# Challenge ROADEF / EURO 2018
# Cutting Optimization
# Problem Description

Lydia Tlilane and Quentin Viaud
Saint-Gobain, Datalab

May 18, 2018

## Contents

# 1 Introduction

The Saint-Gobain Group designs, manufactures and distributes materials and solutions which are key ingredients in the well-being of each of us and the future of all. They can be found everywhere in our living places and our daily life: buildings, transportation, infrastructure and in many industrial applications. They provide comfort, performance and safety while addressing the challenges of sustainable construction, resource efficiency and climate change.

Saint-Gobain Glass France (SGGF), part of the Saint-Gobain Group, is one of the world's leading flat glass manufacturers. It specialises in float glass manufacture and magnetron coated glass producing an array of glass types with different functions: transparency, thermal and acoustic insulation, safety, solar control, decoration, self-cleaning function, etc. The products are intended for a wide variety of domestic and commercial applications including housing equipment (windows, bay windows, interior design), facade, urban development and the realization of major projects.

Flat glass is mostly produced through a process called the " float process ". In this process, various powders (sand, soda, ...) are melted together inside a large furnace in order to create a liquid glass ribbon which is spread over a tin bath and then cooled down to solidify. The obtained infinite ribbon is then cut into large glass sheets (typically 3m x 6m) called jumbo's. Afterwards, these jumbo's are stacked on stillages to be sent to so-called transformers. In general, these jumbo's are not used such as but are most of the time recut into smaller rectangular pieces adapted to the needs of the customers. These smaller pieces of glass are cut according to a cutting pattern which satisfies a certain amount of constraints link to the customer (order, ...) or to the physics of glass (guillotine which means that a piece of glass can only be cut from one edge to the other through the propagation of a crack along a straight line). A cutting plan can be seen as a paving of the jumbo by rectangular pieces of various sizes positioned in such a way that the geometrical glass losses (remaining glass surface too small to cut a new piece) are as low as possible. Actually, the jumbo's are not perfect in terms of quality and may also contain defects inherent to the float process. At the exit of this process, a scanner allows, for each jumbo, to establish the map of defects (position, criticity, ...). These defects can be considered as punctual in our problem. This information is stored in a database which contains the map of defects of each jumbo sold to a transformer. During the cutting process, when a defect is positioned in a cut glass piece, it is, most of the time, rejected as a quality loss. In this case, it is necessary to recut a glass piece of the same size from the jumbo. This reduces significantly the productivity of the line. To avoid this loss, one option is to adapt the cutting plan to the defect map measured at the exit of the float process and position the defects inside the "natural" geometrical glass losses of the cutting plan instead. The goal of this challenge is, for a given sequence of jumbo's and their attached defect maps on one side, and a given batch of glass pieces to cut, to propose an algorithm allowing to reduce as much as possible the glass losses of the cutting process. This means to minimize the number and the dimensions of the cut size containing a defect and to minimize the size of the geometrical glass losses.

# 2 Problem description

This section describes the input data, problem objective, constraints used in this challenge.

## 2.1 Inputs, definitions and notations

**Item:** an item is a glass piece to cut. An item $i$ is characterized by a pair $(w_i, h_i)$ representing its width and height.

**Stack:** a stack $s = (i_1, i_2, \ldots, i_j)$ is an ordered sequence of items such that $i_1 <_{cut} i_2 <_{cut} \ldots <_{cut} i_j$, with $<_{cut}$ the partial order operator. For two items $i_1$ and $i_2$, $i_1 <_{cut} i_2$ means that item $i_1$ has to be cut before item $i_2$. This order comes from some scheduling constraints related to the deliveries and item processing.

**Batch:** a batch $\mathcal{I}$ is the set of items to cut. It corresponds to a customer order. Using the stack notation, the item set $\mathcal{I}$ can be partitioned into $n$ stacks, $\mathcal{I} = \bigcup_{k=1}^{n} s_k$.

**Bin:** a bin is a jumbo obtained at the end of the float process. A bin $b$ is characterized by its width $W_b$, its height $H_b$ and its defect set $\mathcal{D}_b$. Jumbo's are stacked in the factory thus the bin set $\mathcal{B}$ is considered as ordered. Assume that bins are indexed from $\{0, \ldots, |\mathcal{B}|\}$, for two bins $b_1$ and $b_2$, $b_1 <_{cut} b_2$ means that bin $b_1$ is used to cut some items before starting using bin $b_2$. This implies that $b_1$ is removed from the bin stack before starting using bin $b_2$. Bins are assumed in a quantity large enough to cut all items and have the same size. The bin size is standardised. The difference between them is related to their defect set.

**Defect:** a defect $d$ is a tuple $(x_d, y_d, w_d, h_d)$ with $x_d$ is its coordinates on the x-axis, $y_d$ is its coordinates on the y-axis. $w_d$ (resp. $h_d$) is its width (resp. height).

**Guillotine cut:** a guillotine cut on a plate is a cut from one edge of the plate to the opposite edge, parallel to the remaining edge. In other words, the cut is guillotine if when applied to a rectangular plate it produces two new rectangular plates. This type of cut is mandatory to cut glass, it produces cracks otherwise. Figure 1 depicts guillotine and non guillotine patterns.

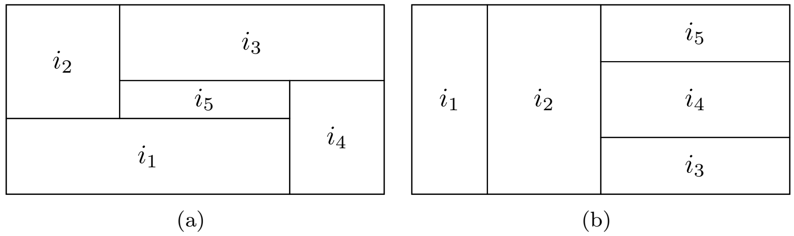



Figure 1: Non guillotine (a) and guillotine (b) patterns. There is no item order.

**Cutting pattern:** a cutting pattern for a given bin is a two-dimensional plan giving indication on how to cut items and the defect location for that bin. In the problem discussed here, a cutting pattern is always composed of guillotine cuts. Figure 2 depicts a guillotine pattern for a bin with one defect and the way to perform cuts.

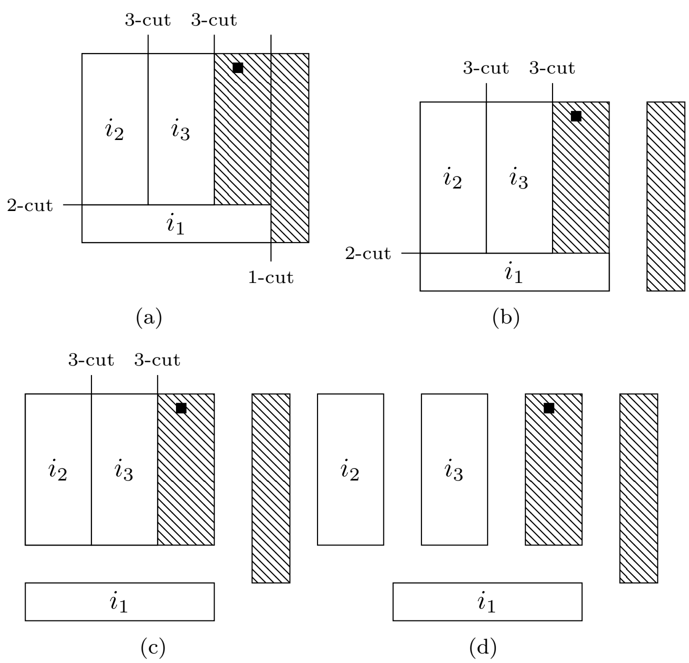



Figure 2: An initial pattern (a) and its variants by performing cuts (b-c-d). There is no item order.

**Waste:** a waste is a rectangular part of a bin which is not an item according to a cutting pattern. A waste occurs during the cutting process and corresponds to a raw material loss. Dashed lines in Figure 2 are losses.

**$\alpha$-cut:** a guillotine cut of depth $\alpha$. Due to technical limitation, the number of guillotine cuts which can be performed to cut an item is often limited by a certain number. For instance, a cutting pattern is said to be 2-stage (resp. 3-stage) when an item has to be cut in at most 2 (resp. 3) guillotine cuts. The notation $\alpha$-cut is used to note the number of cuts performed so far. The cutting pattern in Figure 2 is 3-stage, it requires two cuts to extract item $i_1$, three cuts to extract items $i_2$ and $i_3$.

**Tree representation of a cutting pattern:** a cutting pattern can be represented by a tree. This is possible since a guillotine cut always divides a rectangular plate in two smaller rectangular plates. Its root corresponds to the initial plate, its leafs are either the items or the wastes or the residual. The children of a given node except the leafs are the one obtained after performing a cut at a depth level $\alpha$ from the root node. The tree representation of pattern in Figure 2 is given in Figure 3.

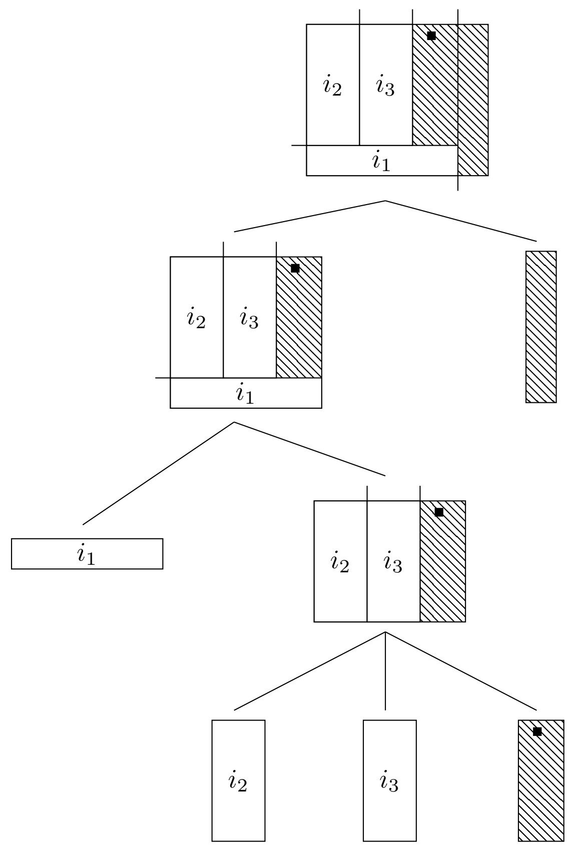



Figure 3: Tree representation of the cutting pattern from Figure 2

## 2.2 Problem objective and constraints

The problem objective is to reduce the glass loss for a given ordered bin set $\mathcal{B}$ and a given item batch to cut $\mathcal{I}$. This problem is related to the two-dimensional bin-packing problem but has extra constraints from industrial requirements. The problem consists in providing two-dimensional cutting patterns that allow to cut all items of the batch $\mathcal{I}$ using the available bins from $\mathcal{B}$ by satisfying some given constraints like avoiding defects, respecting the partial order on item stacks and bins and satisfying some physical constraints like guillotine cuts. Therefore, a solution can be represented by a set of cutting patterns. The constraints of the problem are defined hereinafter but the objective function is described first.

Formally, the objective function is to minimize the geometrical loss of the cutting patterns applied on bins. Since glass leftover can be reused after cutting, one has to take them into account to compute the loss area. To simplify the residual management, only the residual in the last cutting pattern is considered. It is represented by the waste at right of the last 1-cut performed in the last cutting pattern in a solution. The residual of the pattern in Figure 2 corresponds to the dashed part extract after the 1-cut. Let $\mathcal{P} = \{p_1, p_2, \ldots, p_m\}$ be a feasible solution, *i.e.* a set of $m$ cutting patterns satisfying all the constraints of the problem. The loss is measured by summing up the loss area in cutting patterns $\{p_1, p_2, \ldots, p_{m-1}\}$. For the loss in the last cutting pattern $p_m$, only its non residual part is taken into account. Let $r_m$ be the residual part of the pattern $p_m$. The objective function is :

$$\min \quad HWm - Hr_m - \sum_{i \in \mathcal{I}} w_i h_i \tag{1}$$

Glass cutting is subject to some constraints that are related to the items, the bins, the guillotine process, etc. A constraint can be physical, related to the cutting of glass, or organizational to satisfy the orders. An overview of these constraints is presented below. The solutions provided in this challenge must satisfy all these hard constraints. The constraints related to item and bins are the following:

- Only rotation by 90°of items is allowed, *i.e.* the items can be set horizontally or vertically on cutting patterns.
- All items in the batch $\mathcal{I}$ must be cut, *i.e.* a solution must contain the entire batch.
- Item overproduction is not allowed, *i.e.* only the items in $\mathcal{I}$ should be cut and a solution with extra items in cutting patterns is not valid.
- Each item belongs to the stack given in input and modifying the stack of items is not allowed.
- The given sequence of items within stacks in cutting patterns has to be satisfied among cutting patterns. For this, it is important to understand

how to rebuild the sequence of items from a cutting pattern. In the cutting pattern represented in Figure 2, the number on each item represents the output order of this item after snapping guillotine lines in depth first. This means that item $i_1$ is cut first, then item $i_2$, finally item $i_3$. Each item $i_{j'}$ is cut before item $i_{j'+1}$ for all $j' = \{1, 2\}$. This sequence of items can be found with a depth-first-search in the corresponding tree representation in Figure 3. Moreover, since the order of items are inside stacks only, there is no order to ensure between items of different stacks.

- The bins are always horizontal ($W > H$) and their rotation is not allowed.
- The bins must be used in the given order, *i.e* for a feasible set of cutting patterns $\mathcal{P} = \{p_1, p_2, \ldots, p_m\}$, the bins used to create patterns $p_1, p_2, \ldots, p_m$ must correspond exactly to bins $b_1, b_2, \ldots, b_m$ in that order.

The constraints related on how to build a cutting pattern are given below:

- Overlapping of items between them is not allowed.
- Overlapping of an item with one or more defects is not allowed in cutting pattern, *i.e.* an item has to be defect-free.
- It is forbidden to cut through a defect.
- Cutting patterns are two-dimensional and be obtained using guillotine cuts only. The number of cuts allowed to obtain an item is at most 3 (1,2,3-cuts only). However, it is possible to perform at most one 4-cut in a sub-plate obtained after a 3-cut. This means that only one 4-cut is allowed only if no additional cut is required to get two items or one item and a waste. This is known in the literature as allowing trimming. An example is given in Figure 4.

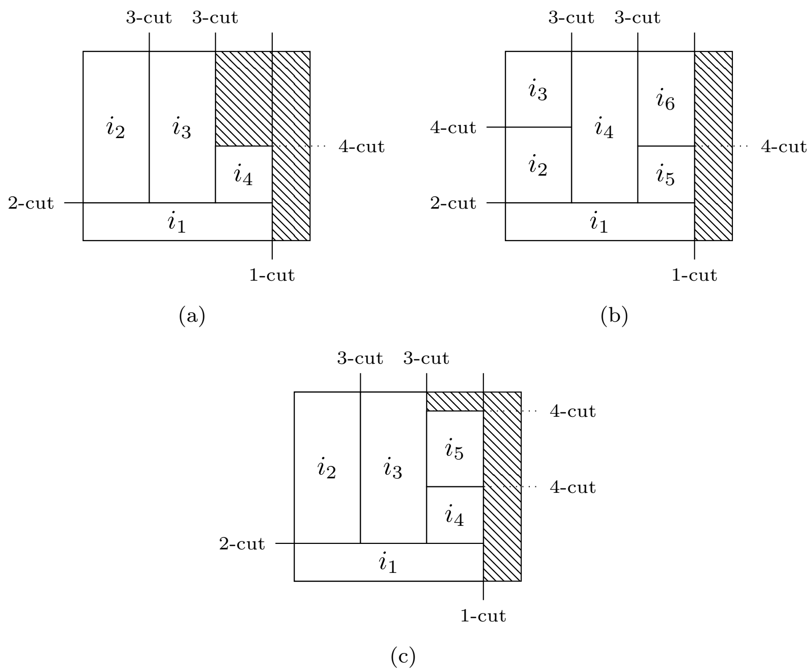



Figure 4: Example of valid cutting patterns (a-b) and a forbidden pattern (c). In pattern (a), the 4-cut is used only to remove the waste to get item $i_4$. In pattern (b), two 4-cuts are used to cut items $i_2, i_3$ and items $i_5, i_6$. This configuration is allowed since a 4-cut is performed in each plate obtained after a 3-cut. In pattern (c), two 4-cuts are performed to cut items $i_4, i_5$ and a waste. This configuration is forbidden since two 4-cuts are performed in the same plate obtained after a 3-cut.

- It is possible to cut an item in less than 3 cuts (see item $i_1$ in Figure 4-(a)).
- 1-cuts are always vertical as shown in Figure 4.
- The minimal width between two consecutive 1-cuts is 100, except for wastes.
- The maximal width between two consecutive 1-cuts is 3500, except for the residual.
- Due to the minimal and maximal widths between two consecutive 1-cuts and from technical limitation, a cutting pattern must contain at least one 1-cut.
- The minimal height between two consecutive 2-cuts is 100, except for wastes.

- The width between two consecutive 1-cuts does not correspond necessarily to the size of an item. Figure 5 illustrates bounds between two consecutive 1-cuts or 2-cuts.

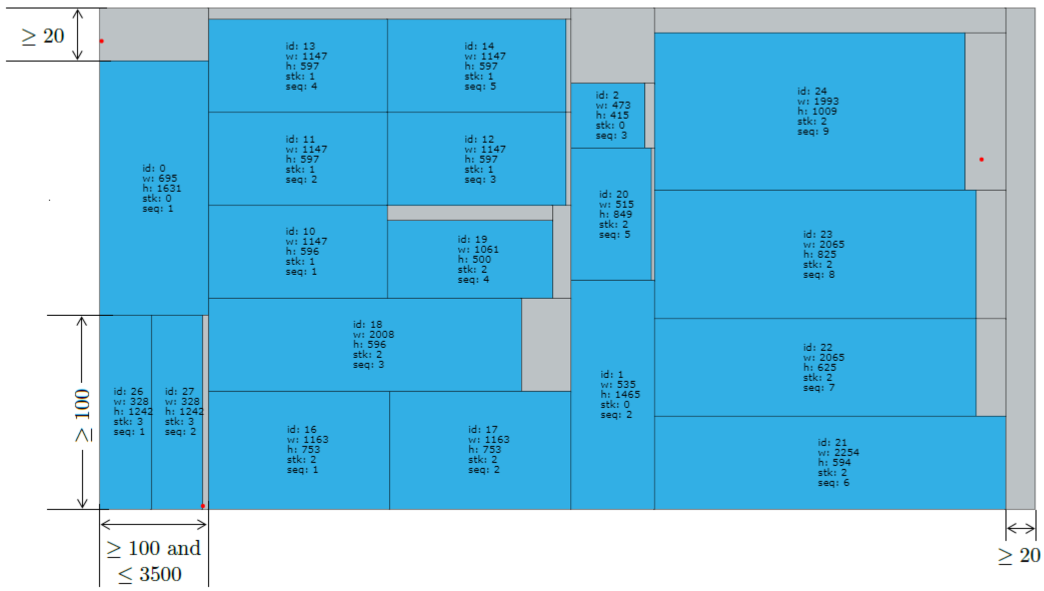



Figure 5: Bounds on distance between two consecutive 1-cuts and 2-cuts. Bounds are measured between two consecutive 1-cuts (or 2-cuts) and/or their distances to edges of the bin.

- Minimal size of every waste area is $(20, 20)$.

# 3 Example

In this section, a small problem instance is considered and its solution is explained. Let the following problem instance with item set $\mathcal{I}$ to cut composed of the following stacks: $\mathcal{I} = s_0 \cup s_1 \cup s_2 \cup s_3$ where $s_0 = \{i_0, i_1, \ldots, i_9\}$,$s_1 = \{i_{10}, i_{11}, \ldots, i_{15}\}$, $s_2 = \{i_{16}, i_{17}, \ldots, i_{25}\}$ and $s_3 = \{i_{26}, i_{27}\}$.

Figure 6 depicts a feasible solution to this instance. Two patterns $p_1$ and $p_2$ are used for bins $b_1$ and $b_2$. The bin size is (6000,3210). The used notation in Figure 6 for each item is:

**id:** corresponds to $i_{id}$ (identifier of the item).

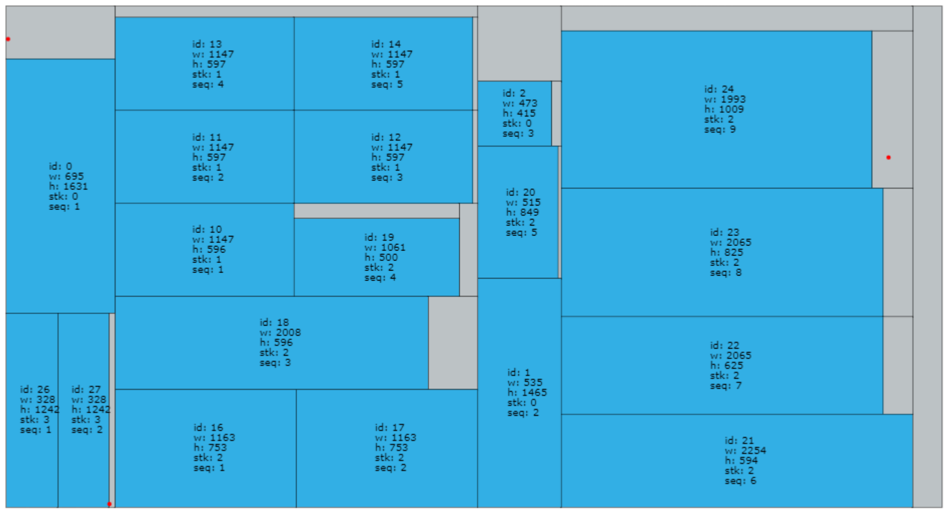
**w:** width of $i_{id}$, *i.e.* size of $i_{id}$ on x-axis of the cutting pattern.

**h:** height of $i_{id}$, *i.e.* size of $i_{id}$ on y-axis of the cutting pattern.

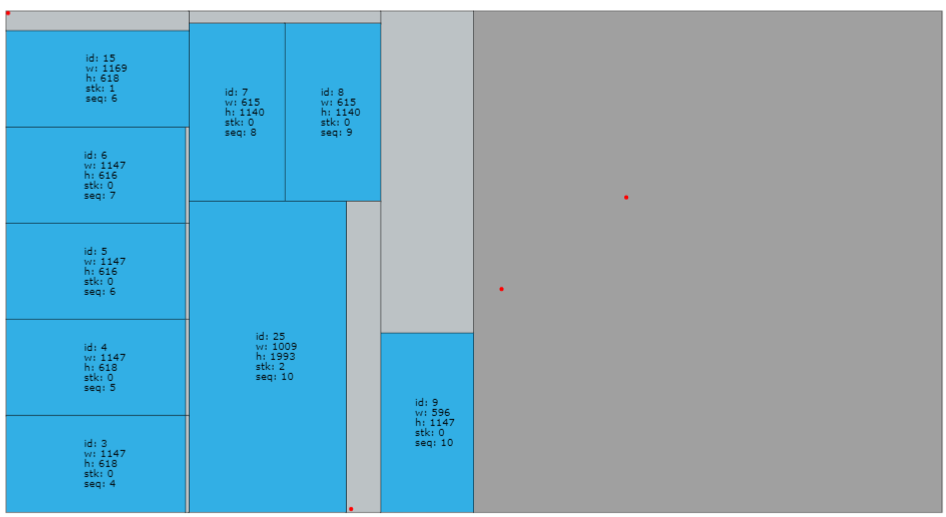
**stk:** corresponds to stack id $\{s_0, s_1, s_2, s_3\}$ in which the item belong $S_{stk} \ni i_{id}$.

**seq:** corresponds to the position of the item in its stack (from 1 to $|s_0|$ for instance).

From Figure 6, the items are cut in the following order (by snapping guillotine lines in depth first): $i_{26}$, $i_{27}$, $i_0$, $i_{16}$, $i_{17}$, $i_{18}$, $i_{10}$, $i_{19}$, $i_{11}$, $i_{12}$, $i_{13}$, $i_{14}$, $i_1$, $i_{20}$, $i_2$, $i_{21}$, $i_{22}$, $i_{23}$, $i_{24}$, $i_3$, $i_4$, $i_5$, $i_6$, $i_{15}$, $i_{25}$, $i_7$, $i_8$, $i_9$. It easy to check that the item are cut according to their position in their respective stacks. The order on bins is ensured since $p_1$ is applied on bin $b_1$ and $p_2$ on bin $b_2$. Since the solution only requires two patterns, $p_2$ is the last one. The sum of width of 1-cuts in $p_2$ is 2995, thus its residual $r$ is 3005.




(a)




(b)

Figure 6: Example of a solution with two cutting patterns $p_1$ (a) and $p_2$ (b) for bins $b_1$ and $b_2$. Red points are defects. Grey part represents the loss area, dark one is the residual area.