Exact and Heuristic MIP Models for Nesting Problems

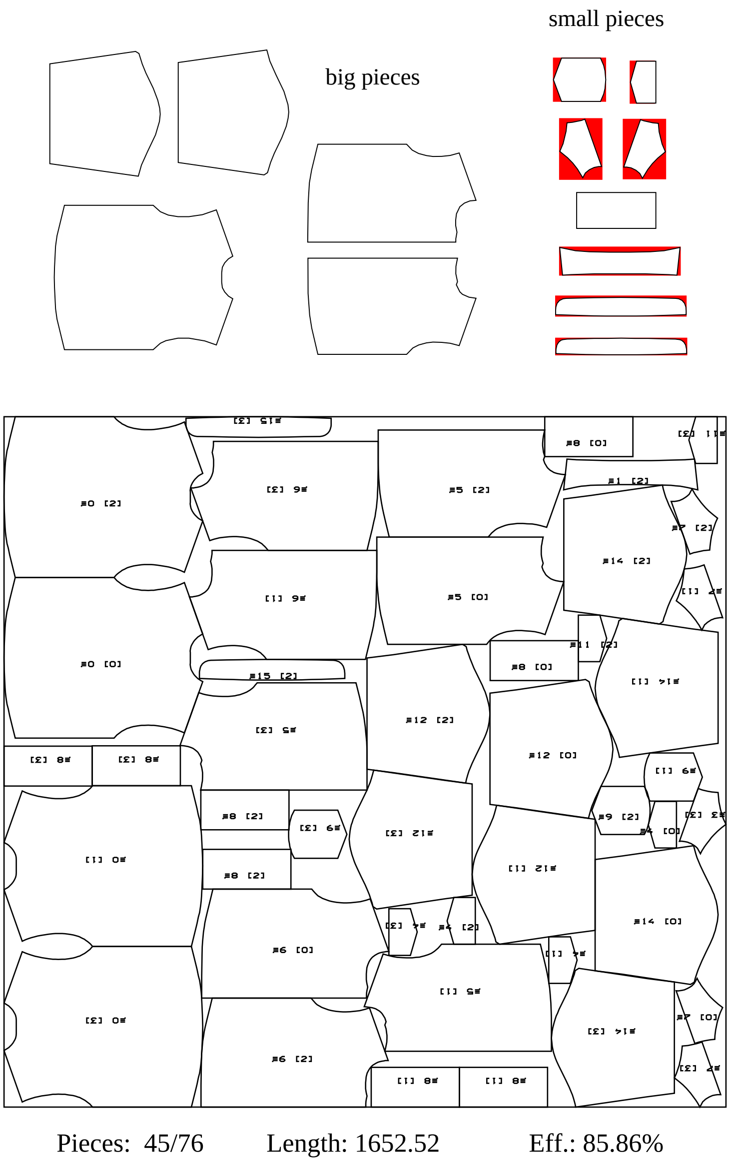
Matteo Fischetti, Ivan Luzzi

DEI, University of Padova

presented at the EURO meeting, Istanbul, July 2003

The Nesting Problem

*Given a set of 2-dimensional **pieces** of generic (irregular) form and a 2-dimensional **container**, find the best non-overlapping position of the pieces within the container.*



Complexity: NP-hard (and **very hard** in practice)

Literature

Heuristics

- J. Blazewicz, P. Hawryluk, R. Walkowiak, *Using a tabu search approach for solving the two-dimensional irregular cutting problem*, AOR 1993
- J.F.C. Oliveira, J.A.S. Ferreira, *Algorithms for nesting problems*, Springer-Verlag 1993
- K.A. Dowsland, W.B. Dowsland, J.A. Bennel, *Jostling for position: local improvement for irregular cutting patterns*, JORS 1998
- ...

Containment & Compaction

- K. Daniels, Z. Li, V. Milenkovic, *Multiple Containment Methods*, Technical Report TR-12-94, Harvard University, July 1994.
- Z. Li, V. Milenkovic, *Compaction and separation algorithms for non-convex polygons and their applications*, EJOR 1995
- K. Daniels, *Containment algorithms for non-convex polygons with applications to layout*, PhD thesis 1995
- ...

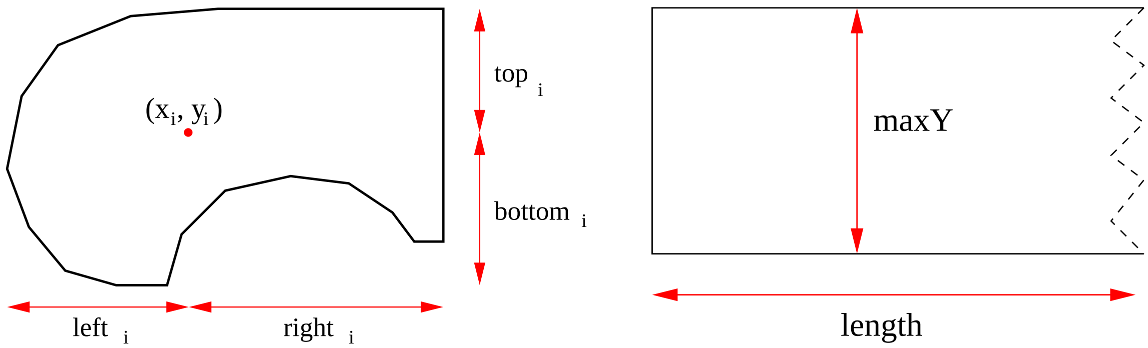
Branch & Bound

- R. Heckmann, T. Lengauer, *Computing closely matching upper and lower bounds on textile nesting problems*, EJOR 1998
- ...

A MIP model for the nesting problem

Input

- We are given a set $\mathcal{P}$ of $n := |\mathcal{P}|$ **pieces**. The form of each piece is defined by a simple polygon described through the list of its vertices. In addition, each piece i is associated with an arbitrary reference point whose 2-dimensional coordinates $\boldsymbol{v_i} = (x_i, y_i)$ will be used to define the placement of the piece within the container.
- The **container** is assumed to be of rectangular form, with fixed height $maxY$ and infinity length.



Variables

- $\boldsymbol{v_i} = (x_i, y_i)$: coordinates of the reference point of piece i
- $length$: right margin of the used area within the container ("makespan")

Objective

Minimize $length$, i.e., maximize the percentage *efficiency* computed as:

$$\textbf{efficiency} = \frac{\sum_{i=1}^{n} area_i}{length * maxY} * 100$$

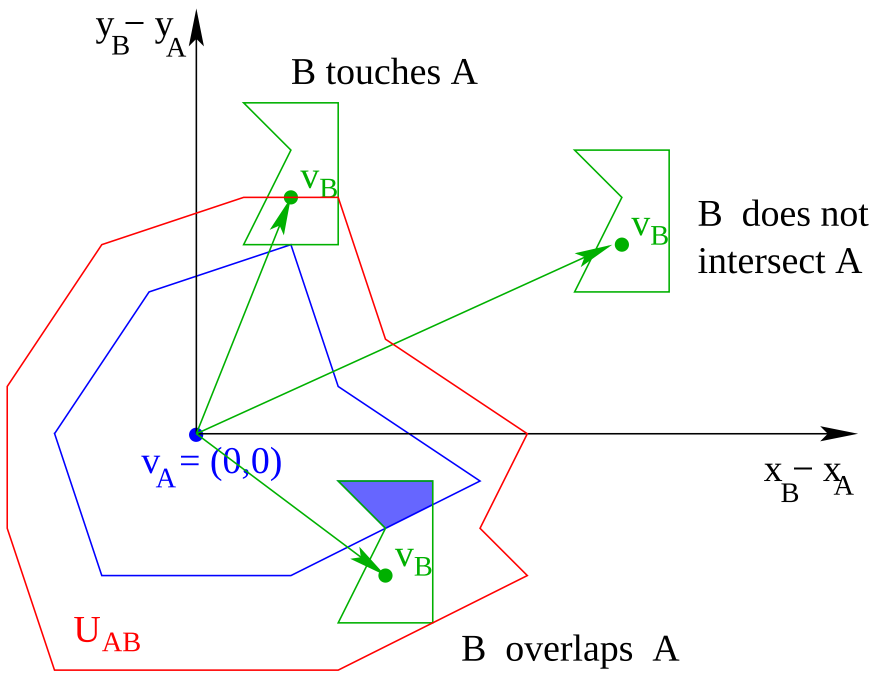
How to check/model the overlap between two pieces?

The **Minkowski sum** of two polygons A and B is defined as:

$$A \oplus B \ = \ \{a + b : \ a \in A, \ b \in B\}$$

The **no-fit polygon** between two polygons A and B is defined as

$$U_{AB} \ := \ A \oplus (-B)$$



Interpretation: place the reference point of polygon A at the origin; then the *no-fit polygon* represents the trajectory of the reference point of polygon B when it is moved around A so as to be in touch (with no overlap) with it.

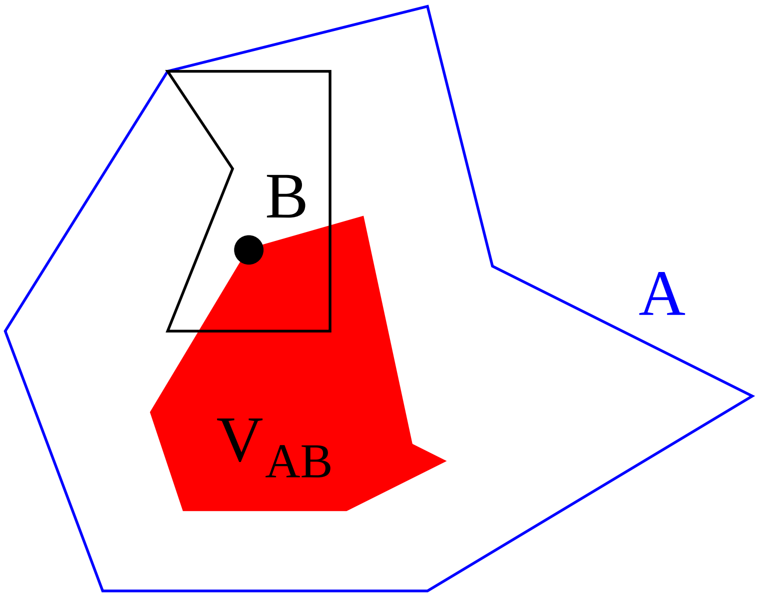
The **Minkowski difference** between polygons A and B is defines as:

$$A \ominus B \; = \; \bigcap_{b \in B} A^b$$

The **containment polygon** corresponding to two polygons A and B is defined as:

$$V_{AB} \; := \; A \ominus (-B)$$

and represents the region of containment (without overlap) of a piece B inside a hole A.

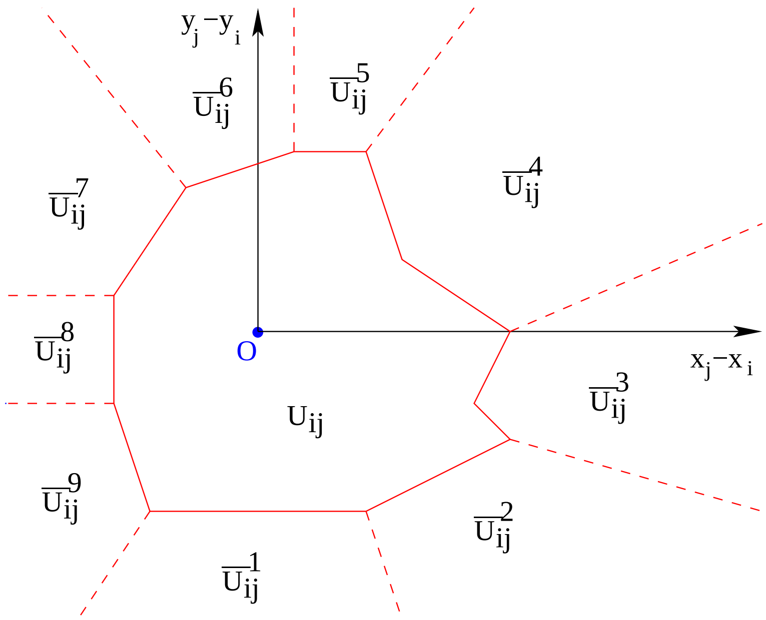


Using the no-fit polygon

How to express the non-overlapping condition between two pieces i and j?

$$\boldsymbol{v_j} - \boldsymbol{v_i} = \begin{pmatrix} x_j \\ y_j \end{pmatrix} - \begin{pmatrix} x_i \\ y_i \end{pmatrix} \notin U_{ij} \iff \boldsymbol{v_j} - \boldsymbol{v_i} \in \overline{U}_{ij}, \quad \forall\, i, j \in \mathcal{P} : \; i < j$$

Partition the non-convex region $\overline{U}_{ij}$ into a collection of m_{ij} disjoint polyhedra $\overline{U}^k_{ij}$ called *slices*.



Each slice can be represented through a set of linear constraints of the form:

$$\overline{U}^{\,k}_{ij} = \{\boldsymbol{u} \in \mathbb{R}^2 : \; \boldsymbol{A^k_{ij}} \cdot \boldsymbol{u} \le \boldsymbol{b^k_{ij}}\}$$

The MIP model

A variant of a model by Daniels, Li, and Milenkovic (1994)

Variables

- $\boldsymbol{v_i} = (x_i, y_i)$: coordinates of the reference point of piece i
- $length$: rightmost used margin of the container
- $z_{ij}^k = \begin{cases} 1 & \text{if } \boldsymbol{v_j} - \boldsymbol{v_i} \in \overline{U}_{ij}^k \\ 0 & \text{otherwise} \end{cases} \qquad \forall\, i, j \in \mathcal{P} : i < j,\ k = 1 \ldots m_{ij}$

Model

$$
\begin{array}{lll}
\min & length + \varepsilon \sum_{i \in \mathcal{P}} (x_i + y_i) & \\
\text{s. t.} & left_i \le x_i \le length - right_i & \forall \\
& bottom_i \le y_i \le maxY - top_i & \forall\, i \in \mathcal{P} \\
& \boldsymbol{A_{ij}^k}(\boldsymbol{v_j} - \boldsymbol{v_i}) \le \boldsymbol{b_{ij}^k} + M(1 - z_{ij}^k) \cdot \mathbf{1} & \\
& & \forall\, i, j \in \mathcal{P} :\ i < j,\ k = 1 \ldots m_{ij} \\
& \sum_{k=1}^{m_{ij}} z_{ij}^k = 1 & \forall\, i, j \in \mathcal{P} :\ i < j \\
& z_{ij}^k \in \{0, 1\} & \forall\, i, j \in \mathcal{P} :\ i < j,\ k = 1 \ldots m_{ij}
\end{array}
$$

Constraint coefficient lifting

Issue: *the use of big-M coefficients makes the LP relaxation of the model quite poor*

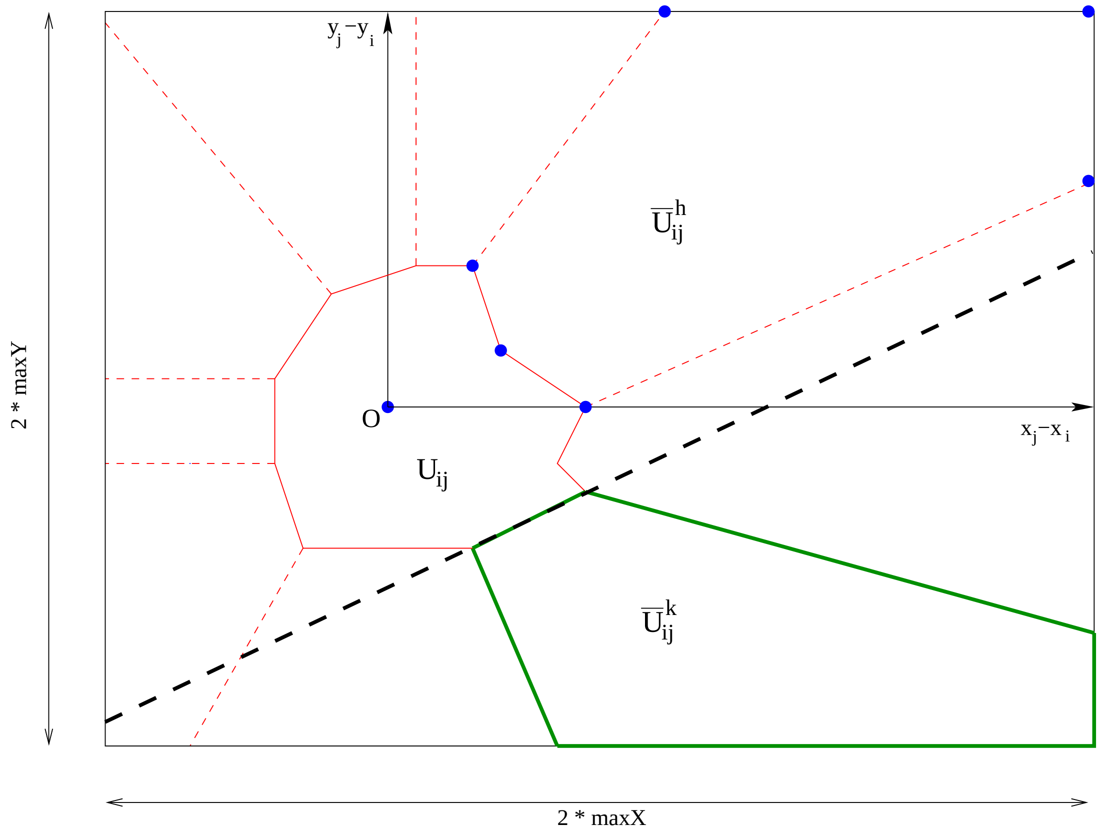
$$\alpha_{ij}^{kf}(x_j - x_i) + \beta_{ij}^{kf}(y_j - y_i) \leq \gamma_{ij}^{kf} + M(1 - z_{ij}^{k}) \qquad \forall\, f = 1 \ldots t_{ij}^{k}$$

Replace the big-M coefficient by:

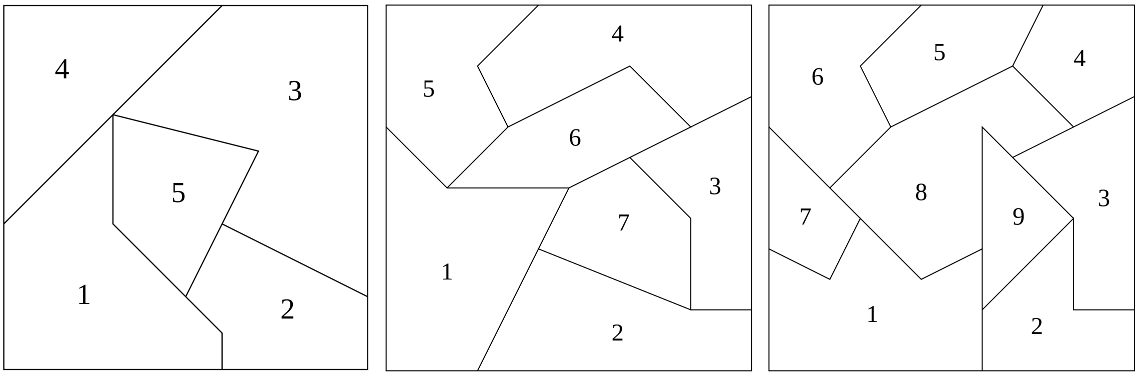
$$\delta_{ij}^{kfh} := \max_{(v_j - v_i) \in \overline{U}_{ij}^{h} \cap B} \alpha_{ij}^{kf}(x_j - x_i) + \beta_{ij}^{kf}(y_j - y_i)$$

so as to obtain (easily computable) lifted constraints of the form:

$$\alpha_{ij}^{kf}(x_j - x_i) + \beta_{ij}^{kf}(y_j - y_i) \leq \sum_{h=1}^{m_{ij}} \delta_{ij}^{kfh}\, z_{ij}^{h}$$



Some computational results



INSTANCE	PIECES	INT	PRIOR	NODES	TIME	GAP
Glass1	5	73	no	470	0.26”	0%
			yes	111	0.11”	0%
Glass2	7	173	no	100,000	97.40”	32.08%
			yes	11,414	13.29”	0%
Glass3	9	302	no	100,000	157.76”	59.82%
			yes	100,000	203.48”	58.70%

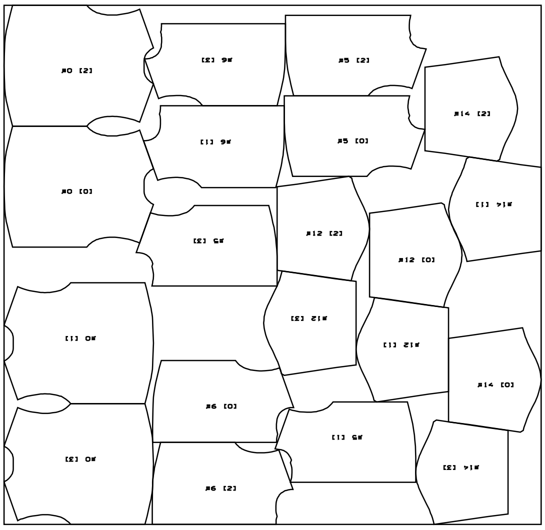
PRIOR yes/no refers to the use of a specific branching strategy based on "clique priorities"

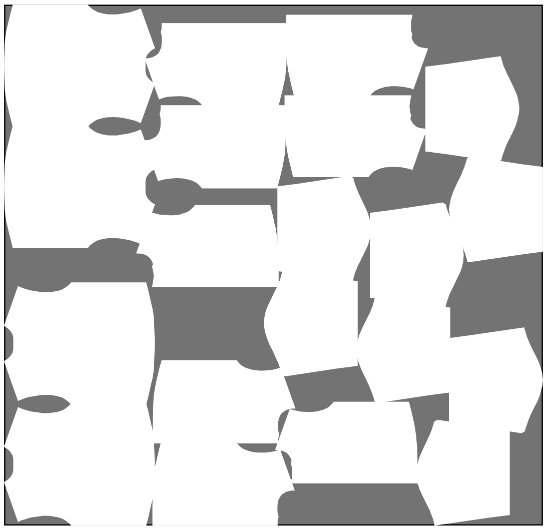
Solved with ILOG-CPLEX 7.0 on a PC AMD Athlon/1.2 GHz

"Not usable in practice for real-world problems"

Multiple Containment Problem

An important subproblem: after having placed the "big pieces", find the best placement of the remaining "small pieces" by using the holes left by the big ones.





A *greedy* approach for placing the small pieces can produce poor results

Aim: Define an approximate MIP for guiding the placement of the small pieces

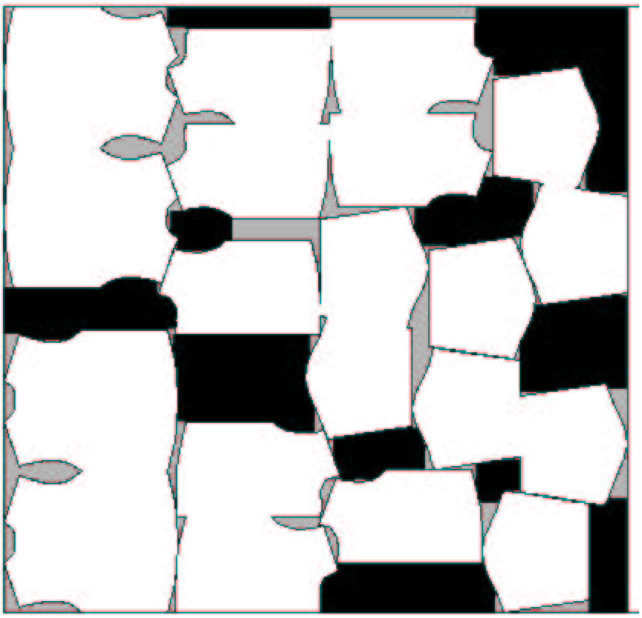
Idea: Small pieces can be approximated well by rectangles

Input

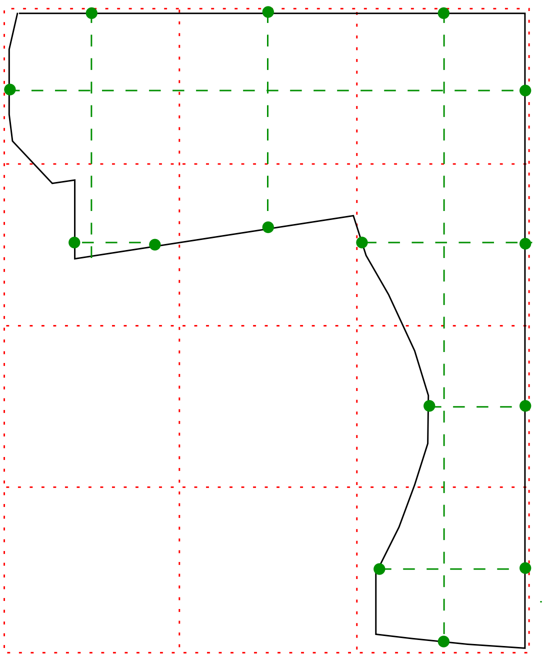
- Set $\mathcal{P}$ of n small pieces
- Set $\mathcal{H}$ of m irregular polygons representing the available *holes*

Geometrical considerations

- rectangular approximation of the small pieces
- original holes and *usable* holes



- placement grid within each hole.



An approximate multiple-containment MIP model

$$\min \quad \sum_{h\in\mathcal{H}} \Big(holeArea_h \cdot U^h - \sum_{r=1}^{R_h} \sum_{c=1}^{C_h} \sum_{p\in\mathcal{P}} pieceArea_p \cdot Z_{rc}^{hp} \Big) + \varepsilon \sum_{h\in\mathcal{H}} \sum_{r=1}^{R_h} \sum_{c=1}^{C_h} (X_{rc}^h + Y_{rc}^h)$$

s. t.

$$\sum_{p\in\mathcal{P}} Z_{rc}^{hp} \le U^h \qquad \forall\, h\in\mathcal{H},\ r=1\ldots R_h,\ c=1\ldots C_h$$

$$\sum_{p\in\mathcal{P}} pieceArea_p \sum_{r=1}^{R_h} \sum_{c=1}^{C_h} Z_{rc}^{hp} \le holeArea_h \qquad \forall\, h\in\mathcal{H}$$

$$X_{rc}^h + \sum_{p\in\mathcal{P}} length_p\, Z_{rc}^{hp} \le X_{r,\,c+1} \qquad \forall\, h\in\mathcal{H},\ r=1\ldots R_h,\ c=1\ldots C_h-1$$

$$X_{rc}^h + \sum_{p\in\mathcal{P}} length_p\, Z_{rc}^{hp} \le rowEnd_r^h \qquad \forall\, h\in\mathcal{H},\ r=1\ldots R_h,\ c=C_h$$

$$\sum_{p\in\mathcal{P}} length_p \sum_{c=1}^{C_h} Z_{rc}^{hp} \le rowLength_r^h \qquad \forall\, h\in\mathcal{H},\ r=1\ldots R_h$$

$$Y_{rc}^h + \sum_{p\in\mathcal{P}} width_p\, Z_{rc}^{hp} \le Y_{r+1,\,c} \qquad \forall\, h\in\mathcal{H},\ r=1\ldots R_h-1,\ c=1\ldots C_h$$

$$Y_{rc}^h + \sum_{p\in\mathcal{P}} width_p\, Z_{rc}^{hp} \le colEnd_c^h \qquad \forall\, h\in\mathcal{H},\ r=R_h,\ c=1\ldots C_h$$

$$\sum_{p\in\mathcal{P}} width_p \sum_{r=1}^{R_h} Z_{rc}^{hp} \le colWidth_c^h \qquad \forall\, h\in\mathcal{H},\ c=1\ldots C_h$$

Bounds on the variables

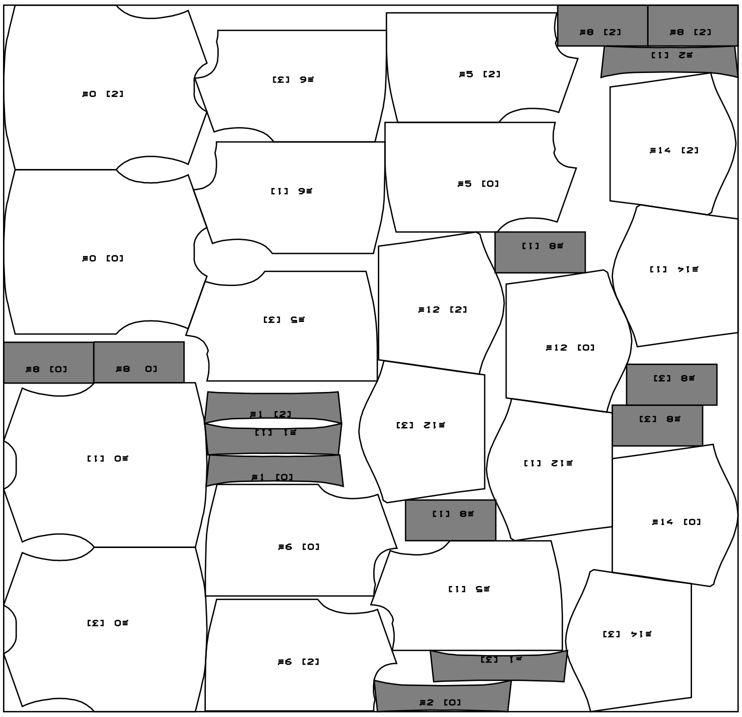
$$\max(rowStart_r^h,\ origX_h + (c-1)\cdot cellLength_h) \ \leq\ X_{rc}^h$$
$$\leq\ \min(rowEnd_r^h,\ origX_h + (c)\cdot cellLength_h)$$
$$\forall\, h \in \mathcal{H},\ \ r = 1 \ldots R_h,\ c = 1 \ldots C_h$$
$$\max(colStart_c^h,\ origY_h + (r-1)\cdot cellWidth_h) \ \leq\ Y_{rc}^h$$
$$\leq\ \min(colEnd_c^h,\ origY_h + r\cdot cellWidth_h)$$
$$\forall\, h \in \mathcal{H},\ \ r = 1 \ldots R_h,\ c = 1 \ldots C_h$$
$$U^h \in \{0,1\} \qquad \forall\, h \in \mathcal{H}$$
$$Z_{rc}^{hp} \in \{0,1\} \qquad \forall\, h \in \mathcal{H},\ p \in \mathcal{P},\ r = 1 \ldots R_h,\ c = 1 \ldots C_h$$

Remark 1: Solvable in short computing time

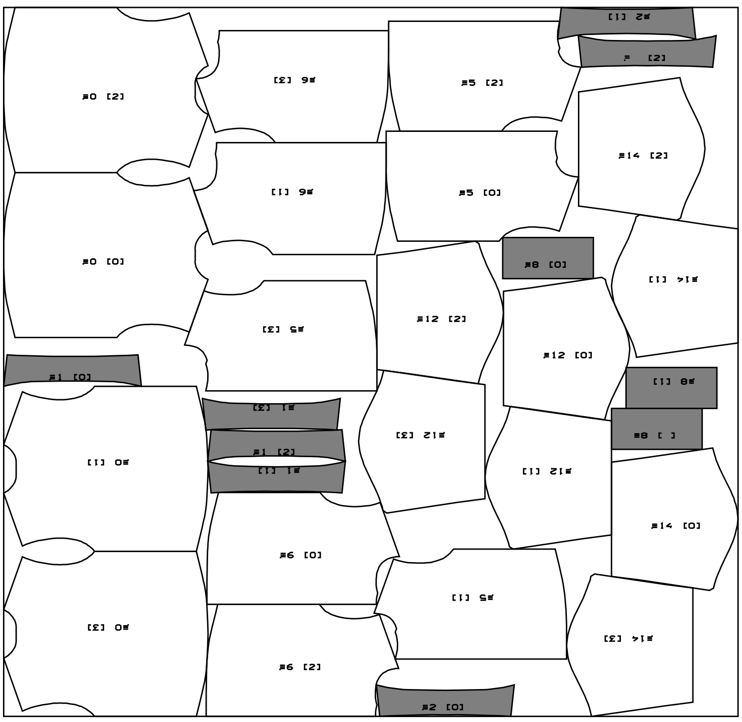
Remark 2: To be followed by a greedy post-processing procedure for fixing possible overlaps

Remark 3: Sequential approach: big pieces placed first, "special pieces" of intermediate size/difficulty second, and "trims" last.

Example: Smart vs. greedy placement of "special pieces"

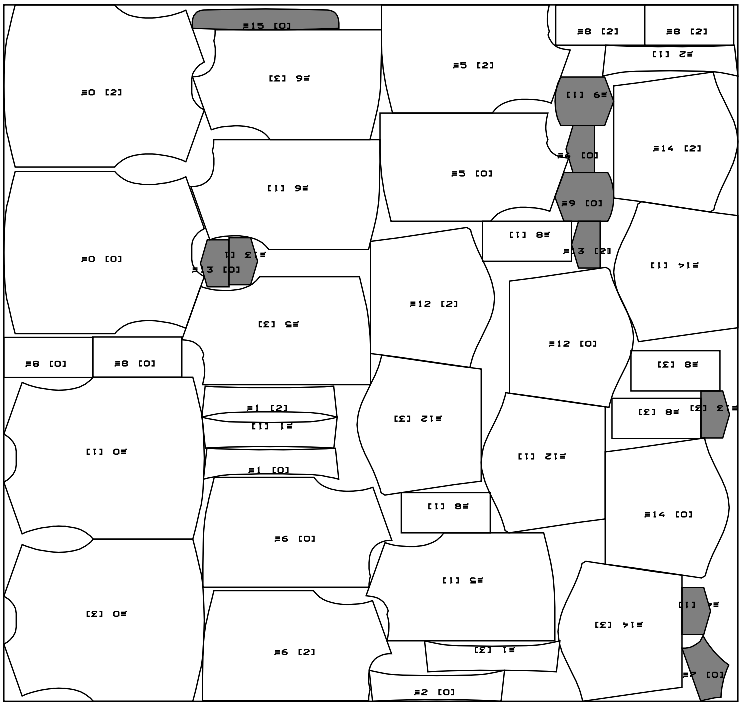


Pieces: 34/76 Length: 1643.53 Eff.: 83.97%

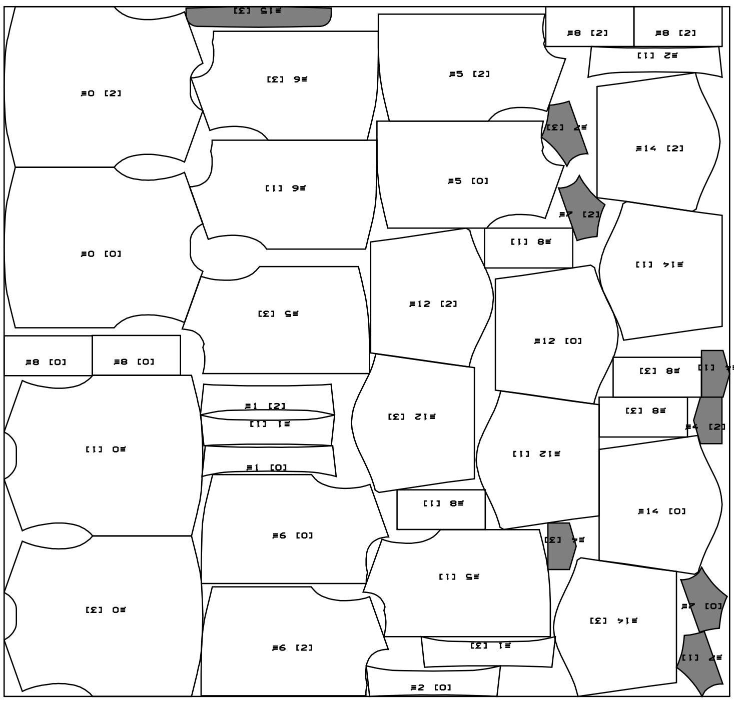


Pieces: 30/76 Length: 1634.55 Eff.: 81.54%

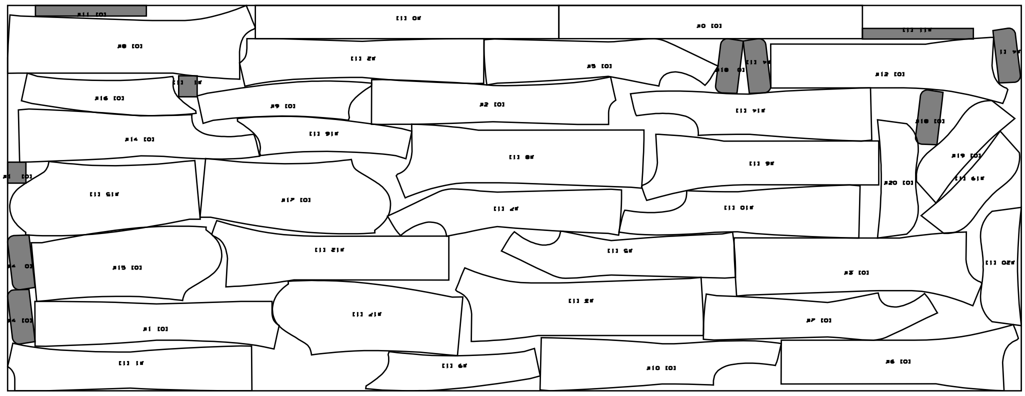
Example (cont'd): Smart vs. greedy placement of "trims"



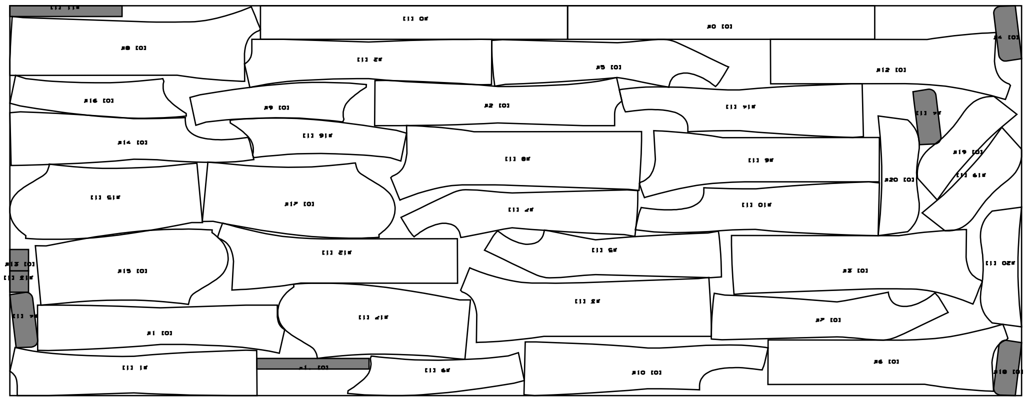
Pieces: 44/76 Length: 1665.50 Eff.: 86.13%



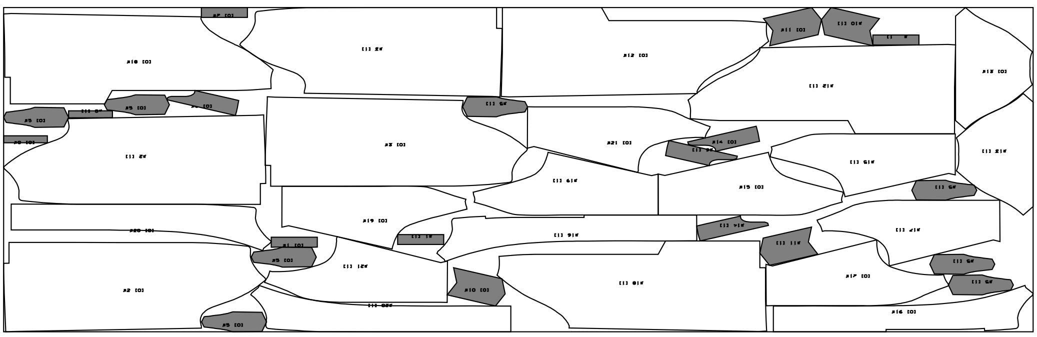
Pieces: 42/76 Length: 1660.87 Eff.: 85.67%



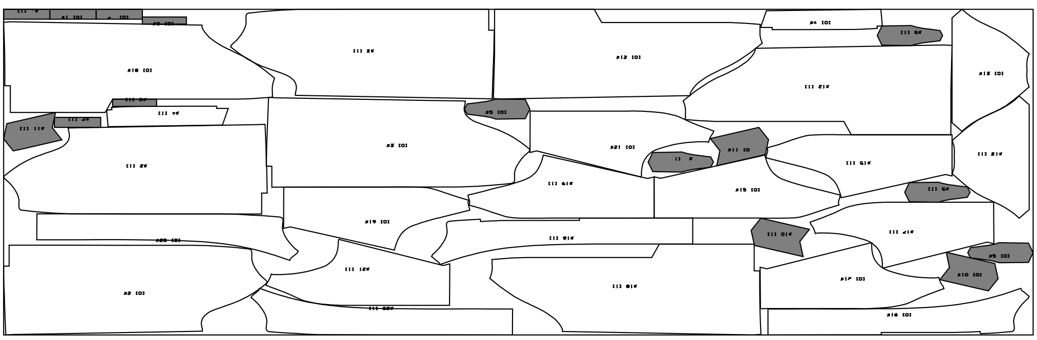
Pieces: 44/50 Length: 3840.28 Efficiency: 82.12 %



Pieces: 42/50 Length: 3838.27 Efficiency: 81.57 %



Pieces: 44/54 Length: 4697.05 Efficiency: 83.74 %



Pieces: 39/54 Length: 4671.81 Efficiency: 83.58 %

Preliminary Computational Results

INSTANCE	PIECES	TRIMS	LENGTH	EFFIC.
82 - group 1				
smart	34/76	14	1643.53	83.97%
greedy	30/76	10	1634.55	81.54%
82 - group 2				
smart	44/76	10	1665.50	86.13%
greedy	42/76	8	1660.87	85.67%
101				
smart	44/50	10	3840.28	82.12%
greedy	42/50	8	3838.27	81.57%
385				
smart	44/54	22	4697.05	83.74%
greedy	39/54	17	4671.81	83.58%