=i} x_q, && \forall i \in I_n, \\
& && \sum_{r \in P_H(d)} y_r \le 1, \\
& && x_q \in \mathbf{Z}_+, \quad \forall q \in P_W(d), \\
& && y_r \in \mathbf{Z}_+, \quad \forall r \in P_H(d).
\end{aligned}
$$

- Exponentially many width patterns and height patterns exist. (Exponential-size model)

New Models: (4) An Arc-flow Model (AF)

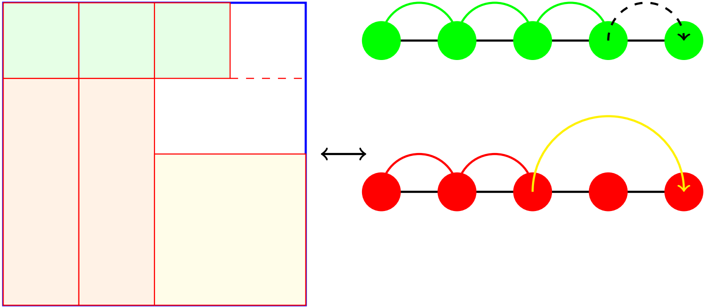
1. Macedo *et al.* [20]:
   - Extend the concept of arc-flow model to the two-staged two-dimensional cutting stock problem
2. Represent patterns as the flow of the certain graph



Height Pattern $(1, 0, 1)$ $\longleftrightarrow$ Height Pattern Graph

New Models: (4) An Arc-flow Model (AF)

1. Macedo *et al.* [20]:
   - Extend the concept of arc-flow model to the two-staged two-dimensional cutting stock problem
2. Represent patterns as the flow of the certain graph



Width Pattern (0, 0, 3)
Width Pattern (2, 1, 0)

Width Pattern Graphs

New Models: (4) An Arc-flow Model (AF)

$$
\begin{aligned}
\text{AF}: \quad \text{maximize} \quad & \sum_{j \in I_n} \sum_{(a,b,i) \in A^j} \pi^j_{(a,b,i)} x^j_{(a,b,i)} \\
\text{subject to} \quad & \sum_{(a,b,i) \in A^0} x^0_{(a,b,i)} - \sum_{(b,c,k) \in A^0} x^0_{(b,c,k)} = \begin{cases} -1 & \text{if } b = 0 \\ 0 & \text{if } b = 1, \dots, H-1 \\ 1 & \text{if } b = H \end{cases}, \\
& \sum_{(c,c+h_j,k) \in A^0} x^0_{(c,c+h_j,k)} - z^j = 0, \quad \forall j \in I_n, \\
& \sum_{(d,e,i) \in A^j} x^j_{(d,e,i)} - \sum_{(e,f,k) \in A^j} x^j_{(e,f,k)} \\
& = \begin{cases} -z^j & \text{if } e = 0 \\ 0 & \text{if } e = 1, \dots, W-1 \\ z^j & \text{if } e = W \end{cases}, \quad \forall j \in I_n, \\
& \sum_{j \in I_n} \sum_{(f,f+w_i,i) \in A^j} x^j_{(f,f+w_i,i)} \le d_i, \quad \forall i \in I_n, \\
& \text{All variables are nonnegative integers.}
\end{aligned}
$$

- The size of the formulation depends on H and W. (Pseudo-polynomial-size model)

Theoretical Analysis

Existence of a Polynomial-size Formulation

Eisenbrand and Shmonin [8] provides the upper bound on numbers of nonzero components in the optimal solution in integer linear programming problems.

Lemma 1

Let p_{max} and d_{max} indicate the maximum value of components in given p and d, respectively. Then, there exists an optimal solution with at most $M = \lceil log_2(n) + log_2(p_{max}) + (n+1)log_2(d_{max}) + log_2(H) \rceil$ different types of width patterns.

- Split the integer variables into binary variables.
- Transform the multiplication of binary variables into linear constraints with a new binary variable.

Existence of a Polynomial-size Formulation: POLY

$$
\begin{aligned}
\text{maximize} \quad & \sum_{i=1}^{n}\sum_{m=1}^{M}\sum_{k=1}^{\hat{D}}\sum_{l=1}^{\hat{D}} 2^{k+l-2} p_i s_{ikml} \\
\text{subject to} \quad & \sum_{m=1}^{M}\sum_{k=1}^{\hat{D}}\sum_{l=1}^{\hat{D}} 2^{k+l-2} s_{ikml} \le d_i, \quad \forall i \\
& \sum_{i=1}^{n}\sum_{k=1}^{\hat{D}} 2^{k-1} w_i \bar{q}_{ik}^{m} \le W, \quad \forall m, \\
& \sum_{m=1}^{M}\sum_{l=1}^{\hat{D}}\sum_{t=1}^{\hat{H}} 2^{l+t-2} r_{mlt} \le H, \\
& \sum_{k=1}^{\hat{D}} 2^{k-1} \bar{q}_{ik}^{m} \le d_i z_i^{m}, \quad \forall i, m,
\end{aligned}
$$

$$
\begin{aligned}
& \bar{q}_{ik}^{m} \le z_i^{m}, \quad \forall i, k, m, \\
& \sum_{t=1}^{\hat{H}} 2^{t-1} \bar{H}_{mt} \ge h_i z_i^{m}, \quad \forall i, m, \\
& s_{ikml} \ge \bar{q}_{ik}^{m} + x_{ml} - 1, \quad \forall i, k, m, l, \\
& s_{ikml} \le \bar{q}_{ik}^{m}, \quad \forall i, k, m, l, \\
& s_{ikml} \le x_{ml}, \quad \forall i, k, m, l, \\
& r_{mlt} \ge x_{ml} + \bar{H}_{mt} - 1, \quad \forall m, l, t, \\
& r_{mlt} \le x_{ml}, \quad \forall m, l, t, \\
& r_{mlt} \le \bar{H}_{mt}, \quad \forall m, l, t, \\
& \text{all variables } s, \bar{q}, \bar{H}, r, z \text{ are binary.}
\end{aligned}
$$

- Many logical constraints $\rightarrow$ weak upper bound
- The first polynomial-sized model
  - $\mathcal{O}(n \log_2(d_{\max})^2 (n \log_2(d_{\max}) + \log_2(p_{\max}) + \log_2(H)))$

Upper bounds provided by the LP-relaxations

Optimal objective value: z^*
Optimal objective value of the LP relaxation of model "M": z^{M}

Theorem 1

$z^* \leq z^{\mathrm{SM}} \leq z^{\mathrm{PM}} \leq z^{\mathrm{LM}}$

Theorem 2

$z^* \leq z^{\mathrm{SM}} \leq z^{\mathrm{AF}}$

Upper bounds provided by the LP-relaxations

Optimal objective value: z^*
Optimal objective value of the LP relaxation of model "M": z^{M}

Theorem 1

$z^* \leq z^{\mathrm{SM}} \leq z^{\mathrm{PM}} \leq z^{\mathrm{LM}}$

Theorem 2

$z^* \leq z^{\mathrm{SM}} \leq z^{\mathrm{AF}}$

LM Explicitly choose strip-defining items and then construct strips.

Upper bounds provided by the LP-relaxations

Optimal objective value: z^*
Optimal objective value of the LP relaxation of model "M": z^{M}

Theorem 1

$z^* \leq z^{\mathrm{SM}} \leq z^{\mathrm{PM}} \leq z^{\mathrm{LM}}$

Theorem 2

$z^* \leq z^{\mathrm{SM}} \leq z^{\mathrm{AF}}$

LM Explicitly choose strip-defining items and then construct strips.

PM Packing predefined strips.

Upper bounds provided by the LP-relaxations

Optimal objective value: z^*
Optimal objective value of the LP relaxation of model "M": z^{M}

Theorem 1

$z^* \leq z^{\text{SM}} \leq z^{\text{PM}} \leq z^{\text{LM}}$

Theorem 2

$z^* \leq z^{\text{SM}} \leq z^{\text{AF}}$

- **LM** Explicitly choose strip-defining items and then construct strips.
- **PM** Packing predefined strips.
- **SM** Packing strips with a predefined height pattern.

Upper bounds provided by the LP-relaxations

Optimal objective value: z^*
Optimal objective value of the LP relaxation of model "M": z^{M}

Theorem 1

$z^* \leq z^{\mathrm{SM}} \leq z^{\mathrm{PM}} \leq z^{\mathrm{LM}}$

Theorem 2

$z^* \leq z^{\mathrm{SM}} \leq z^{\mathrm{AF}}$

- **LM** Explicitly choose strip-defining items and then construct strips.
- **PM** Packing predefined strips.
- **SM** Packing strips with a predefined height pattern.
- **AF** Use graphs to indicate which patterns are used.

Upper bounds provided by the LP-relaxations

Because of the knapsack structure, the following theorem holds:

Theorem 3

$z^{\mathrm{LM}} \le 2z^{\mathrm{PM}} \le 4z^{\mathrm{SM}}$.

Proof.

$z^{\mathrm{LM}} \le 2z^{\mathrm{PM}}$: Compensation for a partly used item in each strip

$z^{\mathrm{PM}} \le 2z^{\mathrm{SM}}$: Compensation for a partly used strip in a plate □

Also, there exists a tight example of the above inequality.

Upper bounds provided by the LP-relaxations

Because of the knapsack structure, the following theorem holds:

Theorem 3

$z^{\mathrm{LM}} \leq 2z^{\mathrm{PM}} \leq 4z^{\mathrm{SM}}$.

Proof.

$z^{\mathrm{LM}} \leq 2z^{\mathrm{PM}}$: Compensation for a partly used item in each strip

$z^{\mathrm{PM}} \leq 2z^{\mathrm{SM}}$: Compensation for a partly used strip in a plate □

Also, there exists a tight example of the above inequality.

$$\rightarrow z^* \leq z^{\mathrm{SM}} \leq z^{\mathrm{PM}} \leq z^{\mathrm{LM}} \leq 2z^{\mathrm{PM}} \leq 4z^{\mathrm{SM}}$$

Computational Analysis

Exact Methods

- LM: Branch-and-cut (Delayed constraint generation)

Exact Methods

- LM: Branch-and-cut (Delayed constraint generation)
- PM: Branch-and-price
- SM: Branch-and-price

Exact Methods

- LM: Branch-and-cut (Delayed constraint generation)
- PM: Branch-and-price
- SM: Branch-and-price
- AF: Branch-and-bound
- POLY: Branch-and-bound

Delayed Constraint Generation of LM

We add the following set of valid inequalities to the original form of LM:

$$x_{jk} \le x_{kk}, \quad \forall k \in \{1, \ldots, N\}, \quad \forall j \in \{k+1, \ldots, N\}.$$

- The number of added inequalities: $\mathcal{O}(N^2)$

Delayed Constraint Generation of LM

We add the following set of valid inequalities to the original form of LM:

$$x_{jk} \leq x_{kk}, \quad \forall k \in \{1, \dots, N\}, \quad \forall j \in \{k+1, \dots, N\}.$$

- The number of added inequalities: $\mathcal{O}(N^2)$

1 Discard the inequalities at the beginning.

Delayed Constraint Generation of LM

We add the following set of valid inequalities to the original form of LM:

$$x_{jk} \leq x_{kk}, \quad \forall k \in \{1, \ldots, N\}, \quad \forall j \in \{k+1, \ldots, N\}.$$

- The number of added inequalities: $\mathcal{O}(N^2)$

1 Discard the inequalities at the beginning.

2 At each node, solve the LP-relaxation and check whether violated inequalities exist.

Delayed Constraint Generation of LM

We add the following set of valid inequalities to the original form of LM:

$$x_{jk} \leq x_{kk}, \quad \forall k \in \{1, \ldots, N\}, \quad \forall j \in \{k+1, \ldots, N\}.$$

- The number of added inequalities: $\mathcal{O}(N^2)$

1. Discard the inequalities at the beginning.
2. At each node, solve the LP-relaxation and check whether violated inequalities exist.
3. Add violated inequalities to the descendants of the node.

Branch-and-price Algorithms of PM and SM

- The number of pattern variables: $\mathcal{O}(2^n)$.

Branch-and-price Algorithms of PM and SM

- The number of pattern variables: $\mathcal{O}(2^n)$.

1 Start with the basic pattern variables.

Branch-and-price Algorithms of PM and SM

- The number of pattern variables: $\mathcal{O}(2^n)$.

1. Start with the basic pattern variables.
2. For each node, solve the LP-relaxation and check whether more pattern variables are needed.

Branch-and-price Algorithms of PM and SM

- The number of pattern variables: $\mathcal{O}(2^n)$.

1. Start with the basic pattern variables.
2. For each node, solve the LP-relaxation and check whether more pattern variables are needed.
3. If needed, generate new pattern variables and repeat the procedure.

Computational Experiments

- Solvers offered by Xpress 8.9 [9]
- Intel(R) Core(TM) i7-4770 CPU @ 3.10GHz and 16GB of RAM
- Time Limit: 600 s.
- Benchmark instances proposed by Hifi and Roucairol (2001) [15]:
    - Small: 16 instances ($40 \times 40 - 130 \times 130$)
    - Large: 20 instances ($200 \times 200 - 900 \times 900$)

Computational Experiments

- Solvers offered by Xpress 8.9 [9]
- Intel(R) Core(TM) i7-4770 CPU @ 3.10GHz and 16GB of RAM
- Time Limit: 600 s.
- Benchmark instances proposed by Hifi and Roucairol (2001) [15]:
  - Small: 16 instances ($40 \times 40 - 130 \times 130$)
  - Large: 20 instances ($200 \times 200 - 900 \times 900$)

1 # of instances solved to optimality within the time limit

Computational Experiments

- Solvers offered by Xpress 8.9 [9]
- Intel(R) Core(TM) i7-4770 CPU @ 3.10GHz and 16GB of RAM
- Time Limit: 600 s.
- Benchmark instances proposed by Hifi and Roucairol (2001) [15]:
  - Small: 16 instances ($40 \times 40 - 130 \times 130$)
  - Large: 20 instances ($200 \times 200 - 900 \times 900$)

1. # of instances solved to optimality within the time limit
2. Gap between the lower bound and z^* (Unsolved instances):

Computational Experiments

- Solvers offered by Xpress 8.9 [9]
- Intel(R) Core(TM) i7-4770 CPU @ 3.10GHz and 16GB of RAM
- Time Limit: 600 s.
- Benchmark instances proposed by Hifi and Roucairol (2001) [15]:
  - Small: 16 instances ($40 \times 40 - 130 \times 130$)
  - Large: 20 instances ($200 \times 200 - 900 \times 900$)

1. # of instances solved to optimality within the time limit
2. Gap between the lower bound and z^* (Unsolved instances):
   - IP gap $= \frac{\text{(Optimal objective value)} - \text{(Best Lower Bound)}}{\text{(Optimal objective value)}} \times 100(\%)$

Computational Experiments

- Solvers offered by Xpress 8.9 [9]
- Intel(R) Core(TM) i7-4770 CPU @ 3.10GHz and 16GB of RAM
- Time Limit: 600 s.
- Benchmark instances proposed by Hifi and Roucairol (2001) [15]:
  - Small: 16 instances ($40 \times 40 - 130 \times 130$)
  - Large: 20 instances ($200 \times 200 - 900 \times 900$)

1. # of instances solved to optimality within the time limit
2. Gap between the lower bound and z^* (Unsolved instances):
   - IP gap $= \frac{\text{(Optimal objective value)} - \text{(Best Lower Bound)}}{\text{(Optimal objective value)}} \times 100(\%)$
3. Gap between the upper bound and z^*

Computational Experiments

- Solvers offered by Xpress 8.9 [9]
- Intel(R) Core(TM) i7-4770 CPU @ 3.10GHz and 16GB of RAM
- Time Limit: 600 s.
- Benchmark instances proposed by Hifi and Roucairol (2001) [15]:
  - Small: 16 instances ($40 \times 40 - 130 \times 130$)
  - Large: 20 instances ($200 \times 200 - 900 \times 900$)

1. # of instances solved to optimality within the time limit
2. Gap between the lower bound and z^* (Unsolved instances):
   - IP gap $= \frac{\text{(Optimal objective value)} - \text{(Best Lower Bound)}}{\text{(Optimal objective value)}} \times 100(\%)$
3. Gap between the upper bound and z^*
   - LP gap $= \frac{\text{(LP-relaxation value)} - \text{(Optimal objective value)}}{\text{(Optimal objective value)}} \times 100(\%)$

Results: Small Instances

- Except for POLY, all models solved all instances to optimality

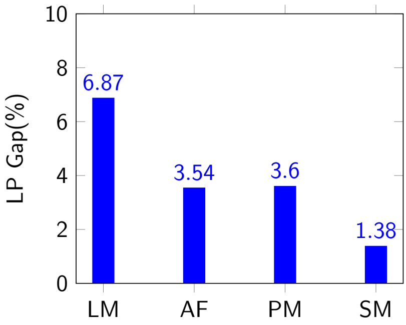


Figure: Average LP gaps.

Results: Small Instances

- Except for POLY, all models solved all instances to optimality

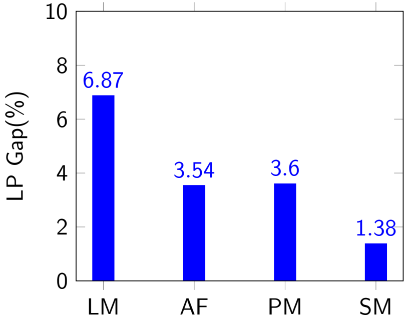


Figure: Average LP gaps.

- z^{AF} and z^{PM}: incomparable

Results: Small Instances

- Except for POLY, all models solved all instances to optimality

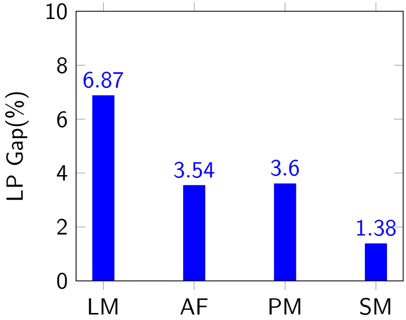


Figure: Average LP gaps.

- z^{AF} and z^{PM}: incomparable
- Far from $z^{\text{LM}} \leq 2z^{\text{PM}} \leq 4z^{\text{SM}}$

Results: Small Instances

- Except for POLY, all models solved all instances to optimality

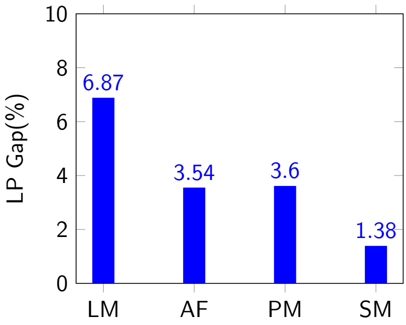


Figure: Average LP gaps.

- z^{AF} and z^{PM}: incomparable
- Far from $z^{\mathrm{LM}} \leq 2z^{\mathrm{PM}} \leq 4z^{\mathrm{SM}}$
- LM: fastest

Results: Small Instances

- Except for POLY, all models solved all instances to optimality

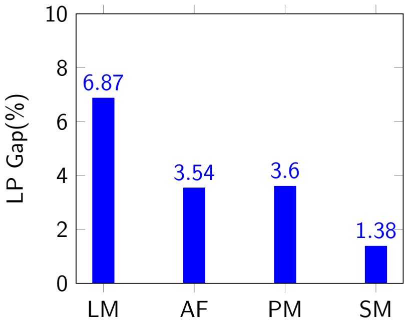


Figure: Average LP gaps.

- z^{AF} and z^{PM}: incomparable
- Far from $z^{\mathrm{LM}} \leq 2z^{\mathrm{PM}} \leq 4z^{\mathrm{SM}}$
- LM: fastest
- LP Gap of POLY: 517.5 (%)

Results: Large Instances

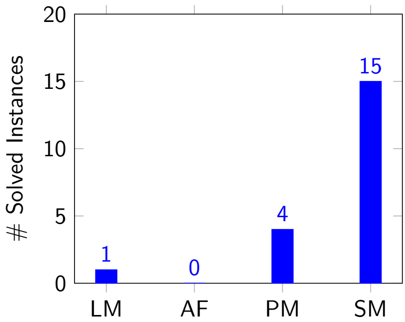


Figure: The number of solved instances.

- AF, LM: vulnerable to n and N

Results: Large Instances

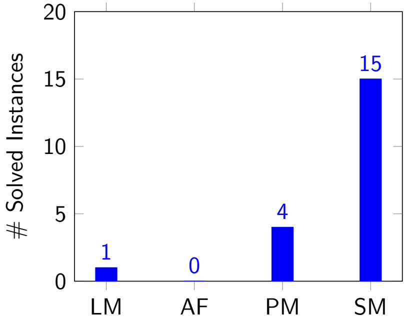


Figure: The number of solved instances.

- AF, LM: vulnerable to n and N
- SM: somtimes LP-relaxation solution → optimal solution

Results: Large Instances

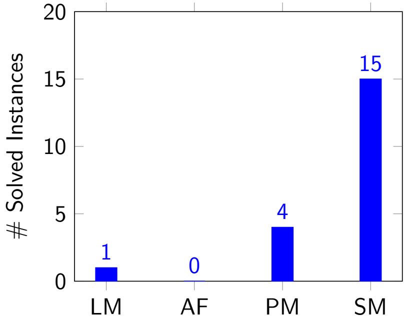


Figure: The number of solved instances.

- AF, LM: vulnerable to n and N
- SM: somtimes LP-relaxation solution $\rightarrow$ optimal solution
- Effectiveness of height patterns

Results: Large Instances

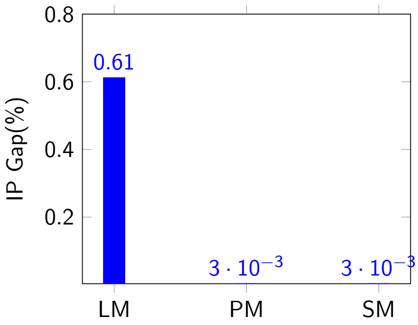


Figure: Average IP Gaps.

- AF: No lower bounds obtained

Results: Large Instances

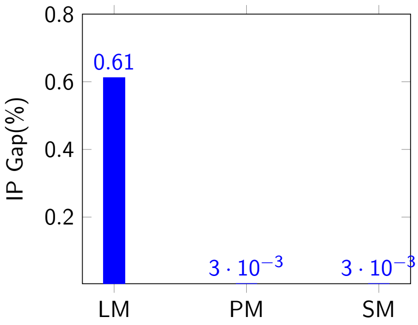


Figure: Average IP Gaps.

- AF: No lower bounds obtained
- Pattern-based models:
  - failed to prove optimality → but provide a useful solution

Results: Large Instances

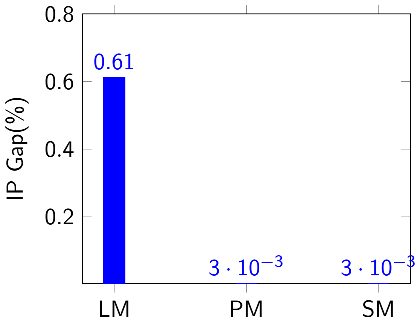


Figure: Average IP Gaps.

- AF: No lower bounds obtained
- Pattern-based models:
  - failed to prove optimality $\rightarrow$ but provide a useful solution
  - quickly find a lower bound

Results: Large Instances

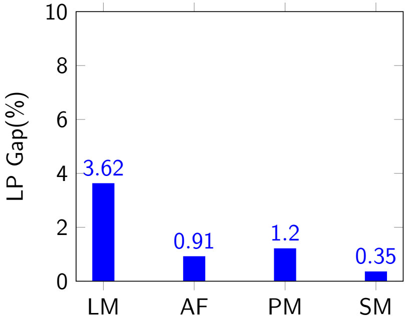


Figure: Average LP gaps.

- AF: decent upper bound

Results: Large Instances

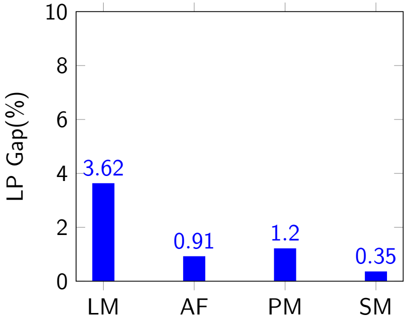


Figure: Average LP gaps.

- AF: decent upper bound
  - but requires a lot of time

Results: Large Instances

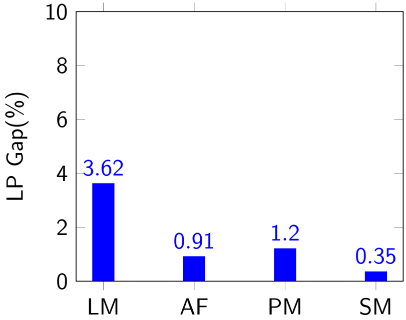


Figure: Average LP gaps.

- AF: decent upper bound
  - but requires a lot of time
- SM: very tight upper bound

Results: Large Instances

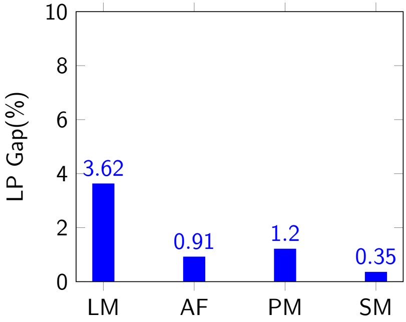


Figure: Average LP gaps.

- AF: decent upper bound
  - but requires a lot of time
- SM: very tight upper bound
- Similar to the case of small instances

Conclusion

Conclusion and Further research

- Introduced formulations and established their theoretical hierarchy

Conclusion and Further research

- Introduced formulations and established their theoretical hierarchy
- SM: competitive theoretical and computational performance

Conclusion and Further research

- Introduced formulations and established their theoretical hierarchy
- SM: competitive theoretical and computational performance
- POLY: the first polynomial-size formulation of the problem

Conclusion and Further research

- Introduced formulations and established their theoretical hierarchy
- SM: competitive theoretical and computational performance
- POLY: the first polynomial-size formulation of the problem
  - But its real usage is not recommended.

Conclusion and Further research

- Introduced formulations and established their theoretical hierarchy
- SM: competitive theoretical and computational performance
- POLY: the first polynomial-size formulation of the problem
  - But its real usage is not recommended.
- LM, AF: limitation in solving large instances.

Conclusion and Further research

- Introduced formulations and established their theoretical hierarchy
- SM: competitive theoretical and computational performance
- POLY: the first polynomial-size formulation of the problem
    - But its real usage is not recommended.
- LM, AF: limitation in solving large instances.
- z^{AF} and z^{LM}?

Conclusion and Further research

- Introduced formulations and established their theoretical hierarchy
- SM: competitive theoretical and computational performance
- POLY: the first polynomial-size formulation of the problem
  - But its real usage is not recommended.
- LM, AF: limitation in solving large instances.
- z^{AF} and z^{LM}?
- Polynomial-size with a decent upper bounds?

References I

R. Alvarez-Valdes, R. Martí, J. M. Tamarit, and A. Parajón.
Grasp and path relinking for the two-dimensional two-stage cutting-stock problem.
INFORMS Journal on Computing, 19(2):261–272, 2007.

G. Belov and G. Scheithauer.
A branch-and-cut-and-price algorithm for one-dimensional stock cutting and two-dimensional two-stage cutting.
European journal of operational research, 171(1):85–106, 2006.

J. O. Berkey and P. Y. Wang.
Two-dimensional finite bin-packing algorithms.
Journal of the operational research society, 38(5):423–429, 1987.

V. M. Bezerra, A. A. Leao, J. F. Oliveira, and M. O. Santos.
Models for the two-dimensional level strip packing problem–a review and a computational evaluation.
Journal of the Operational Research Society, 71(4):606–627, 2020.

References II

A. Caprara and M. Monaci.
On the two-dimensional knapsack problem.
Operations Research Letters, 32(1):5–14, 2004.

M. Conforti, G. Cornuéjols, G. Zambelli, et al.
Integer programming, volume 271.
Springer, 2014.

M. Dolatabadi, A. Lodi, and M. Monaci.
Exact algorithms for the two-dimensional guillotine knapsack.
Computers & Operations Research, 39(1):48–53, 2012.

F. Eisenbrand and G. Shmonin.
Carathéodory bounds for integer cones.
Operations Research Letters, 34(5):564–568, 2006.

FICO® Xpress Optimization, 2020. [Online].
`https://www.fico.com/`.

References III

F. Furini and E. Malaguti.
Models for the two-dimensional two-stage cutting stock problem with multiple stock size.
Computers & Operations Research, 40(8):1953–1962, 2013.

M. R. Garey and D. S. Johnson.
Computers and intractability, volume 174.
freeman San Francisco, 1979.

P. Gilmore and R. E. Gomory.
Multistage cutting stock problems of two and more dimensions.
Operations research, 13(1):94–120, 1965.

M. Hifi.
Exact algorithms for large-scale unconstrained two and three staged cutting problems.
Computational Optimization and Applications, 18(1):63–88, 2001.

References IV

M. Hifi and R. M'Hallah.
Strip generation algorithms for constrained two-dimensional two-staged cutting problems.
European Journal of Operational Research, 172(2):515–527, 2006.

M. Hifi and C. Roucairol.
Approximate and exact algorithms for constrained (un) weighted two-dimensional two-staged cutting stock problems.
Journal of combinatorial optimization, 5(4):465–494, 2001.

H. Kellerer, U. Pferschy, and D. Pisinger.
Knapsack Problems.
Springer, Berlin, Germany, 2004.

S.-J. Kwon, S. Joung, and K. Lee.
Comparative analysis of pattern-based models for the two-dimensional two-stage guillotine cutting stock problem.
Computers & Operations Research, 109:159–169, 2019.

References V

A. Lodi and M. Monaci.
Integer linear programming models for 2-staged two-dimensional knapsack problems.
Mathematical Programming, 94(2-3):257–278, 2003.

M. E. Lübbecke and J. Desrosiers.
Selected topics in column generation.
Operations research, 53(6):1007–1023, 2005.

R. Macedo, C. Alves, and J. V. De Carvalho.
Arc-flow model for the two-dimensional guillotine cutting stock problem.
Computers & Operations Research, 37(6):991–1001, 2010.

J. Martinovic, G. Scheithauer, and J. V. de Carvalho.
A comparative study of the arcflow model and the one-cut model for one-dimensional cutting stock problems.
European Journal of Operational Research, 266(2):458–471, 2018.

References VI

M. Mrad, I. Meftahi, and M. Haouari.
A branch-and-price algorithm for the two-stage guillotine cutting stock problem.
Journal of the Operational Research Society, 64(5):629–637, 2013.

E. Silva, F. Alvelos, and J. V. de Carvalho.
An integer programming model for two-and three-stage two-dimensional cutting stock problems.
European Journal of Operational Research, 205(3):699–708, 2010.

A. Steinberg.
A strip-packing algorithm with absolute performance bound 2.
SIAM Journal on Computing, 26(2):401–409, 1997.

G. Wäscher, H. Haußner, and H. Schumann.
An improved typology of cutting and packing problems.
European journal of operational research, 183(3):1109–1130, 2007.

The End

Thank you for listening.

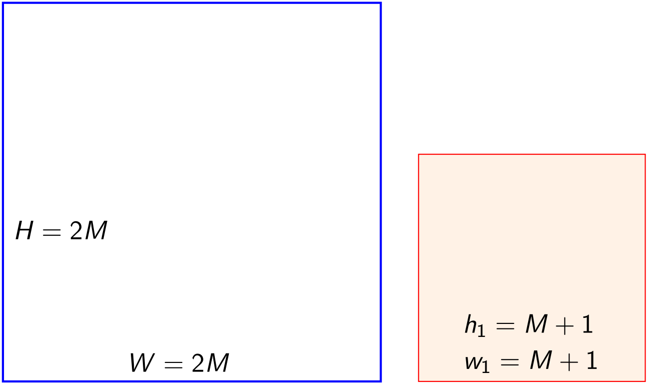
Small Instances

Name	n	W	H	w_{min}	w_{max}	h_{min}	h_{max}	d_{min}	d_{max}	OPT
2	10	40	70	9	31	7	35	1	3	2,535
2s	10	40	70	9	31	7	35	1	3	2,430
3	20	40	70	9	33	11	43	1	4	1,720
3s	20	40	70	9	33	11	43	1	4	2,599
A1s	20	50	60	9	33	11	43	1	4	2,950
A2s	20	60	60	12	33	14	42	1	4	3,423
A3	20	70	80	15	35	14	43	1	4	5,380
A4	20	90	70	9	33	11	43	1	3	5,885
A5	20	132	100	13	69	12	63	1	5	12,553
CHL1	30	132	100	13	69	12	63	1	5	8,360
CHL1s	30	132	100	13	69	12	63	1	5	13,036
CHL2	10	62	55	11	31	9	31	1	3	2,235
CHL2s	10	62	55	11	31	9	31	1	3	3,162
CHL5	10	20	20	1	20	2	14	1	3	363
CHL6	30	130	130	18	69	12	63	1	5	16,572
CHL7	35	130	130	19	57	18	54	1	5	16,728

Large Instances

Name	n	W	H	w_{min}	w_{max}	h_{min}	h_{max}	d_{min}	d_{max}	OPT
ATP30	38	927	152	57	360	7	58	1	9	140,168
ATP31	51	856	964	44	331	50	380	1	9	820,260
ATP32	56	307	124	16	120	6	46	1	9	37,880
ATP33	44	241	983	15	90	52	390	1	9	235,580
ATP34	27	795	456	46	308	22	173	1	9	356,159
ATP35	29	960	649	50	363	34	248	1	9	614,429
ATP36	28	537	244	30	209	20	91	1	9	129,262
ATP37	43	440	881	23	175	51	350	1	9	384,478
ATP38	40	731	358	41	289	19	140	1	9	259,070
ATP39	33	538	501	28	214	48	192	1	9	266,135
ATP40	56	683	138	34	270	6	54	1	9	63,945
ATP41	36	837	367	43	326	32	144	1	9	202,305
ATP42	59	167	291	8	65	21	114	1	9	32,589
ATP43	49	362	917	19	143	46	362	1	9	208,998
ATP44	39	223	496	11	88	29	193	1	9	70,940
ATP45	33	188	578	9	74	49	228	1	9	74,205
ATP46	42	416	514	23	157	40	204	1	9	146,402
ATP47	43	393	554	25	156	32	215	1	9	144,317
ATP48	34	931	254	47	355	18	99	1	9	165,428
ATP49	25	759	449	42	301	23	157	1	9	206,965

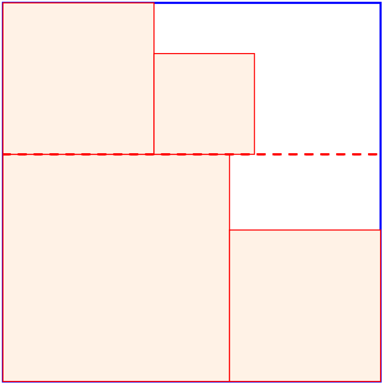
Tight Example



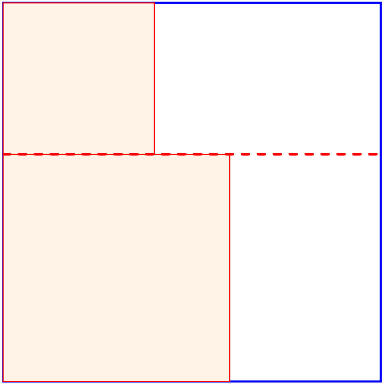
Optimal Objective value: 1

$d_1 = 4$
$p_1 = 1$

Tight Example



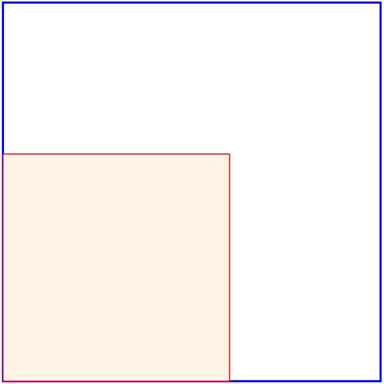
LM $\rightarrow$ 4



PM $\rightarrow$ 2

Feasible Width Pattern: (1)

Tight Example



SM $\rightarrow$ 1

Feasible Width Pattern: (1)

Feasible Height Pattern: (1)