A two-dimensional multi-criteria bin packing problem in the production of printed circuit boards

Jochen Rethmann and Steffen Goebbels

Niederrhein University of Applied Sciences - Institute for Pattern Recognition, Faculty of Electrical Engineering and Computer Science

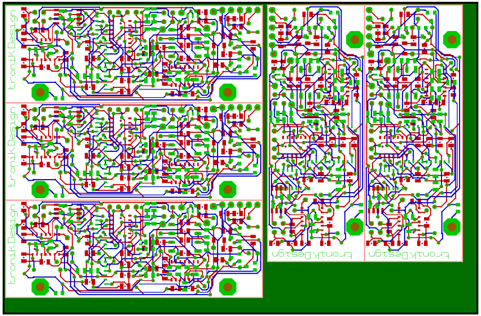
OR 2024 Munich

Hochschule Niederrhein
University of Applied Sciences

iPattern
Institut für Mustererkennung
Institute for Pattern Recognition

Outline

- **The problem** (Precoplat Präzisions-Leiterplatten-Technik GmbH)
- Mixed Integer Linear Program (MILP) formulation
- Results

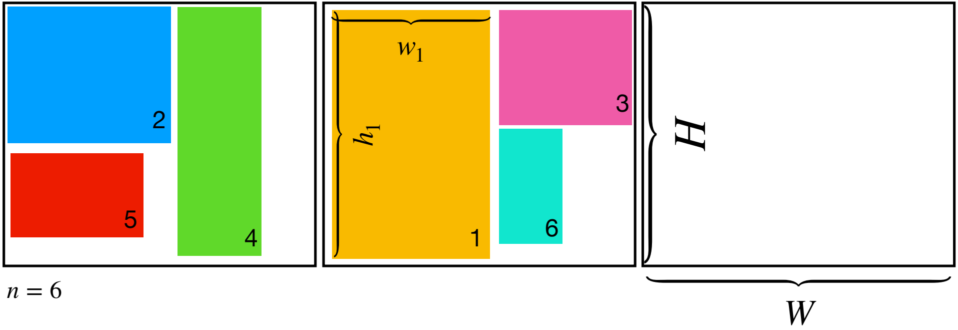


basierend auf “Ulfbastel”, `https://de.wikipedia.org/wiki/Leiterplatte#/media/Datei:Lp3b.png`,

Hochschule Niederrhein
University of Applied Sciences
iPattern
Institut für Mustererkennung
Institute for Pattern Recognition

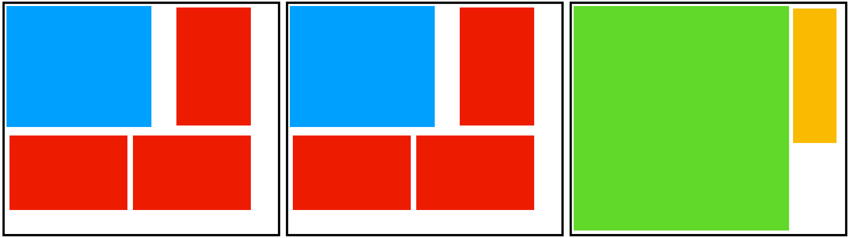
The original 2D bin packing problem

Classical Problem: n heterogeneous small rectangular items $j \in [n] := \{1, \ldots, n\}$ with width w_j and height h_j are given. Place them into the minimum number of bins with width W and height H. Items may be rotated by 90 degrees.



Hochschule Niederrhein
University of Applied Sciences
iPattern
Institut für Mustererkennung
Institute for Pattern Recognition

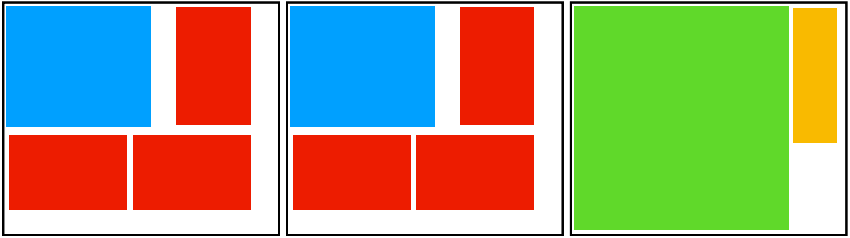
A variant of the 2D bin packing problem



We extend the classical problem:

- Minimise number of bins, maximise occupied area, minimise number of different bin patterns (layouts) to reduce changeover times, minimise costs

A variant of the 2D bin packing problem

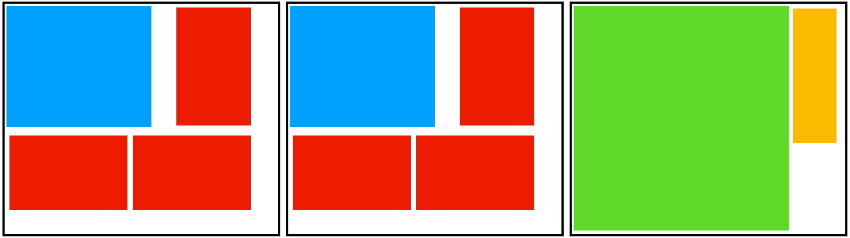


We extend the classical problem:

- Minimise number of bins, maximise occupied area, minimise number of different bin patterns (layouts) to reduce changeover times, minimise costs
- Each item j has a demand of d_j copies.
- It is possible to increase the demand d_j of each item j by a given factor $f_j \geq 1$.

Hochschule Niederrhein
University of Applied Sciences
iPattern
Institut für Mustererkennung
Institute for Pattern Recognition

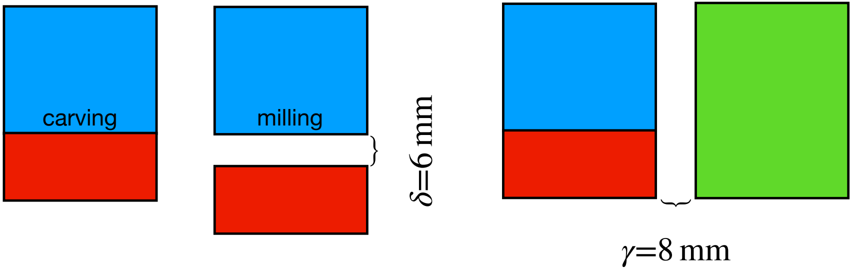
A variant of the 2D bin packing problem



We extend the classical problem:

- Minimise number of bins, maximise occupied area, minimise number of different bin patterns (layouts) to reduce changeover times, minimise costs
- Each item j has a demand of d_j copies.
- It is possible to increase the demand d_j of each item j by a given factor $f_j \geq 1$.
- Additionally, some optional items may be used to increase the occupied area.
- However, optional items are associated with costs $cost_j \geq 0$.

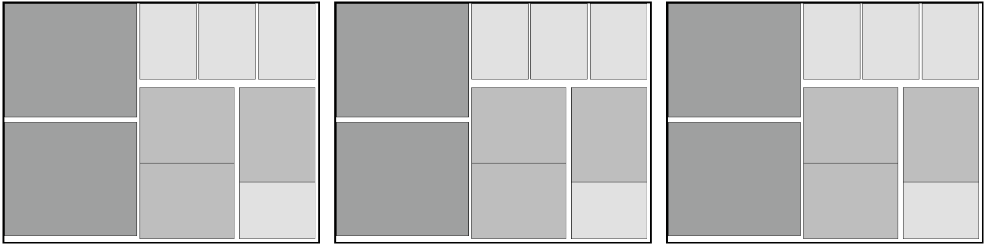
A variant of the 2D bin packing problem (2)



- Milling and carving can be combined. The distance between an item to be milled and all its neighbours must be at least $\delta = 6\,\mathrm{mm}$.
- Guillotine cuts are not mandatory. However, if an edge to be carved or milled ends inside the board, a larger minimum distance of $\gamma = 8\,\mathrm{mm}$ is needed.

Outline

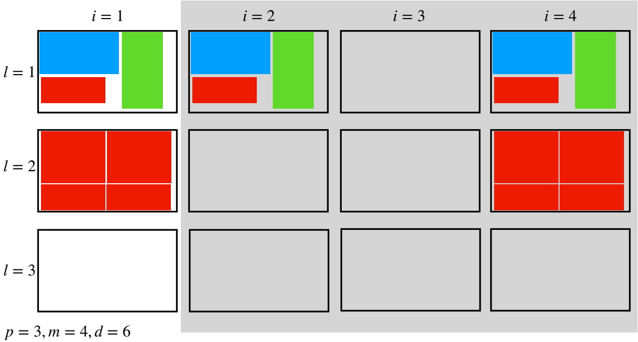
- The problem
- **Mixed Integer Linear Program (MILP) formulation**
- Results



Hochschule Niederrhein
University of Applied Sciences
iPattern
Institut für Mustererkennung
Institute for Pattern Recognition

Multiple bins with the same pattern

- We create a maximum of p potentially different patterns and then duplicate them using bins (l, i). Each pattern $l \in [p]$ can have at most m identical copies, $i \in [m]$. Placement conditions have to be checked only for bins $(l, 1)$.



- Each item j can have at most d copies within one single bin.
- Binary variable $bin_{l,i,j,c}$ is one iff copy $c \in [d]$ of item j exists in bin (l, i).

Key variables

- Number of copies of item j in bin (l, i):

$$cnt_{l,i,j} = \sum_{c \in [d]} bin_{l,i,j,c}.$$

Key variables

- Number of copies of item j in bin (l, i):

$$cnt_{l,i,j} = \sum_{c \in [d]} bin_{l,i,j,c}.$$

- $used_{l,i}$ is set to one iff bin (l, i) is non-empty, i.e.,

$$bin_{l,i,j,c} \leq used_{l,i} \text{ for all } j \in [n], c \in [d].$$

  One objective is to minimise the number of used bins: $\sum_{l=1}^{p} \sum_{i=1}^{m} used_{l,i}$.

Key variables

- Number of copies of item j in bin (l, i):

$$cnt_{l,i,j} = \sum_{c \in [d]} bin_{l,i,j,c}.$$

- $used_{l,i}$ is set to one iff bin (l, i) is non-empty, i.e.,

$$bin_{l,i,j,c} \leq used_{l,i} \text{ for all } j \in [n], c \in [d].$$

  One objective is to minimise the number of used bins: $\sum_{l=1}^{p} \sum_{i=1}^{m} used_{l,i}$.

- If a bin (l, i) is used ($used_{l,i} = 1$) then it must have the same item count than bin $(l, 1)$: For all $l \in [p]$, $2 \leq i \leq m$, $j \in [n]$:

$$-d \cdot (1 - used_{l,i}) <= cnt_{l,1,j} - cnt_{l,i,j} \leq d \cdot (1 - used_{l,i}).$$

Hochschule Niederrhein
University of Applied Sciences
iPattern
Institut für Mustererkennung
Institute for Pattern Recognition

Key variables

- Number of copies of item j in bin (l, i):

$$cnt_{l,i,j} = \sum_{c \in [d]} bin_{l,i,j,c}.$$

- $used_{l,i}$ is set to one iff bin (l, i) is non-empty, i.e.,

$$bin_{l,i,j,c} \leq used_{l,i} \text{ for all } j \in [n], c \in [d].$$

  One objective is to minimise the number of used bins: $\sum_{l=1}^{p} \sum_{i=1}^{m} used_{l,i}$.

- If a bin (l, i) is used $(used_{l,i} = 1)$ then it must have the same item count than bin $(l, 1)$: For all $l \in [p]$, $2 \leq i \leq m$, $j \in [n]$:

$$-d \cdot (1 - used_{l,i}) <= cnt_{l,1,j} - cnt_{l,i,j} \leq d \cdot (1 - used_{l,i}).$$

- The binary variable $rot_{l,j,c}$ is one iff bin-specific copy c of item j is rotated by 90 degrees in bin $(l, 1)$.

Hochschule Niederrhein
University of Applied Sciences
iPattern
Institut für Mustererkennung
Institute for Pattern Recognition

Placement of items within the given range

- An item j may be marked as optional with $o_j = 1$.
- Only if selected, it must also be placed in the specified number of copies.
- To select optional items, we introduce binary variables sel_j with

$$1 - o_j \leq sel_j \text{ for all } j \in [n].$$

- Then for all $j \in [n]$:

$$sel_j \cdot d_j \leq \sum_{l \in [p]} \sum_{i \in [m]} cnt_{l,i,j} \leq sel_j \cdot f_j \cdot d_j.$$

Hochschule Niederrhein
University of Applied Sciences
iPattern
Institut für Mustererkennung
Institute for Pattern Recognition

Objective functions

The objective is to

- use a minimum number of bins,
- use a minimum number of different patterns,
- obtain a largest covered area (i.e. minimum waste),
- minimise the costs of optional items.

Instead of dealing with a Pareto front, we simply use linear scalarisation and minimise

$$\sum_{l=1}^{p} \sum_{i=1}^{m} c_0(i) \cdot used_{l,i} - c_3 \sum_{j \in [n]} \sum_{c \in [d]} \sum_{l \in [p]} \sum_{i \in [m]} bin_{l,i,j,c} \cdot w_j \cdot h_j + c_4 \sum_{j=1}^{n} sel_j \cdot cost_j,$$

where $c_0(1) := c_1$, and $c_0(l) := c_2$ for $l > 1$, weight pattern bins ($l = 1$) differently to bins with pattern copies ($l > 1$).

Hochschule Niederrhein
University of Applied Sciences
iPattern
Institut für Mustererkennung
Institute for Pattern Recognition

Alternative objective functions

- Additionally minimise the number of different and/or the sum of all x- and y-coordinates.
- Avoid maximising the area covered. Require a minimum coverage of $\kappa\%$:

$$\sum_{j\in[n]}\sum_{c\in[d]} bin_{l,i,j,c}\cdot w_j\cdot h_j \geq used_{l,i}\cdot\frac{\kappa}{100}\cdot W\cdot H \text{ for all } l\in[p],\ i\in[m].$$

Hochschule Niederrhein
University of Applied Sciences
iPattern
Institut für Mustererkennung
Institute for Pattern Recognition

Placement conditions

- The items in each bin must not overlap: We use constraints from

  Pisinger, D., Sigurd, M.: *The two-dimensional bin packing problem with variable bin sizes and costs.* Discrete Optimization 2(2), 154–167 (2005).

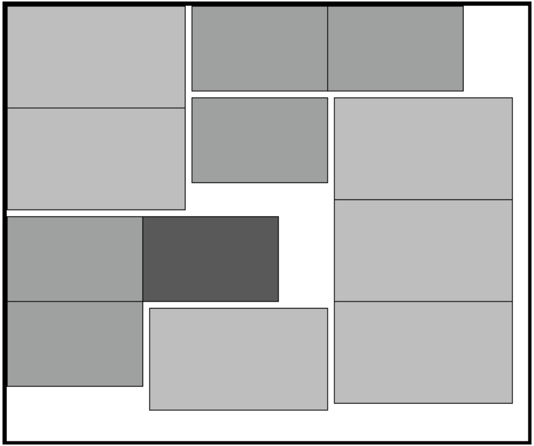
- We integrated rotation infos $rot_{l,j,c}$.
- We added the additional distance constraint required in case of milling: Each item j has to be either milled, then $milled_j = 1$, or carved ($milled_j = 0$). Let

  $$dist_{j,j'} := milled_j \vee milled_{j'} \in \{0, 1\}.$$

  Item j and j' must have a distance of $dist_{j,j'} \cdot \delta$.

Hochschule Niederrhein
University of Applied Sciences
iPattern
Institut für Mustererkennung
Institute for Pattern Recognition

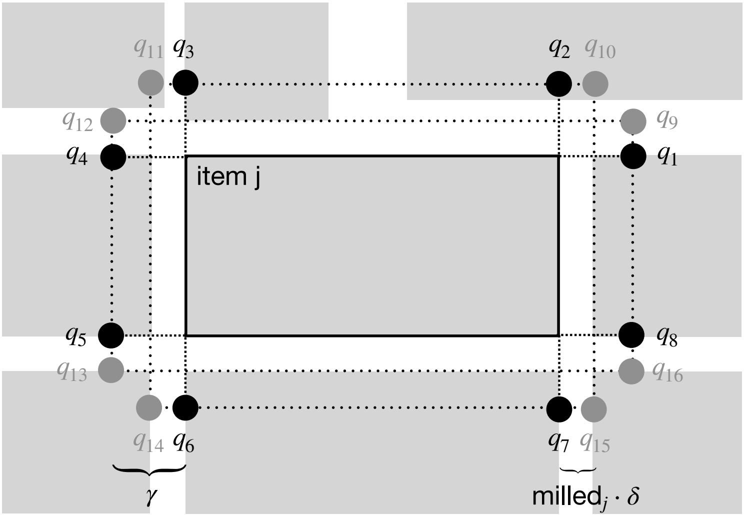
Distance constraint for non-guillotine cuts



Example: Carving with the non-guillotine cut constraint

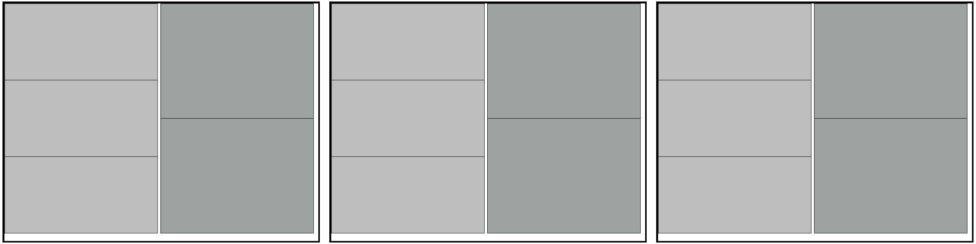
Hochschule Niederrhein
University of Applied Sciences
iPattern
Institut für Mustererkennung
Institute for Pattern Recognition

Distance constraint for non-guillotine cuts (2)



Outline

- The problem
- Mixed Integer Linear Program (MILP) formulation
- **Results**



Hochschule Niederrhein
University of Applied Sciences
iPattern
Institut für Mustererkennung
Institute for Pattern Recognition

Setup of experiments

- IBM CPLEX 22.1.1 optimiser[1] with a time limit of 600 s
- Test system: Ubuntu 22.04.4 LTS, AMD Ryzen 5 5500U CPU, 32 GB RAM.
- Importance of the four individual objectives was ranked in descending order: $c_1 = 1 + \frac{1}{p}$, $c_2 = 1$, $c_3 = \frac{1}{p^2 WHm}$, $c_4 = \frac{c_3}{10n}$.
- We chose $milled_j = cost_j = o_j = 0$, $j \in [n]$

[1] CPLEX version 12.8 found different solutions, sometimes faster, sometimes slower.

Experiments

We investigated the influence of maximising the area covered vs. coverage of $\kappa\%$ and of the non-guillotine cut constraint (values in brackets: no non-guillotine cut constraint)[2].

- Measurements labelled "single": one pattern ($p=1$) in one bin ($m=1$, $d=10$)
- "double": maximum of two patterns ($p=2$) with two bins each ($m=2$, $d=20$), prescribed multiplicities were doubled.

instance, $W \times H$	r_1, 820×620		r_2, 830×630			r_3, 840×640				r_4, 850×650				
item number $j =$	1	2	1	2	3	1	2	3	4	1	2	3	4	5
width $w_j/100$	4	3	3.5	2.5	2	2	2	2	2	2	2	1	0.8	1
height $h_j/100$	4	2	3	2	1.5	3	2	1.5	1	3	2	2	2	1
min. multiplicity d_j	2	3	2	3	4	2	3	4	6	2	3	4	5	8
max. mult. $f_j \cdot d_j$	3	4	3	4	5	3	4	5	7	3	4	5	6	9

instance variant	single	double	single	double	single	double	single	double
max. area	42 s (2 s)	600 s (27 s)	600 s (600 s)	– (**600 s**)	– (600 s)	– (**600 s**)	– (–)	– (**600 s**)
$\kappa\% = 90\%$	1 s (1 s)	600 s (7 s)	100 s (1 s)	– (–)	– (–)	– (–)	– (–)	– (–)
$\kappa\% = 80\%$	1 s (1 s)	600 s (11 s)	121 s (1 s)	– (247 s)	553 s (2 s)	– (–)	565 s (5 s)	– (–)
$\kappa\% = 70\%$	1 s (1 s)	600 s (29 s)	180 s (1 s)	– (**600 s**)	405 s (2 s)	– (**600 s**)	– (3 s)	– (**600 s**)
$\kappa\% = 60\%$	1 s (1 s)	600 s (11 s)	124 s (1 s)	– (33 s)	– (2 s)	– (**600 s**)	– (21 s)	– (**600 s**)

[2] The time to find an optimal solution (only feasible solution if 600 s) is shown.
Bold numbers indicate sub-optimal solutions with too many occupied bins.

Tests with one pattern and many bins

We chose $p = 1$, $m = 10$, $d = 10$ and increased item multiplicities to $d_j = 10 \cdot (j + 1)$, $j \in [4]$, $d_5 = 80$, $f_j \cdot d_j = d_j + 1$, $j \in [5]$ (if an item j belonged to the instance).

instance	r_1	r_2	r_3	r_4
max. area	600 s (10 s)	535 s (8 s)	– (5 s)	– (8 s)
$\kappa\% = 90\%$	215 s (6 s)	– (146 s)	– (2 s, no solution)	– (2 s, no solution)
$\kappa\% = 80\%$	498 s (5 s)	140 s (48 s)	463 s (3 s)	– (23 s)
$\kappa\% = 70\%$	317 s (2 s)	– (9 s)	– (3 s)	– (24 s)
$\kappa\% = 60\%$	600 s (11 s)	600 s (35 s)	600 s (2 s)	– (5 s)

Hochschule Niederrhein
University of Applied Sciences
iPattern
Institut für Mustererkennung
Institute for Pattern Recognition

Conclusions

The runtimes are practicable if problems are limited to one pattern and the non-guillotine cut constraint is omitted.

- To deal with the non-guillotine cut constraint:
  - One can generally specify a uniform distance of 8 mm (or less) between items and remove the constraint for a start. With a solution obtained, the problem with the original distant constraint can be warm-started.
- To deal with multiple patterns:
  - Separate the assignment of items to patterns from the calculation of the pattern layouts, i. e. use two consecutive optimisation problems.

Possible extensions: Additional material balancing conditions may be added due to the requirements of the manufacturing process.

Hochschule Niederrhein
University of Applied Sciences
iPattern
Institut für Mustererkennung
Institute for Pattern Recognition