Two-Stage Adaptable Robust Optimization for Glass Production

Anton Medvedev[2][a], Safia Kedad-Sidhoum[1][b] and Frédéric Meunier[3][c]

[1] *CEDRIC - CNAM, France*

[2] *CEDRIC - CNAM & SGR, France*

[3] *CERMICS - École des Ponts ParisTech, France*

fi

Keywords: Robust Optimization, Finite Adaptability, Two-Stage Optimization, Glass Production.

Abstract: In the glass industry, visual and thermal properties of the glass sheets are obtained via the deposit of thin layers of different materials. A standard way to perform this step is the use of a "magnetron," in which the materials are transferred from cathodes to the sheets using a magnetic field. Since the cathodes are very expensive, activation and replacement decisions have to be carefully decided to keep the cost of the wasted materials low. The production is organized in campaigns and the activation and replacement decisions of the cathodes have to be taken before each campaign. Yet, the exact orders to process during a campaign are only revealed after the decisions have been taken. We focus here on the case of two campaigns, which we model as a two-stage robust optimization problem. We propose a method based on the finite adaptability approach of Bertsimas and Caramanis (2010) combined with the branch-and-bound of Subramanyam et al. (2020). Experiments on real instances show that our method leads to clear diminutions of the cost of wasted material in the worst cases, and—even more interesting—allow to find solutions for cases that are unfeasible with the heuristic used by the practitioners.

1 INTRODUCTION

Within the glass industry, the visual and thermal properties of glass sheets are achieved by applying thin layers of different materials. A common method for accomplishing this is the utilization of a " magnetron," which employs a magnetic field to transfer materials from cathodes to the glass sheets. As cathodes are a costly component, it is crucial to carefully determine how to refill, to replace, and to activate them in order to minimize the wasted material costs. Due to possible changes of the initial production plan, the consumption of cathodes is somewhat uncertain, making the task of finding the best refill, replacement, and activation decisions highly challenging. In this procedure, production is structured into campaigns that include numerous orders. During the intervals between these campaigns, maintenance for the magnetron is carried out and decisions regarding refilling, replacing, and activating the equipment have to be taken.

[a] https://orcid.org/0009-0003-7726-7168
[b] https://orcid.org/0000-0002-2184-2261
[c] https://orcid.org/0000-0002-5331-2863

This magnetron problem is encountered by Saint-Gobain, a French multinational company producing a variety of construction high-performance materials such as coated glass. The magnetron has already been the subject of an academic work, by Gicquel et al. (2010). But the focus of this latter paper is different: while we assume in the present work that the location of the cathodes is fixed, the cited paper addresses the problem of finding the optimal locations of the cathodes in the magnetron.

Uncertainty of the parameters may have crucial impact on the feasibility and optimality of an optimization problem (Ben-Tal and Nemirovski, 2000). The two main viewpoints on uncertainty in optimization are the stochastic viewpoint and the robust one. The assumptions and goals of these two viewpoints are different. Stochastic optimization relies on the probability distribution of the uncertain parameters and in general aims at optimizing the expectation of an objective function. Ruszczyński and Shapiro (2003) provide details on stochastic optimization. Feasibility of all uncertainty realizations is not necessary required. On the other hand, robust optimization relies on a set-based uncertainty model, the solutions are required to be feasible for all uncertainty realiza-

Medvedev, A., Kedad-Sidhoum, S. and Meunier, F.
Two-Stage Adaptable Robust Optimization for Glass Production.
DOI: 10.5220/0012316900003639

In *Proceedings of the 13th International Conference on Operations Research and Enterprise Systems (ICORES 2024)*, pages 229-235
ISBN: 978-989-758-681-1; ISSN: 2184-4372

tions, and in general the aim is to optimize the worst case. Gabrel et al. (2014) provide a comprehensive review of developments of robust optimization and its application areas.

In this paper, we address the uncertainty of the magnetron problem from the robust viewpoint. According to the engineers operating the magnetron, feasibility must be ensured in all cases. Indeed, a lack of material on a cathode might lead to a very costly interruption of the campaign, and, moreover, exposing such a cathode with the magnetic field can even lead to serious damages to the magnetron itself. This already brings the problem in the realm of robust optimization. A second reason comes from the main objective function behind the decisions of the engineers: not to have to perform too many tasks on the cathodes between the campaigns. Once a reasonable number of tasks ensuring feasibility has been identified, the cost of wasted material has to be kept acceptable, even in the worst scenario.

We focus on the two-campaign case, which we model as a "two-stage" robust optimization problem. In such a problem, there is a first series of decisions to be taken (the "here-and-now" decisions), then uncertainty is revealed, and then there is a second series of decisions to be taken (the "wait-and-see" recourse decisions). It is worth noting that even though in the magnetron problem the uncertain data is revealed in two steps—with recourse decisions between these two steps—we manage to get a two-stage problem, with only one step of uncertainty to take into account.

We propose then an efficient method to solve this problem. We follow the finite adaptability approach by Bertsimas and Caramanis (2010), combined with the exact branch-and-bound by Subramanyam et al. (2020). Experiments show that our method outperforms the standard one by two aspects: the cost of wasted material is always smaller in our solutions than in the ones computed with the heuristic used by the practitioners; we show that there exists situations where this latter method concludes to unfeasibility, while our method does find solutions.

In Section 2, a formal description of the problem is given. This description is done in two steps: first, the deterministic version, where all parameters are known when the decisions have to be taken: second, the robust version, where the uncertain parameters are progressively revealed, and where some decisions depend on the uncertainty. This problem is then modeled as a mathematical program in Section 3. Section 4 presents the method proposed to solve the problem after a brief introduction to finite adaptability and to the branch-and-bound approach. In Section 5, numerical results are provided and commented, proving the efficiency of the method.

2 PROBLEM FORMULATION

In this section we present the formulation of the magnetron problem in its two versions, deterministic and robust.

2.1 Deterministic Version

On the magnetron, there are n possible locations for the cathodes. A *campaign* C is a multiset of orders. An order o corresponds to a triple (G_o, p_o, t_o), with $G_o \subseteq [n]$ and $p_o, t_o \in \mathbb{R}_+$. The number p_o is the total power that must be distributed among the cathodes at locations in G_o. The number t_o is the processing time of the order o.

Moreover, the cathodes' locations are partitioned into m subsets denoted by $A_1, \ldots, A_m$. Each of these subsets corresponds to locations assigned to a certain material.

As we focus on a two-campaign setting, an instance of the problem is a pair of campaigns C^1, C^2. The orders in C^1 are processed before those in C^2.

The following decisions have to be taken for each campaign C^c:

- *Refill:* choose a subset $R^c \subseteq [n]$ of locations for which the cathodes will be refilled at the full level just before C^c. The full level at location j is denoted by Q_j.
- *Replacement:* choose a permutation σ^c of $[n]$ such that $\sigma^c(A_i) = A_i$ for all $i \in [m]$: for $j \in A_i$, the cathode at j will be placed at $\sigma^c(j) \in A_i$ just before the refills R^c.
- *Activation:* distribute the power p_o of order $o \in C^c$ among the cathodes at locations in G_o, as $p_o = \sum_{j \in G^o} p^c_{o,j}$.

The number of cathode's refills before C^c cannot exceed a given limit $\overline{r}^c$. The number of replacements between cathodes before C^c cannot exceed a given threshold $\overline{s}^c$. Given $o \in C^c$, the quantities $p^c_{o,j}$ of an activation decision $p_o = \sum_{j \in G_o} p^c_{o,j}$ have to belong to $\{0\} \cup [\underline{p}_j, \overline{p}_j]$, where $\underline{p}_j$ and $\overline{p}_j$ are parameters. In words, this last constraint means that when a cathode is activated, the power of the activation has to lie in a given interval.

For the cathode located at j, we denote by q^1_j the initial quantity of material, by q^2_j the quantity of material just after C^1, and by q^3_j the quantity of material just after C^2. While q^1_j is a parameter, the quantities

q_j^2 and q_j^3 are auxiliary variables. For $c \in \{1,2\}$, the quantity q_j^{c+1} can be expressed as

$$q_j^{c+1} = -\sum_{o \in C^c} t_o p_{o,j}^c + \begin{cases} Q_j & \text{if } j \in R^c. \\ q_{\sigma^c(j)}^c & \text{otherwise.} \end{cases}$$

The first term in the expression of q_j^{c+1} stands for the material consumption implied by campaign C^c and activation decisions on the cathode at j. For $c \in \{2,3\}$ the quantity q_j^c has to be non-negative: $q_j^c \geqslant 0$.

The only cost implied by the refill, the replacement, and the activation decisions is the materials waste cost. For a cathode at location j just before C^c, the latter is $c_j q_j^c$, where c_j is the unit cost of the material located at j. In words, this means that when a cathode at j is refilled, all the current quantity on it is lost. With the notation $F(q^c, R^c) := \sum_{j \in R^c} c_j q_j^c$, the total resulting cost can be written as $F(q^1, R^1) + F(q^2, R^2)$.

2.2 Robust Version

We present now the robust version to handle the uncertain parameters.

For C^c there is a polyhedral uncertainty set Ω^c. The only parameters that depend on the uncertainty are the production times t_o, which we write from now on $t_o(\omega^c)$, where $\omega^c \in \Omega^c$ if $o \in C^c$. We assume that this dependence is affine.

The uncertainty realizations $\omega^1 \in \Omega^1$ and $\omega^2 \in \Omega^2$ are revealed in two steps: ω^1 is revealed after C^1 has been processed, and ω^2 is revealed after C^2 has been processed. This implies that the decisions taken between C^1 and C^2 may depend on ω^1. In particular, the refill decisions of C^2 becomes $R^2(\omega^1)$. No decisions are taken after C^2 has been processed. Yet some auxiliary variables depend on ω^2.

The last changes the robust version implies with respect to the deterministic version are the expression of q_j^2 and q_j^3, the non-negativity of the available quantity of material on the cathodes, and the expression of the objective function.

For all $j \in [n]$ the quantity q_j^2 becomes

$$q_j^2(\omega^1) = -\sum_{o \in C^1} t_o(\omega^1) p_{o,j}^c + \begin{cases} Q_j & \text{if } j \in R^1, \\ q_{\sigma^1(j)}^1 & \text{otherwise,} \end{cases}$$

and the quantity q_j^3 becomes

$$q_j^3(\omega^1, \omega^2) = -\sum_{o \in C^2} t_o(\omega^2) p_{o,j}^c(\omega^1) + \begin{cases} Q_j & \text{if } j \in R^2(\omega^1), \\ q_{\sigma^2(\omega^1)(j)}^2(\omega^1) & \text{otherwise,} \end{cases}$$

for all $\omega^1 \in \Omega^1, \omega^2 \in \Omega^2$.

The non-negativity constraints translate into $q_j^2(\omega^1) \geqslant 0$ and $q_j^3(\omega^1, \omega^2) \geqslant 0$ for all $\omega^1 \in \Omega^1, \omega^2 \in \Omega^2$.

The objective function becomes $F(q^1, R^1) + \max_{\omega^1 \in \Omega^1} F(q^2(\omega^1), R^2(\omega^1))$. Robustness refers to the requirement to be feasible for every realization of the uncertainty and to be minimal for the maximal possible cost.

3 MODELING

In this section we model the magnetron problem in its robust version as a mathematical program.

The constraints related to the number of refills are:

$$\begin{aligned} &\sum_{j=1}^{n} r_j^1 \leqslant \bar{r}^1 \\ &\sum_{j=1}^{n} r_j^2(\omega^1) \leqslant \bar{r}^2 \quad \forall \omega^1 \in \Omega^1 \end{aligned} \tag{1}$$

The variable r_j^1 is binary and takes the value 1 if the cathode located at j is refilled, i.e., if $j \in R^1$. Similarly, the variable $r_j^2(\omega^1)$ is binary and takes the value 1 if the cathode located at j is refilled in the realization ω^1, i.e., if $j \in R^2(\omega^1)$.

The constraints related to the number of replacements are:

$$\begin{aligned} &\sum_{j=1}^{n} \sum_{\substack{k=1 \\ k \neq j}}^{n} y_{j,k}^1 \leqslant \bar{s}^1 \\ &\sum_{\substack{k=1 \\ j \neq k}}^{n} y_{k,j}^1 = \sum_{\substack{k=1 \\ j \neq k}}^{n} y_{j,k}^1 && \forall j \in [n] \\ &\sum_{k=1}^{n} y_{j,k}^1 = 1 && \forall j \in [n] \\ &\sum_{j=1}^{n} \sum_{\substack{k=1 \\ k \neq j}}^{n} y_{j,k}^2(\omega^1) \leqslant \bar{s}^2 && \forall \omega^1 \in \Omega^1 \\ &\sum_{\substack{k=1 \\ j \neq k}}^{n} y_{k,j}^2(\omega^1) = \sum_{\substack{k=1 \\ j \neq k}}^{n} y_{j,k}^2(\omega^1) && \forall j \in [n], \omega^1 \in \Omega^1 \\ &\sum_{k=1}^{n} y_{j,k}^2(\omega^1) = 1 && \forall j \in [n], \omega^1 \in \Omega^1 . \end{aligned} \tag{2}$$

The variable $y_{j,k}^1$ is binary and takes the value 1 if the cathode at j is replaced by the cathode at k, i.e., if $\sigma^1(j) = k$. Similarly, the variable $y_{j,k}^2(\omega^1)$ is binary and takes the value 1 if the cathode at j is replaced by the cathode at k in the realization ω^1, i.e., if $\sigma^2(\omega^1)(j) = k$. Constraints (2) models the permutation in a standard way, as done for instance for the directed TSP; see, e.g., Korte et al. (2011).

The constraints on the activation decisions are:

$$\begin{aligned} &\sum_{j\in G_o} p^1_{o,j} = p_o && \forall o \in C^1 \\ &\sum_{j\in G_o} p^2_{o,j}(\omega^1) = p_o && \forall o \in C^2, \omega^1 \in \Omega^1 . \end{aligned} \tag{3}$$

The constraints on the quantities of material on cathodes at any location before both campaigns are:

$$\begin{aligned} q^2_j(\omega^1) = \; & -\sum_{o\in C^1} t_o(\omega^1) p^1_{o,j} + r^1_j Q_j \\ & +(1-r^1_j)\sum_{k=1}^{n} y^1_{j,k} q^1_k \\ & \qquad \forall j \in [n], \omega^1 \in \Omega^1 \\ q^3_j(\omega^1,\omega^2) = \; & -\sum_{o\in C^2} t_o(\omega^2) p^2_{o,j}(\omega^1) + r^2_j(\omega^1) Q_j \\ & +(1-r^2_j(\omega^1))\sum_{k=1}^{n} y^2_{j,k}(\omega^1) q^2_k(\omega^1) \\ & \qquad \forall j \in [n], \omega^1 \in \Omega^1, \omega^2 \in \Omega^2 \end{aligned} \tag{4}$$

The robust magnetron problem is formulated as follows:

$$\begin{aligned} \min \quad & \sum_{j=1}^{n} c_j r^1_j q^1_j + \max_{\omega^1\in\Omega^1} \sum_{j=1}^{n} c_j r^2_j(\omega^1) q^2_j(\omega^1) \\ \text{s.t.} \quad & (1), (2), (3), (4) \\ & y^1_j, r^1_j \in \{0,1\} && \forall j \in [n] \\ & y^2_j(\omega^1), r^2_j(\omega^1) \in \{0,1\} && \forall j \in [n], \omega^1 \in \Omega^1 \\ & p^1_{o,j} \in \{0\} \cup [\underline{p}_j, \overline{p}_j] && \forall o \in C^1, j \in [n] \\ & p^2_{o,j}(\omega^1) \in \{0\} \cup [\underline{p}_j, \overline{p}_j] \\ & && \forall o \in C^2, j \in [n], \omega^1 \in \Omega^1 \\ & q^2_j(\omega^1) \in [0, Q_j] && \forall j \in [n], \omega^1 \in \Omega^1 \\ & q^3_j(\omega^1,\omega^2) \in [0, Q_j] \\ & && \forall j \in [n], \omega^1 \in \Omega^1, \omega^2 \in \Omega^2 \end{aligned} \tag{P}$$

The problem (P) has non-linear constraints. We are able to linearize all of these constraints and write (P) in the following form without loss of generality:

$$\begin{aligned} \min_{\tilde{x},\tilde{y}(\cdot)} \quad & \tilde{c}^\top \tilde{x} + \max\nolimits_{\omega^1\in\Omega^1} \tilde{d}^\top \tilde{y}(\omega^1) \\ \text{s.t.} \quad & \tilde{A}(\omega^1,\omega^2)\tilde{x} + \tilde{B}(\omega^1,\omega^2)\tilde{y}(\omega^1) \leqslant \tilde{b} \\ & \qquad \forall \omega^1 \in \Omega^1, \omega^2 \in \Omega^2 \end{aligned} \tag{5}$$

with $\tilde{A}$ and $\tilde{B}$ two matrices affinely depending on $(\omega^1,\omega^2) \in \Omega^1 \times \Omega^2$, and $\tilde{b}$, $\tilde{c}$, and $\tilde{d}$ are deterministic vectors. The variables are $\tilde{x}$ and $\tilde{y}(\cdot)$: the variable $\tilde{x}$ is the “here-and-now” variable, whose value has to be determined without knowing the exact $\omega^1 \in \Omega^1$ that will be selected, contrary to $\tilde{y}(\cdot)$—the “wait-and-see” recourse variable—whose value can arbitrarily depend on ω^1.

The problem (5) does not fit in the standard form of two-stage robust optimization problems because of the presence of a second uncertainty ω^2, which is revealed after all decisions have been taken. We use a standard duality technique of the robust optimization (Gorissen et al., 2015), which allows to transform the for all quantifier for $\omega^2 \in \Omega^2$ into an exists quantifier. The obtained problem form is the following:

$$\begin{aligned} \min_{x,y(\cdot)} \quad & c^\top x + \max\nolimits_{\omega\in\Omega} d^\top y(\omega) \\ \text{s.t.} \quad & A(\omega)x + B(\omega)y(\omega) \leqslant b \quad \forall \omega \in \Omega \end{aligned} \tag{6}$$

with A and B two matrices affinely depending on $\omega \in \Omega$, and b, c, and d are deterministic vectors. Note that in (6) the indices for uncertainty realization and for uncertainty set are dropped, as there is only one uncertainty set. The variables are x and $y(\cdot)$: the variable x is the “here-and-now” variable, whose value has to be determined without knowing the exact $\omega \in \Omega$ that will be selected, contrary to $y(\cdot)$—the “wait-and-see” recourse variable—whose value can arbitrarily depend on ω. The problem (6) is a complete adaptability version of a two-stage robust optimization problem, as the recourse is an arbitrary function of uncertainty.

4 METHOD

Solving exactly problem (6) is referred as *complete adaptability*, but this is in general considered as not tractable since it implies in particular computing an optimal recourse function $y(\cdot)$ that may be arbitrarily complicated. Bertsimas and Caramanis (2010) introduced the notion of *finite adaptability* (or *k-adaptability*). This approach bridges the gap between complete adaptability and a “myopic” approach—the *static* variant—taking a fixed recourse decision independently of the uncertainty. We address the magnetron problem via finite adaptability, and rewrite accordingly the magnetron problem as

$$\begin{aligned} \min_{\substack{\Omega_1\cup\cdots\cup\Omega_k \\ x,y_1,\ldots,y_k}} \quad & c^\top x + \max_{i\in[k]} d^\top y_i \\ \text{s.t.} \quad & A(\omega)x + B(\omega)y_i \leqslant b \quad \forall i\in[k] \;\; \forall\omega\in\Omega_i\,. \end{aligned} \tag{P_k}$$

The idea behind the k-adaptability is to split the uncertainty set Ω into k parts and to assign to each part a constant recourse decision. The advantage of such a method compared to the myopic one is that it allows to separate extreme scenarios and cover them with different recourse decisions. Multiple heuristics appeared in the last decade, allowing to solve (P_k), such as Bertsimas and Dunning (2016) or Postek and Hertog (2016). Subramanyam et al. (2020) proposed a branch-and-bound method solving (P_k) efficiently, with proven lower and upper bounds. It is this algorithm which we use to solve the problem.

This branch-and-bound takes as a parameter k, the number of parts in which the uncertainty set will be split. Each node of the branch-and-bound tree corresponds to a collection of k pairwise disjoint subsets $\Omega'_1,\ldots,\Omega'_k \subseteq \Omega$ (a "partial partition"). Branching corresponds to adding an element ω from Ω that is not yet covered by the Ω'_i, to each of the Ω'_i. This way, k branches stems from each node that is not a leaf. A "separation" subroutine tests whether an optimal solution to the partial partition is also valid for a full partition completing it. If this is the case, then we have reached a leaf. If not, the subroutine has actually found an ω to be added to the Ω'_i.

5 NUMERICAL EXPERIMENTS

Before describing briefly the instances and the results, we explain how the uncertainty is modeled in the experiments. We claim that this modeling is quite accurate.

5.1 Modeling Uncertainty

The only parameters that depend on the uncertainty are the production times, which we assume to be of the following form for campaign C^c. Given $\omega^c \in \Omega^c$, we have $t_o(\omega^c) = \hat{t}_o(1+p\omega^c_o)$, where p is a coefficient in $[0,1]$ and $\hat{t}_o$ is a coefficient in $\mathbb{R}$ depending on the order o. The uncertainty set Ω^c is defined as

$$\Omega^c := \left\{ (\omega^c_o)_{o\in C^c} : \omega^c_o \in [-1,1] \text{ and } \sum_{o\in C^c} \hat{t}_o\omega^c_o = 0 \right\}.$$

This is the most elementary way to model the uncertainty so that

- the production times of every order o can take any value in an interval centered at some value $\hat{t}_o$ which corresponds to the predicted value of t_o.
- the total duration of a campaign is fixed (does not depend on the uncertainty).

5.2 Instances and Setting

To assess the performance of the method proposed in Section 4, a real dataset corresponding to seven campaigns has been used to derive three two-campaign instances of the magnetron problem. The dataset corresponds to a magnetron of thirty cathodes of ten different materials. Each instance is of size $n = 30$, $m = 10$ with $|C^1| \in [10,12]$ and $|C^2| \in [10,12]$.

For these experiments, we have chosen to forbid replacements (i.e., the parameters $\bar{s}^1$ and $\bar{s}^2$ are set to zero). We have $\bar{r}^1 = 8$ and $\bar{r}^2 = 14$ for data1 and data2. For data3, we have $\bar{r}^1 = 14$ and $\bar{r}^2 = 8$.

Parameter c_j ranges from 200 to 1000 for $j = 1,\ldots,n,\ j \notin \{9,10,24\}$, $c_9 = c_{24} = 35$ and $c_{10} = 45$.

Regarding uncertainty, the experiments have been conducted with $p \in \{0.1, 0.2, 0.5\}$.

We used the open-source implementation of the algorithm of Subramanyam et al. (2020), based on the C callable library of CPLEX. All experiments were conducted on 32 cores of an Intel Xeon 2.30GHz computer, with a gap limit of 0.1%, a time limit of 3600 seconds and a memory limit of 10Gb.

5.3 Results

The three available instances have been solved using the branch-and-bound algorithm of Subramanyam et al. (2020) for all the values of $k \in [10]$. The solution obtained for k is used as a warm-start for $k+1$. Table 1 gives the numerical results for problem (P_k). The first column refers to the dataset. The next two columns give the parameters p and k respectively. Columns four and five provide the value of the objective function and the CPU time in seconds. The second-to-last column gives the average relative gap between the global lower bound for (P_k) (for the given k) and the achieved objective value (either optimal or provided by the incumbent branch-and-bound tree when the time limit is reached). The last column of the table displays the relative gain brought by k-adaptability for the associated value of k, compared to the 1-adaptability.

5.4 Comments

For the static version a nearly-optimal solution is found most of the time very quickly, but in some settings as for the instance data1 with $p = 0.5$, it is more time consuming. For $k > 1$ the gap significantly grows in most of cases, but for instances data1 and data2

Table 1: Numerical results for problem (P_k).

dataset	p	k	value	time (s)	gap (%)	gain vs $k = 1$ (%)
data1	0.20	1	10,514.60	144.12	0.10	0.00
data1	0.20	2	10,507.20	3,606.84	56.48	0.07
data1	0.20	5	10,498.10	3,606.50	56.45	0.16
data1	0.20	10	10,351.30	359.64	0.09	1.55
data2	0.20	1	4,983.26	1.93	0.10	0.00
data2	0.20	2	4,865.31	3,603.19	7.45	2.37
data2	0.20	5	4,865.31	3,604.79	7.30	2.37
data2	0.20	10	4,865.31	1,126.59	0.10	2.37
data3	0.20	1	8,215.78	1.82	0.04	0.00
data3	0.20	2	8,190.85	3,607.50	27.09	0.30
data3	0.20	5	7,634.37	3,603.38	26.79	7.08
data3	0.20	10	7,634.37	3,605.04	26.79	7.08
data1	0.50	1	25,275.90	957.35	0.10	0.00
data1	0.50	2	22,403.90	3,602.72	39.31	11.36
data1	0.50	5	22,282.40	3,605.31	49.60	11.84
data1	0.50	10	22,282.40	2,252.41	50.33	11.84
data2	0.50	1	5,433.87	2.71	0.00	0.00
data2	0.50	2	5,433.87	3,600.27	6.95	0.00
data2	0.50	5	5,433.87	3,605.70	7.66	0.00
data2	0.50	10	5,433.87	3,613.37	21.93	0.00
data3	0.50	1	11,700.30	1.36	0.04	0.00
data3	0.50	2	11,680.10	3,600.62	7.13	0.17
data3	0.50	5	11,627.40	3,602.26	13.05	0.62
data3	0.50	10	11,627.40	3,602.55	13.05	0.62

with $p = 0.2$ and $k = 10$ a nearly optimal solution is also found with an optimality gap close to zero.

The k-adaptability framework has a great impact on the quality of the obtained solutions. For some instances the gap for $k = 1$ is smaller than the relative gain brought by larger values of k. This means that it is guaranteed that increasing k in these cases leads to better decisions.

We do not know the exact description of the heuristic used by the practitioners but since it uses a fixed recourse, it is dominated by our results for $k = 1$.

Complementary experiments have shown that when $\bar{r}^1 = 12$, data1 with $p = 0.5$ has no feasible solution for $k = 1$, while there are feasible solutions for $k > 1$. It means that for some instances the k-adaptability may not only bring better solutions in terms of the objective function, but also and more importantly may bring feasibility.

6 CONCLUDING REMARKS

We formulated and modeled an industrial problem in the framework of finite adaptability and solved it with a branch-and-bound algorithm developed by Subramanyam et al. (2020). The experimental results show that for some instances finite adaptability brings more optimal solutions and even feasibility.

Two research directions can be further explored. The first one aims at comparing the long term effects of the finite adaptability and the static method. This study is interesting because a good short-term optimization may have bad impact on the long-term results. The second one seeks to adapt the problem formulation in the case of a heavy production planning. In that case the objective would be to minimize the number of refills, leading to the maintenance duration

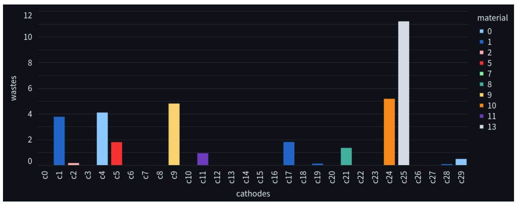

Figure 1: Wasted quantities of material (mm) before the first campaign for the static solution for data1 with $p = 0.50$.

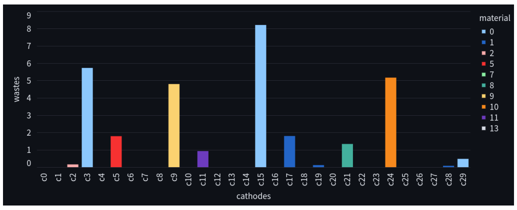

Figure 2: Wasted quantities of material (mm) before the first campaign for the 10-adaptable solution for data1 with $p = 0.50$.

minimization.

ACKNOWLEDGEMENTS

This research was supported by Saint Gobain Research Paris as part of a CIFRE collaboration with the CEDRIC laboratory of the CNAM Paris. We thank our colleagues Amaury Civrac and Sébastien Deschamps from who provided insight and expertise that greatly assisted the research. We also thank Bastien Rolland and his supervisor Tristan Barbe, who started working of this subject as part of an internship.

REFERENCES

Ben-Tal, A. and Nemirovski, A. (2000). Robust solutions of linear programming problems contaminated with uncertain data. *Mathematical programming*, 88(3):411–424.

Bertsimas, D. and Caramanis, C. (2010). Finite adaptability in multistage linear optimization. *IEEE Transactions on Automatic Control*, 55(12):2751–2766.

Bertsimas, D. and Dunning, I. (2016). Multistage robust mixed-integer optimization with adaptive partitions. *Operations Research*, 64(4):980–998.

Gabrel, V., Murat, C., and Thiele, A. (2014). Recent advances in robust optimization: An overview. *European journal of operational research*, 235(3):471–483.

Gicquel, C., Miégeville, N., Minoux, M., and Dallery, Y. (2010). Optimizing glass coating lines: Mip model and valid inequalities. *European Journal of Operational Research*, 202(3):747–755.

Gorissen, B. L., Yanıkoğlu, İ., and den Hertog, D. (2015). A practical guide to robust optimization. *Omega*, 53:124–137.

Korte, B. H., Vygen, J., Korte, B., and Vygen, J. (2011). *Combinatorial optimization*, volume 1. Springer.

Postek, K. and Hertog, D. d. (2016). Multistage adjustable robust mixed-integer optimization via iterative splitting of the uncertainty set. *INFORMS Journal on Computing*, 28(3):553–574.

Ruszczyński, A. and Shapiro, A. (2003). Stochastic programming models. *Handbooks in operations research and management science*, 10:1–64.

Subramanyam, A., Gounaris, C. E., and Wiesemann, W. (2020). K-adaptability in two-stage mixed-integer robust optimization. *Mathematical Programming Computation*, 12(2):193–224.