

ORIGINAL ARTICLE

Robust stock assortment and cutting under defects in automotive glass production

Claudio Arbib¹ | Fabrizio Marinelli²  | Mustafa Ç. Pınar³ | Andrea Pizzuti²

¹Dipartimento di Ingegneria/Scienze dell'Informazione e Matematica, Università degli Studi dell'Aquila, L'Aquila, Italy

²Dipartimento di Ingegneria dell'Informazione, Università Politecnica delle Marche, Ancona, Italy

³Department of Industrial Engineering, Bilkent University, Ankara, Turkey

Correspondence

Fabrizio Marinelli, Dipartimento di Ingegneria dell'Informazione, Università Politecnica delle Marche, Ancona, Italy.

Email: fabrizio.marinelli@staff.univpm.it

Handling Editor: Panos Kouvelis

Abstract

We address an assortment-and-cutting problem arising in the glass industry. The objective is to provide minimum waste solutions that are robust against such raw material imperfections as those possibly occurring with float glass production technology. The stochastic realization of defects is modeled as a spatial Poisson point process. A mixed integer program in the classical vein of robust optimization is presented and tested on data taken from a real plant application. Defective final products must in any case be discarded as waste but, if a recourse strategy is adopted, faults in glass sheets can sometimes be recovered. Closed forms for the computation of faulty item probabilities are provided in simple cases, and obtained via Monte Carlo simulation in more complex ones. The computational results demonstrate the benefits of the robust approach in terms of the reduction of back-orders and overproduction, thereby showing that recourse strategies can enable nonnegligible improvements. Encouraged by this result, the management is presently evaluating the possibility of adopting the proposed model in plant operation.

KEYWORDS

assortment, cutting stock, glass production, robust optimization

1 | INTRODUCTION

In this paper, we deal with the problem of simultaneously (i) finding a limited assortment of stock items and (ii) cutting them with minimum trim loss to fulfill a known demand of different parts. We develop a formally rigorous framework to address the problem when stock sheets can be affected by faults, providing and testing mathematical models for both defect occurrence and robust optimization of the production process. In this way, we show how one can fulfill the requirements with virtually no overproduction, or pending back-orders due to defective items, and with the same (or even less) scrap as a faultless process.

1.1 | Scope of the paper

Our methodology is inspired by a mathematical model, assessed in practice, to optimize stock assortment and cut in

a real production plant. The industrial process in the title, in fact a standard in the production of automotive glass parts, makes use of so-called *float technology*, which consists of two main phases (see Figure 1). In the first phase, large rectangular glass sheets of various sizes and types are manufactured by widening/narrowing and, after cooling, cutting a ribbon of molten glass; the sheets produced are then sent to a warehouse: We will refer to them as the *large items*, with heights (lengths) roughly ranging between 2 and 3 m (4–6 m). In a second phase, small rectangular sheets (the *small items*) are cut from large ones taken from the warehouse, and sent to downstream departments for bending and refinishing.

An example of the optimization of such a process, reported by Arbib and Marinelli (2007, 2009), refers to a plant operated by Pilkington Italia and now owned by Nippon Sheet Glass Co. Ltd. The purpose is the simultaneous control of the assortment of large items produced in the first phase and of the trim loss generated in the second. The word *assortment* refers to the number of distinct large sizes that are stored in the warehouse at any time: this number must be limited to a certain amount p to reduce holding cost and

Accepted by Panos Kouvelis, after two revisions.

This is an open access article under the terms of the [Creative Commons Attribution](https://creativecommons.org/licenses/by/4.0/) License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

© 2022 The Authors. *Production and Operations Management* published by Wiley Periodicals LLC on behalf of Production and Operations Management Society.

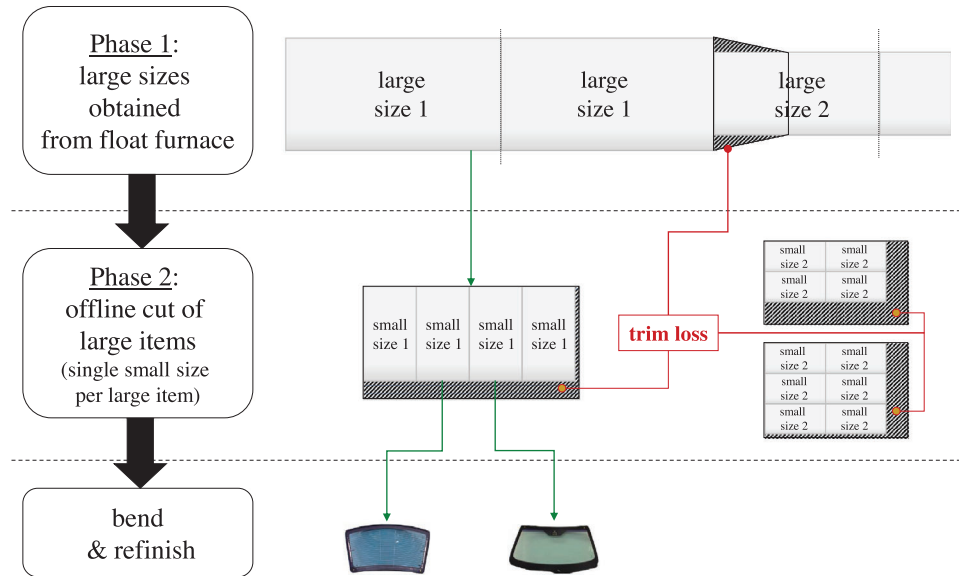


FIGURE 1 Scheme of an automotive glass production process [Color figure can be viewed at wileyonlinelibrary.com]

setups, and to ease handling operations. To give an idea of the problem scale, the production data set includes about 50 distinct small sizes and over 6500 distinct large sizes varying in the min–max height and length ranges with 1 cm pitch: As the desired assortment level p ranges around 20, over 10^{57} different assortment combinations can in principle be considered.

In Arbib and Marinelli (2009), the described stock assortment-and-cutting problem is formulated as a 0–1 linear program and solved by column generation or by an asymptotically exact heuristic. In this model, perfect knowledge of data is assumed. However, the glass ribbon may now and then come out from the furnace with defects (e.g., bubbles). The defective area of a large item cannot be recovered and is instead recycled, with a resulting reduction in yield: The quoted research, however, did not tackle this issue.

1.2 | Handling defect occurrence

Different reactions to the presence of defects (or, more generally, of forbidden areas) can be implemented depending on available technology, production line equipment, and work organization. In the literature, see Section 2 below, most papers on cutting optimization assume that defects are known in advance and given as input data. Consequently, the main focus is on the design of defect-free cutting patterns in a deterministic setting. In this paper, instead, we regard defect occurrence as a random process and approach uncertainty by using a robust optimization model. The model aims at fulfilling a given small item demand with a minimum volume of large items, and defect occurrence is modeled via a suitable uncertainty set, which impacts the solution by reducing the yield of the cutting phase.

Uncertainty is managed by mixing worst-case analysis typical of adversarial models with some stochasticity assumptions: The total number of defects in the planning horizon is a model parameter, and *how many* defects are placed in each large item is an adversarial decision; however, the adversary does not control *where* to place defects in each large item. Yield reduction is therefore modeled, as an expected value, via the probability that the defects present in a large item cannot be recovered.

This setting finds its justification in both problem-specific and practical reasons. On the one hand, uncertainty management is required because large sizes are not given beforehand, being part of the assortment decision: The optimal cutting patterns are in fact defined before the production of large items and hence before defects are detected. On the other hand, online reaction to defects, once these are spotted in a large item, is not practical either: In fact, although defect-free cutting patterns can in principle be computed just before loading large items into the cutting machines, the yield gain achieved would be completely absorbed by the additional costs of material handling, machine setups, and downstream departments operation scheduling.

To date, the plant management ignores any potential and even simple reaction, such as putting the defective large items aside and treating them separately, or changing the cutting patterns to reduce defect incidence: Although defects are detected already at the end of phase 1, they are not considered before the cutting stage, and small items affected by faults are simply discarded before the bending and refinishing stage. However, production numbers would definitely justify a policy against defectiveness. Glass imperfections cause in fact a loss of material, currently estimated around 8%–9%, a considerable magnitude given that the management deemed worth of optimization a trim loss estimated around 4%–5%. It is important to stress that discarded items have

an impact not only on the efficiency with which raw material is used but also, and more importantly, on the completion of order batches.

A question then arises whether an optimization model that is able to handle this type of uncertainty, perhaps coupled with a suitable recourse strategy, can reduce back-orders effectively while preserving efficiency in raw material usage. The result we obtained on a set of real-world instances provided by the plant shows that (i) by protecting against defects, robust optimization substantially reduces back-orders; and (ii) to be achieved, this result does not need the implementation of any particular (and possibly expensive) recourse strategy.

1.3 | Justification of the approach

In this section, we briefly discuss the advantages and costs of possible alternatives to our approach. Indeed, the problem we address is operational and does not consider investments in hardware. However, the specific plant operation should not limit the set of options one can consider. Hence, an analysis of technological and/or operational alternatives is necessary to demonstrate our methodological contribution. This analysis is to be carried out while recognizing that the productivity potentially achievable via technological updates (e.g., defect detection systems, new cutting machines) must, on one hand, justify important strategic investments and, on the other hand, provide a broad assessment of the impact on the whole organization.

- The main technological, but also operational, specificity of the plant considered in this work is that the production process is organized in two phases that are physically separate and time decoupled. In some plants, see, for example Durak and Tüzün (2017), the float line is equipped with defect sensors and bridges, perpendicular to the direction of the glass flow, which can perform cuts directly on the glass ribbon. Defect-free cutting patterns can therefore be designed and applied on-line, avoiding in this way assortment and overproduction issues on the large sizes. However, being integrated with the float line, the cutting system has only a few cutting wheels (≤ 5 in the cited work) that must be positioned at fixed coordinates during the whole production run. The result is a very limited number of feasible cutting patterns and, consequently, potentially high waste levels. A technological update of this type, then, would not imply a reduction of process costs.
- Another specificity of the plant is the way small items are cut from large ones. Downstream cutting machines allow only one size per large sheet. This configuration features a very short operation time and a single out-buffer, meaning high speed, elementary management of the output, and thereby a large plant throughput. Nevertheless, more complex cutting patterns could in principle reduce trim loss. However, complex patterns would make recourse strate-

gies very hard to be computed (and even implemented), and the robust model would become very complex and certainly impractical to solve. Moreover, should one attempt to address this complexity, the trim loss reduction would most likely be negligible. In fact, we find that the solutions obtained with the actual configuration are clearly feasible for machines able to perform more complex cuts, and that they represent a very good benchmark (trim loss below 1%).

- Checking *ex post* the demand not met because of defects, and recovering the requirements with an adequate extra-production, on-the-fly, of the necessary large items, is not viable in float glass production plants. The float production campaign is long, has a costly setup, and is not synchronized with the cut of small sizes. In other words, the small items cut at a given time are obtained from large sheets of a glass type (i.e., color and other technical features), which generally differs from the type the furnace produces at that time, and a float switch to that glass type is materially impossible. This is why any solution approach should take into account the uncertainty of defects realization.
- Finally, any operational reorganization of a plant like the one here considered cannot neglect set-up issues both at the furnace and in the cutting stage.

1.4 | Content

The paper is organized as follows. The literature on cutting problems with defects is briefly discussed in Section 2. Then, for convenience of the reader, the assortment-and-cutting problem described in Arbib and Marinelli (2009) is reported in Section 3 along with the relevant integer program. In Section 4, we introduce the uncertainty model that, in Section 5, is implemented in terms of robust optimization. Section 6 refines the description of the robust model, and an extensive computational experience performed on real-world instances is described in Section 7. The paper is completed with Supporting Information EC.2, where we describe the linearization of the robust model used in numerical tests, and the closed formulas devised to compute fault probabilities for large items with ≤ 2 defects.

2 | LITERATURE REVIEW

Due to both combinatorial richness and relevance in a variety of industrial contexts, a very large amount of theoretical and application-oriented papers addressing cutting optimization problems appeared in the scientific literature during the last six decades, that is, since the seminal papers by Gilmore and Gomory (1961), see, for example, the surveys by Wäscher et al. (2007) and Delorme et al. (2016).

Generally speaking, raw material is always potentially affected by imperfections: knotholes in wooden trunks and boards; bubbles in glass; contaminated areas in steel coils; holes, stains, or streaks in paper and leather sheets. This

means we find the topic addressed in such industrial processes as lumber and furniture, see Ghodsi and Sassani (2005) and Wenshu et al. (2015); shoes and textile, see Özdamar (2000) and Sarker (1988); and paper, see Aboudi and Barcia (1998). However, relatively few contributions in the cutting and packing literature focus on the problems that arise when cutting defective items.

One of the earliest publications on cutting problems with defects dates back to 1968 (Hahn, 1968). Several variants concerning the presence of one (Carnieri et al., 1993; Neidlein et al., 2009) or more (Afsharian et al., 2014; Wenshu et al., 2015) defects per stock item have been studied since then, both in one and two dimensions. Defect kinds range from simple points to rectangular or convex areas, and from faults causing immediate scrap of the items, to defects which only reduce (at various grades) production quality. A related issue, defect handling, is addressed in several ways: from the optimal use of clear zones (i.e., areas unaffected by faults) to cut optimization that assumes product values varying with fault content and/or placement in the stock items (see Sarker, 1988). Those problems are almost always modeled as mixed integer linear programs and solved by dynamic programming (Durak & Tüzün, 2017; Rönnqvist & Åstrand, 1998), Lagrangean relaxation and subgradient optimization (Rönnqvist, 1995).

In most contributions we are aware of, including those referenced above, the defect list, size, and location are given and known in advance. In some cases, the cutting pattern is also known, and the problem calls for finding a new layout that avoids as many defects as possible. This problem is hard even in the one-dimensional case, which is known as the Minimum Defective Subset Sum, see Aboudi and Barcia (1998). To our knowledge, only the relatively old paper by Sculli (1981) handles defects with a stochastic approach: Here, the size of a one-dimensional roll with fringe defects caused by winding is treated as a normal random variable. Expected waste is expressed as a given cumulative density function of the unusable parts at the edges of the roll, and then the optimal position of the knives is obtained analytically. An online optimization setting in glass manufacturing was introduced by Durak and Tüzün (2017); here, cutting bridges directly operate on the glass ribbon, and cuts are performed a few seconds after defects are detected. Therefore, at any given time, defect size and positions are known only for a limited number of stock items, precisely those between the sensor detectors and the cutting bridge. Nevertheless, that partial information allows the authors to address glass defectiveness by solving a sequence of deterministic cutting problems over a rolling horizon. In summary, Sculli (1981) assumes random defectiveness that only affects the edges of one-dimensional roll items, while the application described in Durak and Tüzün (2017) is very similar to our setting, but the authors treat defect positions as known data.

Rather than defect handling, papers on cutting problems deal with uncertainty especially to address the stochastic behavior of costs and demand. For example, robust optimiza-

tion is adopted in Alem and Morabito (2012) to address a combined lot-sizing and cutting-stock problem, and in Ide et al. (2015) with application in the wood cutting industry. Uncertainty in customer demand is addressed by two-stage stochastic optimization in Alem et al. (2010) and Beraldi et al. (2009). A cutting stock problem with random customer demand and random cutting times is investigated in Krichagina et al. (1998) and approximated by linear programming and a dynamic control problem involving Brownian motion. Finally, a branch-and-price method for robust optimization is proposed in Schepler et al. (2022) to address one-dimensional bin packing with items of uncertain sizes.

3 | THE STOCK ASSORTMENT-AND-CUTTING PROBLEM

The glass cut process described in Arbib and Marinelli (2007, 2009) involves a first phase that produces large rectangular items, which are to be cut, during a second phase, into desired amounts of small rectangular items of given sizes. The 0–1 linear programming model proposed in Arbib and Marinelli (2009) to optimize the cut process can be summarized in the following way.

Let S, L , respectively, denote the sets of small sizes required, and of all the feasible large sizes that can in principle be produced in the first phase. Let also d^i , $i \in S$, indicate the number of small items of the i -th size that are demanded, and have therefore to be produced, in a given planning horizon. We define an optimal assortment as

a set of no more than p large sizes that, among all the possible choices,

allow the cut of all the demanded small items with a minimum trim loss.

The rules adopted in the plant and the technological constraints of the downstream cutting machines greatly simplify the cutting patterns, that are limited to those obtaining small items of the same size from each single large item. In particular,

- the small items of a pattern have all the same orientation and, of the two possible orientations, we always select the most productive one;
- the first guillotine cut is always performed horizontally, that is, along the width of the large item, and trim loss is always located on the bottom right side of the large item.

Therefore, referring to a subset $P \subseteq L$ of large sizes, the minimum glass area c_P^i used to cut the required quantity of items of the i -th small size from large items of sizes $k \in P$ can be computed via the following integer knapsack problem:

$$c_P^i = \min \sum_{k \in P} V_k y_k^i \quad (1)$$

$$\sum_{k \in P} a_k^i y_k^i \geq d^i$$

$$y_k^i \geq 0 \text{ and integer,}$$

where $V_k = W_k H_k$ (width \times height) is the area of the k -th large item, a_k^i is the maximum (known) number of small items of the i -th size that can be obtained by cutting one large part of the k -th size, and y_k^i is the number of large items of the k -th size used to get the required number of items of the i -th small size.

Once the c_P^i are known for all small sizes $i \in S$ and for all sets $P \subseteq L$ with no more than p elements, the problem of finding an optimal assortment reads

$$\min \sum_{i \in S} \sum_{P \subseteq L} c_P^i x_P^i \quad (2)$$

$$\sum_{P \subseteq L} x_P^i = 1 \quad i \in S$$

$$\sum_{P \subseteq L: P \ni k} x_P^i \leq u_k \quad i \in S, k \in L$$

$$\sum_{k \in L} u_k \leq p$$

$$x_P^i \geq 0, u_k \geq 0 \text{ and integer} \quad k \in L, i \in S, P \subseteq L.$$

In problem (2), variables are implicitly bound to get values in $\{0, 1\}$. $x_P^i = 1$ means that the whole requirement of the items of the small size i is produced by cutting large items of sizes in the set P and $u_k = 1$ indicates that at least one large item of the k -th size in L is used. The goal is to minimize the total glass area used for the whole production. The first set of constraints ensures that the requirement of each small size is covered by some set P of large sizes. The subsequent inequalities require that if a set P containing the k -th large size is used for production, then that size is part of the assortment chosen. The last inequality limits this assortment to p distinct sizes.

Since the number of x_P^i variables grows rapidly, being $O(|L|^p)$ very large, problem (2) is solved in Arbib and Marinelli (2009) by column generation, pricing variables via (1). In the same paper, an efficient heuristic is also proposed. The idea is to solve integer program (2) only with the variables $x_P^i = x_k^i$ that corresponds to singleton sets $P = \{k\}$. In this way, each small item is produced by large items of the same size (while a given large size can still be used to produce different small sizes). The problem boils down to a p -median (see (13) in Section 5) since it reduces to cluster the small sizes into at most p subsets, and one can prove that its optimum asymptotically approaches the true one, as the demand of the least required small item increases.

In practice, the p -median heuristic is very effective and permitted a reduction of the plant trim loss to one third of the initial amount. In the sequel, we will then refer to this

simplified model, that is, to program (2) written with x_k^i for x_P^i , and call it $[P]$ for short. To limit the number of indexes in the models described in the following sections, we define the set of all possible production lots of items in S via items of L :

$$J = \{j = (i, k) : i \in S, k \in L\}, \quad (3)$$

therefore setting $x_j \triangleq x_k^i, a_j \triangleq a_k^i$. Then, we introduce

$$J_k = \{j \in J : j = (i, k), i \in S\}$$

$$J^i = \{j \in J : j = (i, k), k \in L\} \quad (4)$$

to denote all the lots using large size k and all those returning small size i , respectively. Finally, we indicate as

$$\bar{y}_j \triangleq \min\{y_k^i : a_k^i y_k^i \geq d^i, y_k^i \text{ integer}\} = \left\lceil \frac{d^i}{a_j} \right\rceil \quad (5)$$

the optimal solution of (1) for $P = \{k\}$. That is, for $j = (i, k)$, \bar{y}_j denotes the amount of large items of size k that are used when fulfilling the whole demand of items of size i .

4 | UNCERTAINTY MODEL AND RECOURSE STRATEGIES

Problem (1), and indirectly problem (2), assumes parameters a_k^i to be known integer constants. Due to the nature of the process, however, those parameters are in fact uncertain.

The most widely known approaches to deal with uncertain data are stochastic and robust optimization, see Ben-Tal et al. (2009) and Birge and Louveaux (2011).

The former requires that the probability distributions and possible scenarios are known in order to describe the random variables, and can lead to very large models that are hard to solve. For this reason, it will not be considered here. In the latter, the range of uncertainty can be derived from historical realizations of the random variables. Robustness can be addressed in various ways. The strict concept early proposed by Soyster (1973), where an optimal solution is always feasible for every realization of the random parameters, was later mitigated by Bertsimas and Brown (2009) and Bertsimas and Sim (2004), who proposed an approach where optimal solutions are protected against the change of a given number of uncertain coefficients, generally related to each other in some mathematical way. Compared to other proposals in the literature, and according to Bertsimas and Sim (2004), one of the benefits of a robust approach is that the robust counterpart remains computationally practical independently of the number of uncertainty parameters, meaning that at least it remains in the realm of integer linear programming. This is the main reason that led us to adopt this approach.

Let V_F be an estimate of the total float volume necessary to produce the required small items (a simple lower bound to V_F is $\sum_i d^i v^i$, where $v^i = w^i h^i$ is the area of the i -th small size). A stricter lower bound can be computed by solving problem (2). According to the indications provided by the plant management, defects can be considered point shaped, statistically independent, and uniformly distributed in V_F . Their location can then be usefully modeled as a spatial point process \mathbf{X} , see Daley and Vere-Jones (1988). In particular, we assume that the process is *simple*, that is, no two points ever coincide, and *stationary*, that is, for any fixed point with coordinates \mathbf{x} , the distribution of the point process obtained by shifting each point of \mathbf{X} by \mathbf{x} is identical to the distribution of \mathbf{X} .

More precisely, let $\rho > 0$ be the expected number of defects per unit area and $T(k)$ be a random variable giving the number of defects within the k -th large item, whose area is $V_k = W_k H_k$. On the basis of the above management indications, it is reasonable to assume $\mathbb{E}[T(k)] = \rho V_k$, $\text{Prob}\{T(k) > 1\} = o(V_k)$ for V_k small compared to the float volume V_F , and that defect realizations in disjoint regions (and hence in distinct large items) occur independently. Hence, the random process can be described as a spatial Poisson process with uniform intensity ρ , that is, a point process in \mathbb{R}^2 such that:

- for each large item of size k , $T(k)$ has a Poisson distribution with mean ρV_k ;
- $T(1), T(2), \dots$ are statistically independent.

Due the conditional property of a Poisson point process, the distribution of $T(\cdot)$, conditioned to a total number f of defects in a float campaign, is binomial:

$$\begin{aligned} \varphi_k^t &\triangleq \text{Prob}\{T(k) = t | T(F) = f\} \\ &= \binom{f}{t} \left(\frac{V_k}{V_F}\right)^t \left(1 - \frac{V_k}{V_F}\right)^{f-t}. \end{aligned} \tag{6}$$

Using (6) we can easily evaluate, for each large item, the number t of defects that one can reasonably expect for a given f . As we will see in Section 7, practical cases are essentially covered by values of t between 0 and 6. Accordingly

$$\varphi_k = \sum_{t=1}^f \varphi_k^t \tag{7}$$

will be used to estimate the faulty fraction of large items of size k .

The random variable $T(\cdot)$ is, however, not sufficient to describe yield reduction, which also depends on (i) the position of defects in the large sheet, (ii) the small item produced with that sheet, and (iii) what kind of response to the presence of defects is implemented.

Concerning the third issue, depending on cut feasibility, we consider three degrees of pattern reconfigurability (see Figure 2):

- *No reconfiguration, NR*. Patterns remain unchanged: A defect always causes a yield loss, unless it falls into the scrap area. This is the policy presently adopted in the plant.
- *Constrained reconfiguration, CR*. Patterns can be reconfigured provided that no more cuts are employed than the minimum used in normal operation: That is, strips of scrap can be moved from the bottom or right edge of the large item, but cannot be split up. At present, the plant management is evaluating this option.
- *Unconstrained reconfiguration, UR*. Patterns can be reconfigured also by splitting scrap strips with additional cuts. This recourse strategy is more general and effective (but more complex).

A maximum of 187 small items (laid out on 17 strips with 11 items each) can be obtained from a large item in the instances provided by the plant. A CR that avoids as many faults as possible can easily be obtained by checking all the possible positions of the horizontal and vertical scrap strips. If the cutting pattern consists of n horizontal and m vertical strips of small items, the horizontal and vertical scrap strips can be placed in $n + 1$ and $m + 1$ places, respectively. Consequently, there are no more than $(n + 1)(m + 1)$ possible reconfigurations, among which one chooses that with the least number of small items damaged by defects occurred in the large item.

By contrast, computing the best UR requires a tailored approach, which is beyond the scope of this work.

A defect still causing a yield loss after reconfiguration is called *critical*. Let Q denote a random variable giving the number of critical defects in a large item. For any production lot $j = (i, k) \in J$ and any integer $0 \leq q \leq t$, let

$$\pi_{jq}^t \triangleq \text{Prob}\{Q = q | T(k) = t\} \tag{8}$$

be the probability that a single large item of size k used for the production of small items of size i has q critical defects out of t . We use (8) to parameterize the uncertainty set U , which the yield parameter a_j belongs to, with respect to the expected number f of defects in a float campaign, see Section 5. In particular, for a pair $j = (i, k)$ with *zero trim loss* (i.e., W_k and H_k integer multiples of w_i and h_i) under *NR* strategy we have

$$\pi_{jq}^s = (p_j/a_j)^s \binom{a_j}{q} \left\{ \begin{matrix} s \\ q \end{matrix} \right\} \cdot q!, \tag{9}$$

where $a_j = \lfloor \frac{V_k}{v^i} \rfloor$, $p_j = \frac{a_j v^i}{V_k}$ are the largest number of small items of type i contained in a large item of the k -th size and the probability that one of those small items is hit by a random fault in the large item, respectively. The third factor in the above expression denotes the Stirling number of II type, recursively computed as

$$\left\{ \begin{matrix} s \\ q \end{matrix} \right\} = \left\{ \begin{matrix} s-1 \\ q-1 \end{matrix} \right\} + q \left\{ \begin{matrix} s-1 \\ q \end{matrix} \right\}, \tag{10}$$

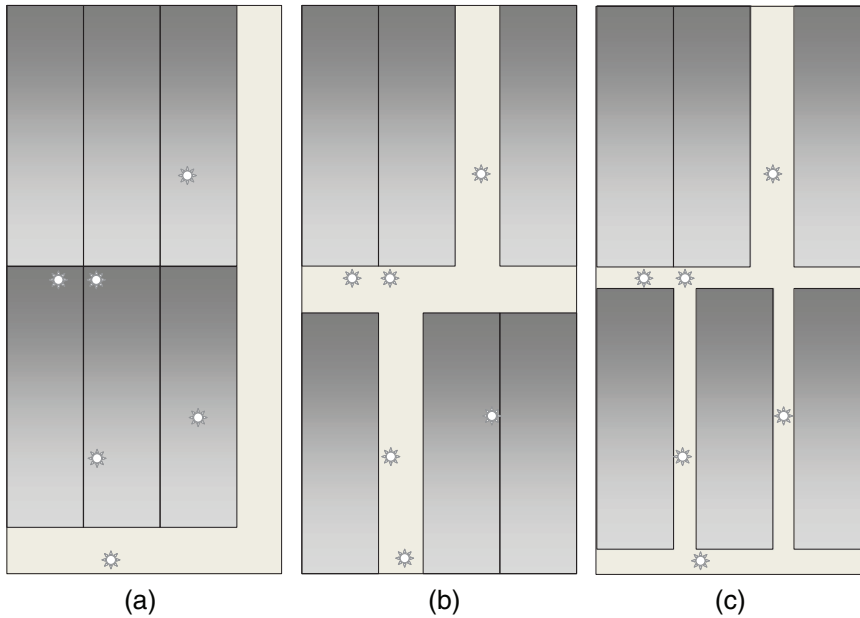


FIGURE 2 (a) No reconfiguration, (b) constrained, and (c) unconstrained reconfiguration [Color figure can be viewed at wileyonlinelibrary.com]

with initial conditions

$$\begin{Bmatrix} s \\ 1 \end{Bmatrix} = \begin{Bmatrix} s \\ s \end{Bmatrix} = 1 \quad (11)$$

for all s . In fact, $\begin{Bmatrix} s \\ q \end{Bmatrix}$ counts the different ways of partitioning s defects into q nonempty subsets, which in turn can be chosen in $\binom{a_j}{q}$ different ways and associated to the defect sets in $q!$ distinct ways.

For *positive trim loss*, we use formula (9) for $s = t, t-1, \dots, q$, introducing a factor $\binom{t}{s} p_j^s (1-p_j)^{t-s}$ to take into account the probability that exactly s out of the t defects hit a small item. We obtain in this way

$$\pi_{jq}^t = \binom{a_j}{q} \cdot q! \sum_{s=q}^t \binom{t}{s} \begin{Bmatrix} s \\ q \end{Bmatrix} \frac{p_j^s (1-p_j)^{t-s}}{a_j^s}. \quad (12)$$

For more sophisticated recourse strategies the expression of π_{jq}^t becomes more complex. Closed formulas to compute π_{jq}^t for $t = 1, 2$ under *CR* and *UR* strategies are provided in the Supporting Information sections EC.2.1 and EC.2.2.

5 | ROBUST MODEL

Formulation (2) with separate pricing (1) is not suitable for modeling uncertainty in the vein of robust optimization, because delayed column generation cannot manage the bundle of uncertain coefficients that goes across the pricing problems. However, the good features of the p -median heuristic described in Arbib and Marinelli (2009) can help in two ways. On one hand, we can directly modify it to try and

address the losses due to defects. On the other hand, we can use it as a starting point to define the (more sophisticated) robust counterpart.

Let us first focus on the first approach. According to the notation in Section 3, for any lot $j = (i, k) \in J$, let $x_j \in \{0, 1\}$ be a binary variable indicating that the whole demand of items of small size $i \in S$ is fulfilled by cutting large size $k \in L$. One can then correct the objective function of the p -median model to take into account the expected loss due to defects. The model reads:

$$\begin{aligned} [P] \quad & \min \sum_{j \in J} c_j x_j & (13) \\ & \sum_{j \in J^i} x_j = 1 & i \in S, \\ & u_k - x_j \geq 0 & k \in L, j \in J_k, \\ & \sum_{k \in L} u_k \leq p, \\ & x_j, u_k \in \{0, 1\} & j \in J, k \in L, \end{aligned}$$

where $c_j = V_k \lceil \frac{d_i}{a_j - \bar{a}_j} \rceil$ is the total expected area used by lot j .

In particular, by overestimating (resp., underestimating) by $t_k = \lceil \rho V_k \rceil$ (resp., by $t_k = \lfloor \rho V_k \rfloor$) the expected number of defects in large item k , one obtains the expected number \bar{a}_j of defective items:

$$\bar{a}_j = \sum_{q=0}^{t_k} (q \pi_{jq}^t) \quad (14)$$

and then the expected number of nondefective small items of the i -th size obtained by one large part of the k -th size, $(a_j - \bar{a}_j)$, to be plugged in the definition of c_j .

Generally speaking, the extra small items required to compensate those lost by defects can be produced by keeping a pool of additional large items of a *generic* size. Observe however that such a simple strategy is largely dominated, in terms of loss minimization, by the solutions of [P] that already provide a suitable amount of large items of a specific size for each small size. As a reference figure, the trim loss of instance I_{15} obtained by [P] is 1.09% (see column V_T of Table 2 in Section 7). An alternative solution computed by the deterministic p -median model, that is, with $\bar{a}_j = 0 \forall j$, and then using the large items in the pool to recover the underproduction (of size 610×320 for instance I_{15} , that is, the largest size in L) has a total trim loss of 2.56%, more than twice that of [P].

With the second, robust approach, let ξ_j be a nonnegative integer variable that denotes the number of extra items of size $k \in L$ that must be cut to compensate the items $i \in S$ lost due to manufacturing defects. Finally, let z_j^t be positive integer parameters that count the number of those large items of size k with t defects that are involved in the production of the items of small size i (independently of the chances that the glass imperfection has to reduce yield, i.e., that a defect hits a small item). We can parameterize the total number f of faults that occur in the production period and, inspired by the budgeted uncertainty à la Bertsimas-Sim Bertsimas and Sim (2004), we define the uncertainty sets $U_z^i, i \in S$

$$U_z^i = \{z \in \mathbb{N}^{|J| \times f} : \sum_{j \in J^i} \sum_{t=1}^f t z_j^t \leq f^i\}, \quad (15)$$

where the budget controlling robustness f^i , an upper bound to the number of possible defects in the production of the items of small size i , is set to $\frac{v^i d^i}{V_F} f$. That is, it is assumed to be proportional to the area required by i (in (15) and the following, for simplicity of notation, we assume $z_j^t = 0$ for $t > f_i$).

Let now ℓ_{jq} be the number of large items, out of the $\sum_{t=1}^f z_j^t$ defective ones, where faults reduce yield by q units. The relevant uncertainty set U_ℓ depends on z and reads:

$$U_\ell = \{\ell \in \mathbb{N}^{|J| \times f} : \ell_{jq} \leq \sum_{t=q}^f \pi_{jq}^t z_j^t, j \in J, 1 \leq q \leq f\}, \quad (16)$$

where, by definition (8), π_{jq}^t is the probability that a generic large item in the j -th production lot outputs q faulty items when affected by $t \geq q$ faults. The robust counterpart of problem (2) then reads

$$\min \sum_{k \in L} V_k \sum_{j \in J_k} (\bar{y}_j x_j + \xi_j) \quad (17)$$

$$\sum_{j \in J^i} a_j (\bar{y}_j x_j + \xi_j) - \max_{\ell \in U_\ell} \sum_{j \in J^i} \sum_{q \leq f} q \ell_{jq} \geq d^i \quad i \in S, \quad (18)$$

$$\bar{y}_j x_j + \xi_j - \sum_{t=0}^f z_j^t \geq 0 \quad j \in J, \quad (19)$$

$$\sum_{j \in J^i} x_j = 1 \quad i \in S, \quad (20)$$

$$u_k - x_j \geq 0 \quad k \in L, j \in J_k, \quad (21)$$

$$\sum_{k \in L} u_k \leq p, \quad (22)$$

$$x_j, u_k \in \{0, 1\}, \xi_j \geq 0 \text{ and integer}, \quad (23)$$

$$z_j^t \in U_z, \ell_{jq} \in U_\ell. \quad (24)$$

The goal is to minimize the total glass volume used for production. For any given lot $j = (i, k)$, the terms V_k and \bar{y}_j in the objective function (17), respectively, indicate the area of the k -th large size and, as per (5), the number of items of large size k required to fulfill the demand of small size i . Inequalities (18) ensure fulfillment of small item requirements, discounting the losses due to large items imperfections. In these constraints, ℓ_{jq} is an uncertain coefficient varying in U_ℓ . Inequalities (19) state that the defective large items of size k cannot be more than those actually produced. Condition (20) limits the production of each small size to a single large size. The use of large size k is triggered by inequalities (21). Finally, inequality (22) limits the assortment of distinct large sizes used.

Unlike Bertsimas-Sim's, the uncertainty sets U_z^i and U_ℓ are not polyhedral as the parameters z and l are integer valued. To overcome this difficulty while preserving the degree of robustness, that is, all the integer points of U_z^i and U_ℓ , we first relax the integrality of ℓ_{jq} and, according to the definition of U_ℓ , replace its occurrences in (18) by $\sum_{t=q}^f \pi_{jq}^t z_j^t$. Furthermore, indicating by $\omega^i \geq 0$ the maximum loss due to defects in the production of small size i , constraints (18) are rewritten:

$$\sum_{j \in J^i} a_j (\bar{y}_j x_j + \xi_j) - \omega^i \geq d^i \quad i \in S. \quad (25)$$

The worst-case scenario occurs when, for $i \in S$, the number of defective small items of size i

$$\omega^i = \sum_{j \in J^i} \sum_{q=1}^f \sum_{t=q}^f q \cdot \pi_{jq}^t z_j^t \quad (26)$$

is maximized over the nonpolyhedral (discrete) uncertainty set

$$U_z^i = \{z \in \mathbb{N}^{|J^i| \times f} : \sum_{j \in J^i} \sum_{t \leq f} t z_j \leq f^i, \sum_{t=1}^f z_j \leq \varphi_k(\bar{y}_j x_j + \xi_j) \quad j = (i, k) \in J^i\}, \quad (27)$$

thereby removing (19) from (17)–(23). The first inequality limits to f^i the number of defects in the production of small parts of size i . The second limits to the portion estimated by (7) the number of defects in a generic production lot j that uses the k -th large item.

Maximizing (26) subject to (27) gives, for each $i \in S$, an integer optimization problem (as defined on a discrete uncertainty set) and, to our knowledge, general methods for discrete uncertainty sets only work for very few scenarios. However, an upper bound to the maximum ω^i can be obtained by the continuous relaxation of (26) and (27) assuming x_j and ξ_j momentarily fixed. Since such problems are feasible and bounded for any x_j and ξ_j , by strong duality the bounds exist for any $i \in S$ and can then be computed solving the LP dual of (26) and (27):

$$\begin{aligned} \min \quad & f^i \lambda^i + \sum_{j=(i,k) \in J^i} \varphi_k(\bar{y}_j x_j + \xi_j) \lambda_j \\ t \lambda^i + \lambda_j \geq \quad & \sum_{q \leq t} q \pi_{jq} \quad j \in J^i, t \leq f, \\ \lambda^i, \lambda_j \geq 0 \quad & j \in J^i. \end{aligned} \quad (28)$$

Now, any solution of (28) provides a dual bound to the loss in the worst case. Then, the max term in (18) can be replaced by the above dual objective function, which results in the following conservative approximation of the robust model (17)–(24):

$$[R] \quad \min \sum_{k \in L} V_k \sum_{j \in J_k} (\bar{y}_j x_j + \xi_j) \quad (29)$$

$$\sum_{j=(i,k) \in J^i} (\bar{y}_j x_j + \xi_j) (a_j - \varphi_k \lambda_j) - f^i \lambda^i \geq d^i \quad i \in S, \quad (30)$$

$$t \lambda^i + \lambda_j \geq \sum_{q \leq t} q \pi_{jq} \quad j \in J, t \leq f, \quad (31)$$

$$(20)–(23) \quad (32)$$

$$\lambda^i, \lambda_j \geq 0. \quad (33)$$

The resulting model has bilinear terms $x_j \lambda_j$ and $\xi_j \lambda_j$ that can however be linearized by standard techniques, see Supporting Information EC.2.3.

While it is not among the purposes of this paper to undertake a stochastic programming approach, a two-stage SP

formulation is sketched below in the interest of discussion. This is stated as

$$\min \sum_{k \in L} V_k \sum_{j \in J_k} \bar{y}_j x_j + Q(\xi) \quad (34)$$

subject to the constraints of [P] and the binary nature of x_k and u_k variables, where we define

$$Q(\xi) = \min \mathbb{E}_{\ell, z} \left[\sum_{k \in L} V_k \sum_{j \in J_k} \xi_j \right] \quad (35)$$

subject to constraints (18) and (19). Here, one has to deal with a two-stage model involving integer variables in both stages in addition to approximating (if one is unable to compute it accurately) an expected value over random variables z (the number of large items with t defects) and ℓ_q (the number of large items where faults reduce yield by q units). Without entering into the details of working with such a formulation, it is debatable whether it would lead to a more practical result or to superior performance in both computational and economical terms. There are several authoritative references on stochastic programming dealing with complexity issues and approximation procedures, for example, Shapiro et al. (2009). In terms of complexity, our approximate (robust) model remains in the realm of integer linear programming. It can be argued that a similar statement will be true for the case of SP model. However, it is not our place to develop this point further here. Nevertheless, we hope that the present paper inspired others to undertake further work along these lines.

6 | TECHNICAL DETAILS OF THE ROBUST FORMULATION

The complexity of model [R] rapidly grows with the number $|L|$ of distinct large sizes and of alternative patterns considered. To make resolution viable by commercial solvers, it is crucial to control the size of those sets. The latter set is in fact controlled, as model [R] adopts only one pattern per lot. Pattern features are summarized by parameter a_j , $j = (i, k)$, which counts the largest number of size i small items one can obtain by a single large item of size k . In this sense, each cutting pattern is a *maximal* one.

Although the probability of a faulty item decreases when a maximal pattern is turned into a nonmaximal one by removing one or more items, the adoption of maximal patterns is justified by the following result.

Proposition 1. *If an NR or CR strategy is implemented, see Section 4, model [R] always has an optimal solution that consists of maximal patterns only.*

Proof. Let us focus on a pair $j = (i, k) \in J$ and, for the sake of conciseness, drop the indexes of variables and parameters. Suppose that mn small items of size $w \times h$, arranged in n horizontal strips of m items each, are obtained by a maximal

pattern applied to a large item of size $W \times H$. Let $r = W - mw$ be the width of the vertical waste, and $s = (H - nh)$ the height of the horizontal one. After the CR of a maximal pattern, all the potential coordinates of each item's bottom-left corner are in $Z = (s\alpha + wi, r\beta + hi)$, with $i = 0, \dots, m - 1, j = 0, \dots, n - 1$, and $\alpha, \beta \in \{0, 1\}$ (see Figure 2b). Let \tilde{Z} be the analogous set of a nonmaximal pattern. Clearly, $\tilde{Z} \subseteq Z$, and the same trivially holds for the NR strategy. Therefore, the space left free by the items removed from a maximal pattern cannot be used to avoid defects on the remaining items. \square

As for the set L used in the formulations, in the original problem by Arbib and Marinelli (2007), candidate large sizes are preprocessed, for each instance, according to the following procedure. First, define the set L^0 of ideal large sizes for which there exists at least a small item i that can be cut without trim loss. Formally,

$$L^0 = \bigcup_{i \in I} L_i^0 = \bigcup_{i \in I} \{k \in L : W_k = \alpha_i w_i, H_k = \beta_i h_i\}, \quad (36)$$

where α_i, β_i are positive integers ranging in

$$\left\lfloor \frac{W_{\min}}{w_i} \right\rfloor \leq \alpha_i \leq \left\lceil \frac{W_{\max}}{w_i} \right\rceil, \quad \left\lfloor \frac{H_{\min}}{h_i} \right\rfloor \leq \beta_i \leq \left\lceil \frac{H_{\max}}{h_i} \right\rceil, \quad (37)$$

and $W_{\min}, W_{\max}, H_{\min},$ and H_{\max} , respectively, denote upper and lower bounds on the large sizes in L . A restricted set of large sizes sufficient to guarantee minimum trim loss solutions in the deterministic setting, that is, in the absence of defects, can be computed from L^0 in view of the following property proved in Arbib and Marinelli (2007):

Proposition 2. For any $h, k \in L^0$, let l_{hk} be the smallest size in L containing both h and k . Let $\tilde{L} = \{l_{hk} : h, k \in L^0, h \neq k\}$. Then, there is an optimal solution that uses only large sizes in \tilde{L} .

In our data set, \tilde{L} contains a number of sizes between 13 and 461, much smaller than the potential 6560 large sizes in L (see Table EC.1 in the Supporting Information Section EC.1), thus making viable the solution of the integer programs described in the above sections.

However, once the aleatory presence of defects is introduced, it is possible that \tilde{L} does not contain enough distinct sizes to warrant trim loss minimization. In fact, with more sizes available, the expected loss can be reduced at the expense of some additional waste. This consideration deserves a brief comment.

Consider the simple case of $d = 36$ items required in the size 0.9×2.5 , and suppose for simplicity that no more than one defect per large item ever occurs. Suppose also that \tilde{L} consists of the single ideal size 2.7×5 (hence $a = 6$ and $\bar{y} = 6$), and that $f = \rho V_F = 9$ defects are expected (i.e., f is sufficiently large to ensure $\varphi^1 = 1$ for each large item employed). With these values, the probability of a critical defect under the

CR strategy is $\pi_1^1 = 1$ (i.e., k is an ideal size), and an optimum is achieved by adding

$$\xi = \left\lceil \frac{d}{a - \pi_1^1} - \bar{y} \right\rceil = 2 \quad (38)$$

large items, for a total employed area of 108 square meters. But if one adopts a large size one decimeter wider, namely, 2.8×5 , the probability of a critical defect decreases to $\pi_1^1 = 0.857$ (see Section EC.2.1 in the Supporting Information), the additional large items needed reduces to $\xi = 1$ and the total area used decreases to 98 square meters (-9.26%).

Although in the example a minimum is achieved by a large size not in \tilde{L} , a preliminary numerical test shows that only 3 out of 15 instances benefit of a super-set of \tilde{L} , and the gain is always very marginal ($\leq 0.004\%$).

7 | COMPUTATIONAL EXPERIENCE

The models discussed above were tested on real data provided by the plant. Besides a comparison of CPU time, the test aims to evaluate the benefits, in terms of expected reduced loss and under/overproduction, obtained with random defects when the solutions of the robust model are implemented with different recourse strategies (see Section 4): NR, that is the action currently adopted in the plant, and CR. Because UR is quite far from the current practice and since, as noted in Section 4, computing the best UR is not straightforward, we left the UR strategy out from our numerical test.

The test compares four approaches:

- the deterministic p -median model $[P_0]$ obtained by setting $\bar{a}_j = 0$ in (13),
- the p -median models $[P_+]$ and $[P_-]$ derived from (13) by updating demand according to an over- and an underestimate of the expected number of defects per large item,
- the linearized version (described in Section EC.2.3, see the Supporting Information) of robust model $[R]$.

Solutions were tested by means of (i) a Monte Carlo simulation that generates defects according to a spatial Poisson process and (ii) an adversarial deterministic model that generates the maximum expected number of critical defects.

7.1 | Setting

The test was carried out on 15 problem instances, each one representing a production campaign: 10 instances are those used in Arbib and Marinelli (2007); the remaining five, namely, $I_2, I_3, I_7, I_8,$ and I_{14} in Table EC.1 in the Supporting Information, were newly provided by the plant. The details of test instances are reported in the Supporting Information Section EC.1. All the integer programs were solved using Gurobi

9.1.1 with default setting on a QUEMU virtual CPU 1.5.3 machine (8 cores) at 2.26 MHz with 8 GB of RAM. Gurobi precision parameter mipgap was set to $1e-9$. A limit for the running time was set at 7200 s.

Since, according to (6), seven or more defects in the same large item are relatively rare (a priori assumptions attribute to this event a probability of $4.14 \cdot 10^{-3}$), the probability that constraint (31) for $t \geq 7$ must be satisfied by solutions that already fulfill the same constraint for $t < 7$ is negligible. Indeed, we already obtained good results by limiting the model to $t \leq 2$. The probabilities π_{jq}^t for the NR strategy were computed by (12). For CR strategy, π_{j1}^1 and π_{jq}^2 were, respectively, obtained by (EC.1) and (EC.2) in the Supporting Information, while for $t = 3, \dots, 6$ the π_{jq}^t was estimated by Monte Carlo simulation. As we observed a strong correlation between number of items produced and area used (both in terms of overproduction and expected defective small items), in the following, we will comment on area results only.

7.2 | Analysis of results

Let $(\bar{x}, 0)$ and $(\bar{x}, \bar{\xi})$ be the solutions computed by the three [P] models and by [R], respectively, and let $\bar{Y}_j = (\bar{y}_j \bar{x}_j + \bar{\xi}_j)$ be the total number of large items used in lot $j \in J$. Let us then introduce the following figures:

- \bar{V}_F : total float area used:

$$\bar{V}_F = \sum_{k \in L} V_k \sum_{j \in J_k} \bar{Y}_j. \quad (39)$$

- V_R, \bar{V}_P : total area of small items required and produced:

$$V_R = \sum_{i \in I} v^i d^i \quad \text{and} \quad \bar{V}_P = \sum_{i \in I} v^i \sum_{j \in J^i} a_j \bar{Y}_j, \quad (40)$$

- \bar{V}_D : expected area of defective small items:

$$\bar{V}_D = \sum_{j=(i,k) \in J} v^i \bar{Y}_j \sum_{t=1}^f \sum_{q=1}^t q \sigma_{jq}^t, \quad (41)$$

where σ_{jq}^t is the probability that a large item in lot $j = (i, k)$ is affected by t defects, $q \leq t$ of which are critical. This is given by the compound probability $\varphi_k^t \pi_{jq}^t$, see (6) and (8). In the evaluation of these probabilities, we used $f = \rho \bar{V}_F$ and, for the sake of numerical precision, limited the binomial in (6) to $t \leq 10$.

The solution quality of $(\bar{x}, \bar{\xi})$ and $(\bar{x}, 0)$ is evaluated through the percent area overproduced V_O and the percent trim loss V_T . Namely,

$$V_O = 100 \cdot \frac{\bar{V}_P - V_R}{V_R}, \quad V_T = 100 \cdot \frac{\bar{V}_F - \bar{V}_P}{\bar{V}_F}. \quad (42)$$

The impact of defects on the solutions is taken into account by evaluating, for $(\bar{x}, \bar{\xi})$ and $(\bar{x}, 0)$:

- the percent deviation E_S of the expected area of sound items obtained versus the total area V_R required:

$$E_S = 100 \cdot \frac{(\bar{V}_P - \bar{V}_D) - V_R}{V_R}; \quad (43)$$

- the percent expected waste E_W , including both trim loss and discarded defective items:

$$E_W = 100 \cdot \frac{(\bar{V}_F - \bar{V}_P) + \bar{V}_D}{\bar{V}_F}; \quad (44)$$

- the percent expected number E_B of back-orders, that is, the percent of small sizes whose requirement was not completely fulfilled.

We computed \bar{V}_D not only by the above formula, but also via Monte Carlo simulation and by an adversarial integer program that models the worst-case scenario.

7.2.1 | Monte Carlo simulation

Computing \bar{V}_D requires to estimate the impact of defects on the solutions $(\bar{x}, 0)$ and $(\bar{x}, \bar{\xi})$. In a Monte Carlo simulation, each iteration reproduces the cut process according to a realization of glass defects, where defect number and positions are regarded as input random variables sampled from the appropriate probability distributions. Since in all the instances solved the small sizes are treated as integers, we can as well assume integer coordinates for all defect positions. Moreover, we can also assume the large items of any given solution sequenced in any arbitrary order. The outcome of each iteration (i.e., the defective area) is measured compliantly with the reaction adopted, that is, by just counting how many small parts are hit by a defect in case of NR strategy, by finding a reconfigured pattern that minimizes the faulty items (as briefly described in Section 4) in case of CR strategy.

The number of defects \hat{f} is sampled from a Poisson distribution with mean $\rho \bar{V}_F$ and process intensity ρ set to 0.1 points per square meter; then $[\hat{f}]$ positions, each indicating the two coordinates of a defect, are independently sampled from a uniform distribution in \bar{V}_F .

In particular, the large item \hat{k} and the coordinates (\hat{w}, \hat{h}) within \hat{k} of the sampled defect are obtained by first generating a random value $p \in [0, \sum_{j \in J} W_j H_j \bar{Y}_j]$ and then mapping p into \hat{k} and (\hat{w}, \hat{h}) in the following way. Let (\hat{j}, \hat{r}) be the last index such that $\sum_{j \in J} \sum_{r=1}^{\bar{Y}_j} W_j H_j < p$ and let $\hat{p} = \sum_{j=1}^{\hat{j}} W_j H_j \bar{Y}_j + \hat{r} W_{\hat{j}} H_{\hat{r}}$. Then,

$$\begin{aligned} \hat{k} &= \hat{r} + \sum_{j=1}^{\hat{j}} \bar{Y}_j, \hat{h} = \lfloor \frac{p - \hat{p} - 1}{W_{\hat{j}}} \rfloor + 1, \text{ and} \\ \hat{w} &= p - \hat{p} - W_{\hat{j}}(h - 1). \end{aligned} \quad (45)$$

Once all the defects have been generated, realizations of the area \hat{V}_D of defective small parts are computed and the statistic \bar{V}_D updated. The simulation is iterated $N = 10$ times at each call, terminating when $|\hat{V}_D - \bar{V}_D| < 10^{-6}$.

7.2.2 | Worst-case simulation

Given a solution $\bar{X} = (\bar{x}, \bar{\xi})$ and being $\bar{J} = \{j \in J : \bar{x}_j = 1\}$ the set of the production lots chosen with \bar{X} , the worst realization of defects with respect to \bar{X} can be computed by the following adversarial integer linear program:

$$\max \sum_{j=(i,k) \in \bar{J}} v^i \sum_{t=1}^f \sum_{q=1}^t q \pi_{jq}^t z_j^t \tag{46}$$

$$\sum_{t=1}^f t z_j^t \leq f^i \quad j = (i, k) \in \bar{J}, \tag{47}$$

$$\sum_{t=1}^f z_j^t \leq \varphi_k \bar{Y}_j \quad j = (i, k) \in \bar{J},$$

$$z_j^t \geq 0, \text{ integer} \quad j \in \bar{J}, 1 \leq t \leq f. \tag{48}$$

As in [R], z_j^t indicates how many large items in a produced lot j are hit by t defects. The objective function (46) equals the expected number of critical defects, to be maximized. By inequality (47), faults are distributed among large items respecting the maximum number f^i of defects allowed per small size. By constraint (48), the defective large items in each lot j cannot exceed, as a whole, the expected number $\varphi_k \bar{Y}_j$ of large items with faults employed by the solution.

7.3 | Solution analysis for a small instance

Before giving the aggregate experimental results, let us compare two solutions of instance I_2 , computed via $[P_0]$ and $[R]$, to illustrate the improvement obtained by the robust model. The demand of this problem, one of the smallest of our benchmark set, must be manufactured using no more than $p = 4$ large sizes. Given the float volume necessary for this production, we estimated $f = 476$ defects as a whole (see Table EC.1 in the Supporting Information). The first four columns of Table 1 detail request, size, and defect estimation for each small item $i \in S$ (the latter parameter being used in [R]). Next, section $[P_0]$ reports the large sizes selected by the p -median model to form the production lots j , with the associated cut productivity a_j and cut number \bar{y}_j . The rightmost section reports the same information for the solution computed by [R], and indicates also the extra large items $\bar{\xi}_j$ used to compensate losses due to defects.

The large sizes selected by model [R] partly differ from those chosen by model $[P_0]$: In detail, to produce small size 76×122 , $[P_0]$ uses the ideal large size 304×610 and so there

TABLE 1 Details of instance I_2 , and the solutions obtained by $[P_0]$ and $[R]$. In bold are the different large sizes chosen by [R].

Instance I_2	$[P_0]$				$[R]$								
	d^i	w^i	h^i	f^i	W_k	H_k	a_j	\bar{y}_j	W_k	H_k	a_j	\bar{y}_j	$\bar{\xi}_j$
600	56	211	78	280	450	10	60	280	450	10	60	7	
300	76	122	31	304	610	20	15	304	496	16	19	2	
150	46	124	10	280	450	18	9	304	496	24	7	0	
1500	52	180	155	312	540	18	84	312	540	18	84	8	
1500	53	142	125	318	580	24	63	318	580	24	63	4	
900	53	145	77	318	580	24	38	318	580	24	38	3	

is no trim loss in this case; [R] uses instead the smaller (but not ideal) size 304×496 . Also, $[P_0]$ produces small size 46×24 by the same large size used for the former, whereas [R] chooses 304×496 .

We launched a Monte Carlo simulation on both solutions, obtaining,

- for $[P_0]$, an expected waste $E_W = 10.05\%$, of which 2.08% is trim loss; the remaining 7.97% is the discarded defective area, not compensated at all by glass area overproduction ($V_O = +0.85\%$). In fact the solution entails high levels of expected sound item underproduction ($E_S = -7.37\%$) and back-orders ($E_B = 83.33\%$).
- For [R], +8.65% additional glass area (with $V_O = +9.73\%$), mild expected overproduction of sound items ($E_S = +0.82\%$) and no back-orders; the total waste E_W is reduced to 9.90%, 1.93% of which is trim loss.

In conclusion, the main result obtained with [R] is that a high back-order level is reduced to zero with no increase (actually, with a slight reduction) of total waste and trim loss.

7.4 | Results for NR strategy

For each instance, Table 2 describes the solution, respectively, computed by the deterministic p -median model $[P_0]$, by the probabilistic p -median models $[P_+]$, $[P_-]$, and by the robust model [R] when an NR strategy is adopted. In particular, the CPU columns report the running time (s) to get $(\bar{x}, \bar{\xi})$: A “-” indicates that the solver has reached the time limit; in that case, the subsequent column reports a nonzero optimality gap (%). The remaining columns give the overproduction V_O and the trim loss V_T found.

Tables 3 and 4 describe the protection that solutions offer against defects. This feature is evaluated by the expected values E_S , E_W , E_B , computed by Monte Carlo simulation and worst-case analysis. Table 5 shows the computational details of Monte Carlo simulations. For the worst-case analysis, we just observe that all instances of model (46)–(48) were solved in less than 0.125 s.

The p -median models (both deterministic and probabilistic) are much easier to solve than the robust counterpart of

TABLE 2 Solutions with *no reconfiguration* strategy. In bold are the averages over instances $I_1 - I_{11}$.

Inst	$[P_0]$			$[P_+]$			$[P_-]$			$[R]$			
	CPU (sec)	V_O (%)	V_T (%)	CPU (sec)	V_O (%)	V_T (%)	CPU (sec)	V_O (%)	V_T (%)	CPU (sec)	gap (%)	V_O (%)	V_T (%)
I_1	0.06	0.01	0.32	0.02	10.78	0.31	0.02	5.26	0.32	0.31	0	8.07	0.32
I_2	0.09	0.85	2.08	0.03	12.95	2.14	0.03	7.68	1.67	1.39	0	10.43	1.94
I_3	0.08	0.02	1.57	0.02	12.78	1.64	0.02	6.21	1.60	0.79	0	9.53	1.59
I_4	0.17	0.05	1.16	0.05	8.18	1.11	0.05	4.09	1.09	13.14	0	6.45	1.16
I_5	0.20	0.02	0.51	0.05	12.08	0.46	0.05	5.90	0.48	51.55	0	9.09	0.51
I_6	0.17	0.17	1.70	0.02	16.25	1.73	0.02	7.97	1.69	55.07	0	11.82	1.71
I_7	0.53	0.03	0.35	0.06	11.96	0.34	0.06	5.85	0.33	7037.28	0	8.91	0.36
I_8	0.20	0.03	2.27	0.06	11.49	2.34	0.06	5.65	2.27	755.73	0	8.64	2.31
I_9	0.30	0.04	1.11	0.13	9.71	1.15	0.13	4.78	1.15	4087.64	0	7.66	1.14
I_{10}	0.45	0.08	0.95	0.11	11.46	0.87	0.11	5.70	0.88	5115.95	0	8.35	0.97
I_{11}	0.39	0.02	1.11	0.11	11.94	1.12	0.11	5.82	1.10	-	0.52	8.83	1.12
Avg	0.24	0.12	1.19	0.06	11.78	1.20	0.06	5.90	1.14	1711.88	0.05	8.89	1.19
I_{12}	1.41	0.01	0.49	0.31	11.21	0.46	0.31	5.46	0.47	-	2.23	5.97	0.50
I_{13}	1.89	0.03	0.67	0.39	12.51	0.68	0.39	6.11	0.65	-	3.16	6.88	0.69
I_{14}	1.70	0.01	2.76	0.61	24.89	2.78	0.61	11.75	2.76	-	6.80	12.21	2.83
I_{15}	3.39	0.03	1.05	0.72	11.15	1.09	0.72	5.45	1.08	-	5.47	8.06	1.07

TABLE 3 Results of Monte Carlo simulation. In bold are the averages over instances $I_1 - I_{11}$.

Inst	$[P_0]$			$[P_+]$			$[P_-]$			$[R]$		
	E_S (%)	E_W (%)	E_B (%)	E_S (%)	E_W (%)	E_B (%)	E_S (%)	E_W (%)	E_B (%)	E_S (%)	E_W (%)	E_B (%)
I_1	-7.24	7.54	100.00	2.75	7.54	0.00	-2.37	7.54	100.00	0.24	7.54	25.00
I_2	-7.76	10.43	83.33	3.25	10.54	0.00	-1.50	10.06	83.33	1.00	10.31	0.00
I_3	-8.91	10.36	100.00	2.69	10.44	0.00	-3.28	10.39	100.00	-0.26	10.38	83.33
I_4	-6.08	7.22	100.00	1.54	7.18	0.00	-2.29	7.15	100.00	-0.09	7.23	55.56
I_5	-8.42	8.90	100.00	2.60	8.88	0.00	-3.05	8.88	100.00	-0.14	8.92	50.00
I_6	-10.39	12.07	100.00	3.97	12.11	0.00	-3.43	12.07	92.86	0.02	12.08	35.71
I_7	-8.02	8.37	100.00	2.93	8.38	0.00	-2.68	8.37	93.33	0.12	8.40	40.00
I_8	-8.13	10.24	100.00	2.35	10.35	0.00	-2.98	10.26	100.00	-0.25	10.30	47.06
I_9	-7.22	8.29	100.00	1.70	8.37	0.00	-2.85	8.35	94.74	-0.18	8.35	52.63
I_{10}	-7.50	8.46	100.00	2.98	8.41	0.00	-2.33	8.40	95.24	0.11	8.50	38.10
I_{11}	-8.09	9.13	100.00	2.82	9.17	0.00	-2.78	9.13	100.00	-0.02	9.16	42.31
Avg	-7.98	9.18	98.48	2.69	9.22	0.00	-2.69	9.15	96.32	0.06	9.20	42.70
I_{12}	-7.90	8.35	100.00	2.39	8.36	0.00	-2.89	8.35	100.00	-2.38	8.34	41.38
I_{13}	-8.92	9.56	100.00	2.41	9.59	0.00	-3.40	9.55	97.14	-2.74	9.63	40.00
I_{14}	-14.53	16.89	100.00	6.68	16.96	0.00	-4.52	16.91	100.00	-4.11	16.97	97.22
I_{15}	-7.87	8.87	100.00	2.34	8.92	0.00	-2.89	8.90	96.00	-0.49	8.90	88.00

$[P_0]$: all the $[P]$ instances were solved in less than 4 s (0.36 on average), whereas only 10 $[R]$ instances were solved to optimality within the time limit. For the remaining instances, we obtained solutions with optimality gap of 3.18% on average. In the tables, we reported the values averaged on $I_1 - I_{11}$

only, that is, on instances for which we could find optimal or near-optimal solutions with both models.

The probabilistic and robust solutions exhibit, as expected, a better global behavior. Being unaware of defects occurrence, model $[P_0]$ provides for a quite small overproduction

TABLE 4 Results of worst-case scenario. In bold are the averages over instances $I_1 - I_{11}$.

Inst	[P ₀]			[P ₊]			[P ₋]			[R]		
	<i>E_S</i> (%)	<i>E_W</i> (%)	<i>E_B</i> (%)	<i>E_S</i> (%)	<i>E_W</i> (%)	<i>E_B</i> (%)	<i>E_S</i> (%)	<i>E_W</i> (%)	<i>E_B</i> (%)	<i>E_S</i> (%)	<i>E_W</i> (%)	<i>E_B</i> (%)
<i>I</i> ₁	-7.33	7.64	100.00	2.64	7.64	0.00	-2.47	7.64	100.00	0.13	7.64	50.00
<i>I</i> ₂	-7.93	10.60	83.33	3.05	10.72	0.00	-1.68	10.22	83.33	0.80	10.50	0.00
<i>I</i> ₃	-9.05	10.50	100.00	2.53	10.58	0.00	-3.43	10.53	100.00	-0.41	10.52	83.33
<i>I</i> ₄	-6.15	7.29	100.00	1.47	7.25	0.00	-2.36	7.22	100.00	-0.16	7.29	55.56
<i>I</i> ₅	-8.54	9.02	100.00	2.46	9.01	0.00	-3.18	9.01	100.00	-0.28	9.04	58.33
<i>I</i> ₆	-10.61	12.28	100.00	3.71	12.33	0.00	-3.66	12.28	92.86	-0.23	12.31	42.86
<i>I</i> ₇	-8.13	8.48	100.00	2.79	8.51	0.00	-2.81	8.48	93.33	-0.01	8.52	53.33
<i>I</i> ₈	-8.25	10.36	100.00	2.21	10.47	0.00	-3.11	10.38	100.00	-0.39	10.43	52.94
<i>I</i> ₉	-7.32	8.38	100.00	1.59	8.47	0.00	-2.95	8.44	94.74	-0.29	8.45	57.89
<i>I</i> ₁₀	-7.60	8.56	100.00	2.86	8.52	0.00	-2.44	8.51	95.24	-0.01	8.60	57.14
<i>I</i> ₁₁	-8.21	9.25	100.00	2.69	9.29	0.00	-2.91	9.25	100.00	-0.15	9.28	57.69
Avg	-8.10	9.30	98.48	2.54	9.34	0.00	-2.82	9.27	96.32	-0.09	9.32	51.73
<i>I</i> ₁₂	-8.01	8.47	100.00	2.26	8.47	0.00	-3.01	8.46	100.00	-2.50	8.45	65.52
<i>I</i> ₁₃	-9.06	9.70	100.00	2.25	9.73	0.00	-3.55	9.70	97.14	-2.90	9.77	65.71
<i>I</i> ₁₄	-14.89	17.25	100.00	6.22	17.32	0.00	-4.93	17.27	100.00	-4.52	17.32	100.00
<i>I</i> ₁₅	-7.98	8.98	100.00	2.21	9.04	0.00	-3.01	9.02	96.00	-0.62	9.01	86.00

TABLE 5 Performance of Monte Carlo simulation with NR strategy. In bold are the averages over instances $I_1 - I_{11}$.

Inst	[P ₀]				[P ₊]				[P ₋]				[R]			
	Iter	CPU (s)	Avg. \hat{f} Avg. \hat{f}	Hits (%)	Iter	CPU (s)	Avg. \hat{f} Avg. \hat{f}	Hits (%)	Iter	CPU (s)	Avg. \hat{f} Avg. \hat{f}	Hits (%)	Iter	CPU (s)	\hat{f}	Hits (%)
<i>I</i> ₁	520	3.21	15188	96.03	550	2.84	16823	96.02	1000	5.17	15979	96.02	990	6.94	16405	96.02
<i>I</i> ₂	1000	0.25	442	93.72	1000	0.19	496	93.51	920	0.18	470	93.81	1000	0.28	484	93.68
<i>I</i> ₃	610	6.25	23596	93.96	290	2.67	26629	93.87	860	7.28	25054	93.92	510	6.12	25846	93.94
<i>I</i> ₄	680	3.00	8906	95.76	590	1.92	9628	95.79	680	2.22	9259	95.77	500	2.33	9477	95.76
<i>I</i> ₅	560	6.18	15430	95.24	480	3.73	17286	95.20	1000	7.11	16326	95.22	760	9.57	16825	95.22
<i>I</i> ₆	520	1.59	3959	93.02	740	1.62	4594	92.92	520	1.08	4267	93.02	790	2.63	4417	93.00
<i>I</i> ₇	310	5.07	22166	95.58	970	10.15	24795	95.56	630	7.10	23447	95.57	390	6.85	24133	95.57
<i>I</i> ₈	1000	24.37	23124	93.69	1000	12.84	25792	93.57	730	9.84	24427	93.64	970	22.70	25125	93.64
<i>I</i> ₉	310	3.95	11933	95.26	140	1.08	13092	95.19	290	2.03	12504	95.21	800	9.85	12840	95.20
<i>I</i> ₁₀	130	1.23	9120	95.27	910	4.80	10140	95.20	590	2.95	9622	95.23	230	2.06	9876	95.23
<i>I</i> ₁₁	340	25.80	66498	94.81	520	22.31	74422	94.78	420	16.91	70346	94.80	700	50.21	72347	94.80
Avg	544	7.35	18215	94.76	654	5.83	20336	94.69	695	5.62	19246	94.75	695	10.87	19798	94.73
<i>I</i> ₁₂	320	47.58	102101	95.52	710	59.20	113484	95.51	620	48.92	107640	95.52	290	40.85	108191	95.53
<i>I</i> ₁₃	590	69.92	72443	94.82	420	27.34	81492	94.77	430	26.39	76836	94.80	1000	116.28	77405	94.76
<i>I</i> ₁₄	40	10.48	160569	90.01	540	91.65	200569	89.93	110	16.73	179418	89.95	40	10.96	180293	89.92
<i>I</i> ₁₅	290	79.87	119480	94.99	290	40.27	132800	94.93	110	14.50	125977	94.93	390	135.53	171053	74.65

(0.12% on average) essentially due to the leftovers produced by the last large item of each lots. By contrast, models [P₊], [P₋], and [R] protect the solutions against defect realizations by overproducing, on average, +11.78%, +5.90%, and +8.89%, respectively.

It is interesting to observe that the models behave very similarly in terms of trim loss (precisely, that of [P₋] is on average slightly less than [R]: 1.14% vs. 1.19%). Robustness is however achieved, besides overproduction, by employing different assortments of large items to locally mitigate the

impact of critical defects (see Section 7.3). In detail, in $[R]$ solutions the assignment of small sizes to large ones differs from $[P_0]$ solutions by 15.25% on average, with a peak of 53% in I_7 .

We also observe that $[R]$ almost fully balances the negative and positive net productions of the probabilistic p -median models. The overproduction V_O , in fact, represents a *price of robustness* that is completely paid back by a null mean percent deviation E_S of the expected faultless area versus the total area required. Indeed, the initial overproduction of $[R]$ solutions is fully absorbed by defects realization. Both simulation and worst-case analysis show that, though the expected waste E_W is in practice the same (9.20% vs. 9.18% on average), the net production obtained via $[R]$ is just a little above demand in the simulation (mean $E_S = +0.06\%$) and below demand in the worst case (the mean $E_S = -0.09\%$). Instead, $[P_0]$ solutions end up with a definitely more pronounced underproduction (7.98% on average), a value that is consistent with the company estimate of losses due to imperfections.

A direct consequence, very detrimental for production costs, is that in all instances but I_2 model $[P_0]$ does not allow the completion of any order batch ($E_B = 98.48\%$ on average and 7.71% mean underproduction per order). On the other hand, the mean percent of back-orders recorded by robust solutions is reduced to 42.7%. This value is still large but in this case the mean per-order underproduction is as small as 0.32%, a value that greatly helps keep the whole plant operation smooth. In fact, the expected losses \bar{a}_j used in the objective functions of the probabilistic p -median models are mean values not able to precisely capture the realization of defects. On average, the net production obtained via $[P_+]$ (resp., $[P_-]$) increased (resp., decreased) by 2.69%.

About the Monte Carlo simulation, Table 5 shows, for each model, the number of iterations of the simulation, the CPU time, the average number \hat{f} of sampled faults per repetition, and the average percentage of critical defects. Few seconds CPU time were generally spent to simulate the solutions of instances I_1 – I_{11} . The largest deviation from this behavior is the 50.21 seconds recorded to simulate the $[R]$ solution of instance I_{11} . The mean CPU time ranges between 5.62 ($[P_-]$) and 10.87 s ($[R]$). The average \hat{f} is larger for the models ($[P_+]$ and $[R]$) that embed a larger estimation of faults, reaching 74,422 defects per iteration. The hit rate is similar for all models, about 94.7% on average.

We finally observe that the sample means are almost always slightly lower than the corresponding analytical values (with a maximum difference of 0.012%). However, the values of all the analytical means fall in the 95% confidence intervals of the statistical parameters obtained by Monte Carlo simulation. The small bias is probably due to the fact that, for the sake of numerical precision, we approximated $\varphi^t = 0$ for $t > 10$, that is, a large item with more than 10 defects is regarded as an event with negligible chances.

7.5 | Results for CR strategy

Tables 6, 7, and 8 give the results obtained with CR strategy. Table 6 has no $[P_0]$ section since it is the same as Table 2: in fact, $[P_0]$ solutions are not based on probabilities and therefore are identical. The other models behave very similarly with CR and NR strategy, with a slight general reduction, in the former case, of both trim loss and overproduction. With model $[R]$, trim loss reduces to 1.18% (vs. 1.19%) and overproduction to 8.53% (vs. 8.89%).

If one reads the outcome of the Monte Carlo simulation, $[P_0]$ turns out to produce the same percentage of back-orders as with NR . On the other hand, the mean percent underproduction and the mean total waste E_W decrease: the former from 7.98% to 7.68%, the latter from 9.18% to 8.88%.

We observed similar variations in $[R]$, $[P_+]$, and $[P_-]$ solutions. In particular, the mean total waste E_W of $[R]$ solutions decrease by 0.32%. Instead, unlike $[P_0]$ solutions, the mean percent back-orders E_B of the $[R]$ solutions obtained with CR decreases by some 1%.

Overall, the implementation of CR strategy keeps the percent deviation E_S at same level as NR strategy, but can reduce total waste by 3.5% (from 9.20% to 8.88%), indeed a nonnegligible improvement given the high production volume of the plant, and also considering that E_W is mainly lowered because faulty area is in turn reduced by 4% on average. However, the CR strategy may require additional cuts and/or cutter set-ups that can considerably increase processing time, thus possibly cancelling the benefits of a better response to defects. An assessment from the plant management is therefore required in this regard.

A final note on the performance of the simulations: Table 9 shows the details of Monte Carlo simulation for CR strategy. Our considerations are similar to those done for Table 5, but some differences appear when comparing simulation performance of CR and NR . With the former strategy, the CPU time required for instances I_1 – I_{11} is generally larger than with the latter, reaching 89.19 s with model $[P_+]$ in instance I_{11} . On the whole, the mean simulation time with CR strategy is 2.42 times that with NR . This growth can be explained by a +7.83% iterations to converge, and also by the enumeration required to find the best pattern reconfiguration. The faults sampled on the whole in CR and NR are comparable in number, with a slight reduction (−0.3%) for CR ; but the hit ratio lowered on average by 3.49%, resulting in a mean 4% reduction of defective area. This is due to the benefits of this reconfiguration policy. Considering the potential benefit of CR w.r.t. NR , the hit ratio reduction could seem modest. It should be noted, however, that hit ratios refer to solutions employing only maximal cutting patterns. These, on one hand, ensure very small trim loss but, on the other hand, offer little room for defect avoidance operation.

TABLE 6 Solutions with *constrained reconfiguration* strategy. In bold are the averages over instances $I_1 - I_{11}$.

Inst	$[P_+]$			$[P_-]$			$[R]$			
	CPU (s)	V_O (%)	V_T (%)	CPU (s)	V_O (%)	V_T (%)	CPU (s)	gap (%)	V_O (%)	V_T (%)
I_1	0.05	10.62	0.31	0.08	5.19	0.31	0.26	0	7.94	0.31
I_2	0.03	12.32	2.10	0.03	6.98	1.66	1.09	0	9.73	1.93
I_3	0.05	11.88	1.57	0.14	5.73	1.56	0.90	0	9.06	1.55
I_4	0.09	7.85	1.08	0.08	3.89	1.08	12.06	0	6.19	1.15
I_5	0.09	11.85	0.46	0.09	5.79	0.47	49.49	0	8.96	0.50
I_6	0.08	15.26	1.72	0.17	7.47	1.64	99.02	0	11.32	1.68
I_7	0.20	11.79	0.34	0.17	5.75	0.33	-	0.13	8.78	0.35
I_8	0.20	10.53	2.27	0.20	5.07	2.21	809.10	0	8.07	2.25
I_9	0.31	9.25	1.10	0.31	4.56	1.13	-	1.06	7.35	1.13
I_{10}	0.48	10.83	0.85	0.55	5.42	0.83	-	2.01	8.00	0.97
I_{11}	0.27	11.36	1.10	0.36	5.52	1.09	-	0.34	8.45	1.11
Avg	0.17	11.23	1.17	0.20	5.58	1.12	97.20	0.32	8.53	1.18
I_{12}	0.83	10.95	0.46	1.00	5.34	0.46	-	2.28	5.79	0.49
I_{13}	1.27	12.19	0.65	1.27	5.94	0.64	-	2.52	6.68	0.68
I_{14}	0.88	22.83	2.96	1.30	10.90	2.68	-	5.84	11.44	2.76
I_{15}	2.39	10.61	1.07	2.44	5.17	1.08	-	11.38	7.69	5.21

TABLE 7 Results of Monte Carlo simulation

Inst	$[P_0]$			$[P_+]$			$[P_-]$			$[R]$		
	E_S (%)	E_W (%)	E_B (%)	E_S (%)	E_W (%)	E_B (%)	E_S (%)	E_W (%)	E_B (%)	E_S (%)	E_W (%)	E_B (%)
I_1	-7.14	7.44	100.00	2.71	7.44	0.00	-2.34	7.44	75.00	0.22	7.45	50.00
I_2	-7.37	10.05	83.33	3.14	10.10	0.00	-1.67	9.62	83.33	0.82	9.90	0.00
I_3	-8.47	9.93	100.00	2.37	9.93	0.00	-3.25	9.92	100.00	-0.20	9.91	83.33
I_4	-5.85	6.99	100.00	1.47	6.93	0.00	-2.25	6.93	100.00	-0.09	7.00	44.44
I_5	-8.30	8.78	100.00	2.52	8.76	0.00	-3.02	8.76	100.00	-0.13	8.80	41.67
I_6	-9.91	11.59	100.00	3.65	11.62	0.00	-3.33	11.53	100.00	0.10	11.58	35.71
I_7	-7.91	8.26	100.00	2.89	8.27	0.00	-2.66	8.26	100.00	0.12	8.29	26.67
I_8	-7.59	9.71	100.00	2.09	9.73	0.00	-2.94	9.67	100.00	-0.18	9.72	47.06
I_9	-6.95	8.02	100.00	1.56	8.06	0.00	-2.77	8.06	84.21	-0.20	8.08	52.63
I_{10}	-7.19	8.15	100.00	2.76	8.07	0.00	-2.24	8.03	90.48	0.14	8.18	28.57
I_{11}	-7.76	8.80	100.00	2.66	8.82	0.00	-2.71	8.80	100.00	-0.02	8.83	46.15
Avg	-7.68	8.88	98.48	2.53	8.89	0.00	-2.65	8.82	93.91	0.06	8.88	41.48
I_{12}	-7.74	8.20	100.00	2.32	8.20	0.00	-2.84	8.19	100.00	-2.38	8.18	44.83
I_{13}	-8.74	9.38	100.00	2.34	9.37	0.00	-3.35	9.35	100.00	-2.73	9.44	45.71
I_{14}	-13.80	16.18	100.00	5.99	16.26	0.00	-4.45	16.14	100.00	-3.95	16.19	94.44
I_{15}	-7.56	8.56	98.00	2.21	8.58	0.00	-2.80	8.58	94.00	0.35	11.67	38.00

8 | CONCLUSIONS AND FUTURE RESEARCH

We addressed a stock assortment-and-cutting problem with the goal of constructing optimal solutions that are robust

against imperfections of raw material. The problem and the computational results presented here refer to a real application in the glass industry. Defect occurrence was modeled as a Poisson point process. On that basis, we developed approaches that protect solutions against the worst

TABLE 8 Results of worst-case scenario

Inst	[P ₀]			[P ₊]			[P ₋]			[R]		
	E _S (%)	E _W (%)	E _B (%)	E _S (%)	E _W (%)	E _B (%)	E _S (%)	E _W (%)	E _B (%)	E _S (%)	E _W (%)	E _B (%)
I ₁	-7.23	7.53	100.00	2.61	7.53	0.00	-2.43	7.53	75.00	0.13	7.53	50.00
I ₂	-7.51	10.20	83.33	2.96	10.25	0.00	-1.85	9.77	83.33	0.64	10.06	16.67
I ₃	-8.60	10.06	100.00	2.23	10.06	0.00	-3.39	10.05	100.00	-0.35	10.04	83.33
I ₄	-5.91	7.05	100.00	1.40	6.99	0.00	-2.31	6.99	100.00	-0.15	7.05	44.44
I ₅	-8.42	8.90	100.00	2.38	8.89	0.00	-3.15	8.88	100.00	-0.26	8.92	50.00
I ₆	-10.11	11.79	100.00	3.43	11.81	0.00	-3.54	11.72	100.00	-0.12	11.78	42.86
I ₇	-8.03	8.38	100.00	2.76	8.38	0.00	-2.78	8.38	100.00	-0.01	8.40	46.67
I ₈	-7.75	9.87	100.00	1.91	9.89	0.00	-3.11	9.82	100.00	-0.35	9.87	52.94
I ₉	-7.05	8.12	100.00	1.46	8.15	0.00	-2.87	8.15	84.21	-0.29	8.17	52.63
I ₁₀	-7.28	8.24	100.00	2.66	8.17	0.00	-2.34	8.13	90.48	0.03	8.27	42.86
I ₁₁	-7.86	8.91	100.00	2.55	8.93	0.00	-2.82	8.90	100.00	-0.13	8.93	69.23
Avg	-7.80	9.00	98.48	2.40	9.01	0.00	-2.78	8.94	93.91	-0.07	9.00	50.15
I ₁₂	-7.85	8.31	100.00	2.20	8.31	0.00	-2.95	8.30	100.00	-2.49	8.28	58.62
I ₁₃	-8.87	9.51	100.00	2.19	9.50	0.00	-3.49	9.49	100.00	-2.87	9.57	71.43
I ₁₄	-14.15	16.53	100.00	5.57	16.59	0.00	-4.84	16.48	100.00	-4.34	16.53	100.00
I ₁₅	-7.65	8.65	100.00	2.10	8.68	2.00	-2.91	8.67	96.00	2.35	9.91	48.00

TABLE 9 Performance of Monte Carlo simulation with CR strategy

Inst	[P ₀]				[P ₊]				[P ₋]				[R]			
	Iter	CPU (s)	Avg. \hat{f} (%)	Hits (%)	Iter	CPU (s)	Avg. \hat{f} (%)	Hits (%)	Iter	CPU (s)	Avg. \hat{f} (%)	Hits (%)	Iter	CPU (s)	Avg. \hat{f} (%)	Hits (%)
I ₁	850	18.69	15181	94.78	790	15.90	16792	94.82	930	16.82	15968	94.80	290	6.24	16395	94.80
I ₂	470	0.37	444	89.13	1000	0.58	493	89.20	820	0.46	467	89.09	1000	0.69	480	89.13
I ₃	820	23.42	23589	90.15	540	14.72	26393	90.11	450	12.11	24944	90.15	1000	30.08	25716	90.22
I ₄	730	9.51	8905	91.62	700	9.69	9591	91.75	520	5.62	9243	91.73	510	6.24	9454	91.70
I ₅	850	33.21	15423	93.69	880	18.29	17242	93.68	780	15.56	16309	93.70	480	11.72	16811	93.70
I ₆	1000	7.69	3956	89.24	550	2.65	4556	89.07	990	4.53	4242	89.08	810	4.86	4396	89.26
I ₇	860	33.67	22153	94.57	520	15.71	24765	94.53	870	24.78	23417	94.60	520	18.62	24101	94.47
I ₈	710	38.65	23126	88.93	680	22.62	25556	88.84	850	27.46	24274	88.94	1000	41.78	24980	88.96
I ₉	840	21.16	11925	91.83	390	7.39	13029	91.81	450	8.46	12475	91.70	290	7.36	12808	91.85
I ₁₀	760	13.86	9114	91.89	650	9.11	10085	91.85	1000	14.53	9586	91.91	1000	17.74	9834	91.82
I ₁₁	150	20.18	66494	91.15	870	89.19	74008	91.13	110	10.79	70130	91.16	590	78.49	72093	91.18
Avg	731	20.04	18210	91.54	688	18.71	20228	91.53	706	12.83	19187	91.53	681	20.35	19734	91.55
I ₁₂	420	103.31	102101	93.53	140	25.89	113234	93.54	520	89.22	107513	93.55	520	127.84	108002	93.51
I ₁₃	850	159.63	72425	93.02	650	85.56	81227	92.90	120	15.32	76706	92.95	290	53.65	77268	92.95
I ₁₄	410	142.69	160586	85.65	280	75.70	197625	84.90	510	125.99	177901	85.79	420	168.32	178940	85.59
I ₁₅	230	81.79	119487	91.03	470	118.84	132129	90.84	90	21.84	125621	90.82	420	200.68	167363	34.25

possible distribution of f defects over the glass sheets to be cut.

Depending on the recourse strategy used, a defect on a glass sheet may or may not be avoided. With the simplest reaction considered in the present case, that is, NR ,

the recourse simply consists in doing nothing. The small items hit are just discarded with no attempt to reconfigure the patterns. In more general and effective (but also complex) reactions, patterns can be reconfigured by moving the strips of scrap (CR), or even by splitting them with

additional cuts (UR) in order to let defects fall in the trim loss.

Within each glass sheet, we then evaluate the expected loss of produced items via the conditioned probability of finding a defect unrecoverable with the recourse strategy adopted. Closed forms for the computation of such probabilities were provided in simple cases, and obtained via Monte Carlo simulation in more complex ones.

Starting from a deterministic p -median model that minimizes the total glass area used, two alternative approaches were presented. In the first one (models $[P_+]$ and $[P_-]$), the expected loss by defects is just plugged into the objective function of the p -median model, and in the second one (model $[R]$), the robust counterpart of the p -median model is constructed in the classical vein of robust optimization. Computational tests were performed on real data taken from the application. The tests show that:

- both approaches mitigate the pronounced net production deficit and back-order volumes found by deterministic solutions, while keeping trim loss pretty much at the same (low) level.
- Nonetheless, $[P_+]$ and $[P_-]$ seem to not completely hit the target. The former nullifies back-orders at the cost of a non-negligible extra production (which is regarded as waste). The latter substantially reduces net production deficit, but leaves back-order percentages almost untouched.
- Conversely, the robust model $[R]$ halves the amount of back-orders with a net production that deviates from requirements by only 0.06%. The price of robustness, consisting in the initial overproduction, is then completely absorbed.
- Finally, a redesigned easy-to-implement recourse strategy, CR , achieves a nonnegligible 3.5% reduction of total waste, mainly attributed to a mean 4% reduction of the defective area. This is a valuable figure in case defective items are not treated as waste but as lower value products.

In our opinion, the results open interesting perspectives and research challenges in the field of stock cutting. In fact, despite the recognized importance of defect handling in industrial applications, very few studies can be found on this subject. To the best of our knowledge, none of them address the problem of finding robust solutions. Among the questions left open we can list the following:

- How should robustness against defects be dealt with in standard cutting stock, or in general one- and two-dimensional knapsack/bin packing problems? Clearly, the possibility of solving model $[R]$ algorithmically (by, e.g., column generation and/or branch-and-price) is both interesting and relevant.
- It is not clear whether a stochastic programming formulation would lead to a better performance in both computational and economical terms. This topic is worth a separate investigation and left as a potential future project.

- What is the complexity of fault recovery with UR recourse strategy or more complex ones?

Finally, in our case study, trim loss and defective items are not distinguished but altogether treated as waste. This is not the case for industrial production in general. For example, some defect types are tolerated in wood-board cutting and the items affected are not treated as waste but as lower value production. In such cases, it would be interesting to understand how robust solutions would change, and in particular whether they would or would not make a larger use of trim loss as a means to increase protection against value loss.

ACKNOWLEDGMENTS

We gratefully acknowledge the technical support of the R&D department of Nippon Sheet Glass Co. Ltd., with special thanks to Dr. Eng. Alessandro Consorte and Marcello Romano for the help given in the analysis of defect generation. We are also very grateful to the AE and the anonymous reviewers for the in-depth reading and stimulating comments, which helped us improve the quality of this contribution. Finally, we would like to thank Prof. Simon Wigley for the kind editorial support.

ORCID

Fabrizio Marinelli  <https://orcid.org/0000-0003-0405-3110>

REFERENCES

- Aboudi, R., & Barcia, P. (1998). Determining cutting stock patterns when defects are present. *Annals of Operational Research*, 82, 343–354.
- Afsharian, M., Niknejad, A., & Wäscher, G. (2014). A heuristic, dynamic programming-based approach for a two-dimensional cutting problem with defects. *OR Spectrum*, 36(4), 971–999.
- Alem, D. J., & Morabito, R. (2012). Production planning in furniture settings via robust optimization. *Computers & Operations Research*, 39, 139–150.
- Alem, D. J., Munari, P. A., Arenales, M. N., & Ferreira, P. A. V. (2010). On the cutting stock problem under stochastic demand. *Annals of Operations Research*, 179, 169–186.
- Arbib, C., & Marinelli, F. (2007). An optimization model for trim loss minimization in an automotive glass plant. *European Journal of Operational Research*, 183(3), 1421–1432.
- Arbib, C., & Marinelli, F. (2009). Exact and asymptotically exact solutions for a class of assortment problems. *INFORMS Journal on Computing*, 21(1), 13–25.
- Ben-Tal, A., El Ghaoui, L., & Nemirovski, A. (2009). *Robust optimization*. Princeton University Press.
- Beraldi, P., Bruni, M. E., & Conforti, D. (2009). The stochastic trim-loss problem. *European Journal of Operational Research*, 197(1), 42–49.
- Bertsimas, D., & Brown, D. B. (2009). Constructing uncertainty sets for robust linear optimization. *Operations Research*, 57(6), 1483–1495.
- Bertsimas, D., & Sim, M. (2004). The price of robustness. *Operations Research*, 52(1), 35–53.
- Birge, J., & Louveaux, F. (2011). *Introduction to stochastic programming*. Springer.
- Daley, D. J., & Vere-Jones, D. (1988). *An introduction to the theory of point processes*. Springer.
- Durak, B., & Tüzün, D. A. (2017). Dynamic programming and mixed integer programming based algorithms for the online glass cutting problem with defects and production targets. *International Journal of Production Research*, 55(24), 7398–7411.

- Carnieri, C., Mendoza, G. A., & Luppold, W. G. (1993). Optimal cutting of dimension parts from lumber with a defect: A heuristic solution procedure. *Forest Products Journal*, 43(9), 66–72.
- Delorme, M., Iori, M., & Martello, S. (2016). Bin packing and cutting stock problems: Mathematical models and exact algorithms. *European Journal of Operational Research*, 255(1), 1–20.
- Gilmore, P. C., & Gomory, R. E. (1961). A linear programming approach to the cutting-stock problem. *Operations Research*, 9, 849–859.
- Ghods, R., & Sassani, F. (2005). Real-time optimum sequencing of wood cutting process. *International Journal of Production Research*, 43(6), 1127–1141.
- Hahn, S. G. (1968). On the optimal cutting of defective sheets. *Operations Research*, 16(6), 1100–1114.
- Ide, J., Tiedemann, M., Westphal, S., & Haiduk, F. (2015). An application of deterministic and robust optimization in the wood cutting industry. *4OR*, 13, 35–57.
- Krichagina, E. V., Rubio, R., Taksar, M. I., & Wein, L. M. (1998). A dynamic stochastic stock-cutting problem. *Operations Research*, 46(5), 690–701.
- McCormick, G. P. (1976). Computability of global solutions to factorable nonconvex programs: Part I—Convex underestimating problems. *Mathematical Programming*, 10, 147–175.
- Neidlein, V., Vianna, A. C. G., Arenales, M. N., & Wäscher, G. (2009). Two-dimensional guillotineable-layout cutting problems with a single defect—An AND/OR-graph approach. In B. Fleischmann, K-H. Borgwardt, R. Klein, & A. Tuma (Eds.), *Operations research proceedings 2008 Part 3* (pp. 85–90). Springer.
- Özdamar, L. (2000). The cutting-wrapping problem in the textile industry: Optimal overlap of fabric lengths and defects for maximizing return based on quality. *International Journal of Production Research*, 38, 1287–1309.
- Rönnqvist, M. (1995). A methods for the cutting stock problem with different qualities. *European Journal of Operational Research*, 83, 57–68.
- Rönnqvist, M., & Åstrand, E. (1998). Integrated defect detection and optimization for cross cutting of wooden boards. *European Journal of Operational Research*, 108, 490–508.
- Sarker, B. R. (1988). An optimum solution for one-dimensional slitting problems: A dynamic programming approach. *Journal of the Operational Research Society*, 39, 749–755.
- Schepler, X., Rossi, A., Gurevsky, E., & Dolgui, A. (2022). Solving robust bin-packing problems with a branch-and-price approach. *European Journal of Operational Research*, 297(3), 831–843.
- Sculli, D. (1981). A stochastic cutting stock procedure: Cutting rolls of insulating tape. *Management Science*, 27(8), 946–952.
- Shapiro, A., Dentcheva, D., & Ruszczyński, A. (2009). Lectures on stochastic programming: Modelling and theory. *MPS-SIAM series on optimization*. SIAM, Philadelphia.
- Soyster, A. L. (1973). Convex programming with set-inclusive constraints and applications to inexact linear programming. *Operations Research*, 21, 1154–1157.
- Wäscher, G., Haußner, H., & Schumann, H. (2007). An improved typology of cutting and packing problems. *European Journal of Operational Research*, 183(3), 1109–1130.
- Wenshu, L., Dan, M., & Jinzhuo, W. (2015). Study on cutting stock optimization for decayed wood board based on genetic algorithm. *The Open Automation Control Systems Journal*, 7, 284–289.

SUPPORTING INFORMATION

Additional supporting information can be found online in the Supporting Information section at the end of this article.

How to cite this article: Arbib, C., Marinelli, F., Pinar, M., & Pizzuti, A. (2022). Robust stock assortment and cutting under defects in automotive glass production. *Production and Operations Management*, 31, 4154–4172.
<https://doi.org/10.1111/poms.13812>