Robust Integrated Production-Maintenance Scheduling for an Evaporation Network

C.G. Palacín[a,*], J.L. Pitarch[a], C. Jasch[b], C.A. Méndez[c], C. de Prada[a]

[a] *Systems Engineering and Automatic Control DPT, Universidad de Valladolid. C/ Real de Burgos, s/n. EII Sede Mergelina. Valladolid, 47011, Spain*

[b] *Lenzing Aktiengesellschaft, Werkstraße 2, 4860 Lenzing, Austria*

[c] *Center for Advanced Process Systems Engineering (CAPSE). INTEC (UNL - CONICET). Industrial Engineering DPT (FIQ-UNL). Güemes, 3450 - Santa Fe (3000), Argentina*

Abstract

This work aims to reduce the global resource consumption in an industrial evaporation network by better tasks management and coordination. The network works in continuous, processing some products in several evaporation plants, so optimal load allocation and product-plant assignment problems appear. The plants have different features (capacity, equipment, etc.) and their performance is affected by fouling inside the heat exchangers and external factors. Hereby, the optimizer has to decide when maintenance operations have to be triggered. Therefore, a mixed production/maintenance scheduling problem arises. The plant behavior is approximated by surrogate linear models obtained experimentally, allowing thus the use of mixed-integer linear optimization routines to obtain solutions in acceptable time. Furthermore, uncertainty in the weather forecast and in the production plan is also considered via a two-stage stochastic programming approach. Finally, a trade-off analysis between other objectives of interest is given to support the decision maker.

Keywords: Production scheduling, Stochastic optimization, Integration, Fouling, Maintenance prediction, Similarity index

1. Introduction

Improved coordination of the operation in industrial sites can lead to enormous savings in the energy and resource consumption, and consequently to the

*Corresponding author
Email address: carlos.gomez@autom.uva.es (C.G. Palacín)

reduction of production costs, through the development of better decision support systems. This is the reason which motivates the development of methods and software for efficiency monitoring, coordinated process control and optimal planning and production scheduling of factories, industrial plants and parks under dynamically changing market conditions (Krämer & Engell, 2017). In particular, linking the real-time operational layers with the site-wide optimisation through *scheduling* approaches is receiving increased interest (Engell & Harjunkoski, 2012; Adamson et al., 2017).

These type of problems are positioned on the upper levels of the control hierarchy (see Figure 1) and usually involve both real-valued quantities (efficiency indicators, load assignment, etc.) and choosing between different discrete options (task execution, path decisions or available equipment). A scheduling problem has to consider three main points: which tasks have to be performed and when they will be executed, the equipment that will perform these tasks and the resources/time that will be required (and then allocated) for each task. It is noteworthy to say that the problem complexity is usually high and computational demands increase exponentially with the number of tasks.

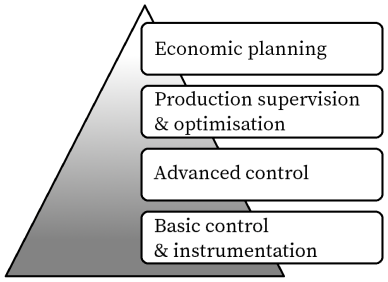


Figure 1: Automation pyramid

Scheduling is a complex and "exotic" task in industry, which is still often based on expert rules. However, these problems can be mathematically formulated, translated to optimization problems which involve both binary/integer and real decision variables, and solved efficiently using mixed-integer programming (MIP) (Méndez et al., 2006). According to the nature of the involved mathematical models (hybrid linear or nonlinear), these optimizations are formulated via constraint programming, mixed-integer linear (MILP) or nonlinear (MINLP) programming (Floudas, 1995). Hence, nowadays the computer-aided decision support tools are getting more and more significance in order to help operators to take better decisions (Harjunkoski et al., 2014). Nevertheless, there still exist many challenges to face with for the successful deployment of integrated scheduling tools in industrial environments(Harjunkoski, 2016).

This paper focuses on the *integration* between real-time optimization (RTO) and scheduling in a continuous industrial evaporation network with several plants and several products to concentrate. RTO is often used in large-scale systems

to seek online for the "optimal" control set points and design configurations to best face possible changes (production, weather disturbances, equipment failures, changeovers, etc.) (Adamson et al., 2017), but it usually bases on the current plant state and does not consider what can happen in the future. In the evaporation network, each plant runs in continuous with its own nominal efficiency, but this efficiency also decreases with time due to fouling effects inside the heat exchangers. Indeed, long-term effects which reduce performance, such as fouling or catalyst deactivation, are a common issue in industry. Therefore, *instantaneous* decisions may not be the best ones looking some time ahead. This drawback can be partially addressed by including models for such long-term effects in each plant RTO, e.g., the previous work from the authors for an evaporation plant in Pitarch et al. (2017).

Furthermore, these fouling effects force to periodically perform cleaning tasks in order to recover plant efficiencies, but not all plants can be stopped at the same time, so a mixed maintenance/production scheduling problem arises. Several people have devoted efforts along the last decades to deal with these kind of problems: Smaïli et al. (1999); Pogiatzis et al. (2012) proposed a MINLP approach while Lavaja & Bagajewicz (2004) and Casas-Liza et al. (2005) also proposed a time discretization to recast the problem as a MILP one. Recently, Biondi et al. (2017) presented a framework for industrial sites to integrate these two interactive decision processes in a multi-scale scheduling problem. In such work, the decay in equipment performance is modeled using the concept of "residual useful life" (RUL), a capacity resource which decreases over time. The evaporation plants in our case do not present a loss in capacity due to fouling (at least during the largest time they have been in operation without cleaning), but an increment in the specific steam consumption to achieve the desired evaporation set point. So the RUL approach is not directly applicable here. We present a predictive scheduling approach which uses a surrogate model including fouling to represent the plant behavior and we propose an adaptation of the general precedence approach to efficiently tackle the mixed maintenance/production problem, where three main types of tasks coexists: normal operation, standby and cleaning.

Moreover, uncertainty is always present when facing real problems (mismodeling, unplanned changes, disturbances, etc.). Therefore, considering uncertainty since the design phase in order to search for *robust* solutions is key. Robustness can be provided by forcing a single schedule to fulfill a bunch of scenarios, sampled according to expected realizations of the uncertainty (Kouvelis et al., 2000). However, this is usually a very conservative solution, i.e., normally the worst scenario does not realize. Thus, a less conservative option is using a multi-stage stochastic optimization (Grossmann et al., 2016), which benefits from the assumption that the uncertainty can be more precisely known in the future, so the schedule can provide different decisions for each scenario beyond the initial stage. Here we propose a two-stage optimization approach by considering uncertainty in the outdoor weather and in the production plan.

In addition, several *conflicting* optimization criteria such as efficiency, robustness or productivity appear (Yenisey & Yagmahan, 2014; Ruiz-Femenia et al., 2013). Therefore, based on the above approach and the concept of *similarity* between schedules, we propose a multi-criteria scheduling problem. Then, the offline analysis and visualization of the Pareto front (Reynoso-Meza et al., 2013) gives the plant engineers a guide about the right trade-offs to accomplish.

The rest of the paper is structured as follows: next section describes the evaporation process and discusses a suitable plant model; Section 3 describes the evaporation network together with the problem constraints, presents the proposed scheduling model and formulates the overall problem via disjunctive programming; then the two-stage stochastic and multiobjective optimizations are presented in sections 4 and 5 respectively; Section 6 shows some obtained solutions for normal operating conditions and analyzes the obtained Pareto front; and finally, a last section summarizes the main conclusions.

2. The evaporation process

This work is part of a series of actions to improve resource efficiency in Lenzing AG, a large industrial site which produces man-made cellulose fibers using wood as raw material. The main source of energy consumption in this site is located in the so-called spinning process, where the cellulose pulp extracted is extruded jointly with a solution of chemicals and water, in order to provide the fibers with the desired mechanical properties. These chemicals have a sensible economic value, so they must be reused. Thus, these aqueous solutions (henceforth products) are sent to an evaporation system in order to remove the main part of water. Finally, the output concentrate enters into a crystallization process.

2.1. Process description

Each evaporation plant is formed by a series of heat exchangers, evaporation chambers, condensers and cooling systems (see Figure 2). The evaporation is achieved by a multiple-effect process, as follows. The product enters the plant by the inlet located in the extreme of the evaporation chambers 2. Then it mixes with some concentrate product and flows through the heat exchangers in order to reach an adequate temperature (control set point). The first set of heat exchangers reuse steam coming from evaporation chambers 1, whereas the last set use fresh steam provided by boilers (the main source of energy consumption). Afterwards, the heated product enters sequentially into the low pressure evaporation chambers 1. Finally, a second evaporation is achieved in chambers 2, where the pressure is decreased using a condenser connected to a cooling system, typically a cooling tower. Last, part of the concentrate leaves the system by overflow and the remaining is mixed with the product inlet, being recirculated to the process (control set point).

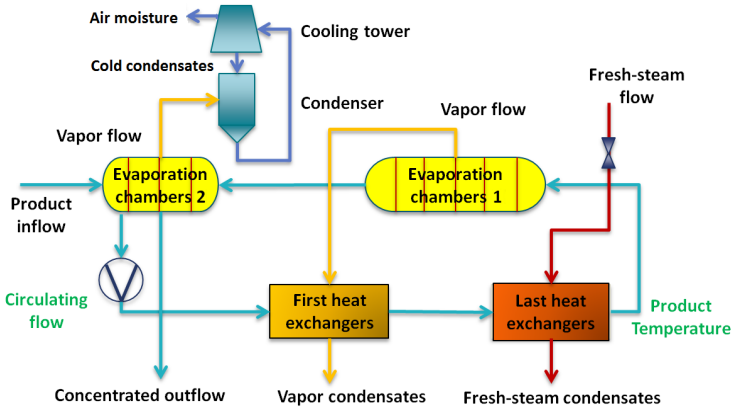


Figure 2: Scheme of a single multi-effect evaporation plant.

2.2. Plant surrogate modeling

The detailed modeling and optimization of a single evaporation plant was already addressed in previous works from the authors (Palacín et al., 2015; Pitarch et al., 2017). In these works, some Resource Efficiency Indicators (REI) were first defined in order to measure process efficiency in real time. The main REIs were the specific steam consumption and the normalized cost per time unit of operation. A nonlinear grey-box model, whose core is based on first principles, was developed for optimization purposes. Then, the proposed Real-Time Optimization (RTO) showed up some patterns for the optimal operation of a single plant and these were implemented following the concepts of self-optimizing control.

In addition, inside the heat exchangers a dirt layer grows during normal operation, caused by the deposition of organic material present in the products. This fouling effect slowly decreases the heat-transmission coefficient over time. Hence, a complex model to predict the evolution of these coefficients was identified experimentally and introduced into an economic optimization. In that way, an optimal cleaning policy for a single plant operating in isolation was already proposed in Pitarch et al. (2017). However, when considering several plants and products in a network, other aspects such as optimal assignment of products to plants or coordination of the maintenance operations have to be considered. Formulating such problem via MINLP becomes computationally challenging if the grey models developed in the above reference are used to represent the plants.

Therefore, thanks to the advantage that *near optimal* operation is currently achieved in each plant by the self-optimizing controller, we can build local surrogate plant models computed in different operating conditions (Bartz-Beielstein & Zaefferer, 2017), either from simulation with the nonlinear model or directly from process measurements. So, given a fouling state, these simula-

tions/measurements allow to record a static map of the cost function (fresh steam consumption) as a function of the outdoor temperature T_{out} and the product inlet P. This mapping turned out to be quite linear with these variables, so a linear approximation has been computed by least-squares identification. In addition, using experimental data recorded from the plant operating several months[1], an average increase of the specific steam consumption around 16% can be identified between consecutive cleaning operations (see Figure 3). Therefore, a linear evolution of the fouling effect can also be assumed, mainly depending on the time the evaporator has been in operation.

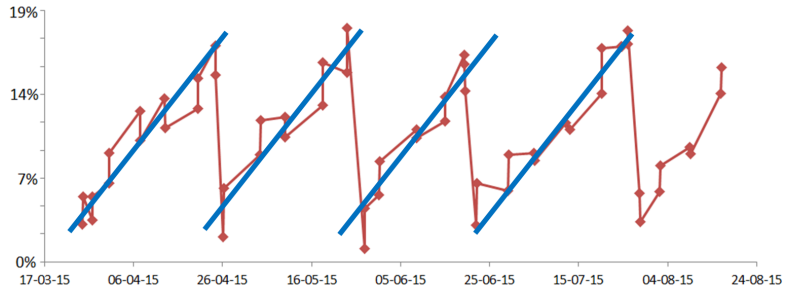


Figure 3: Evolution of the percentage increase of specific steam consumption due to fouling for several operation cycles.

In the end, the surrogate model representing the cost function for an evaporation plant v, processing a product p at time instant t, reads as follows:

$$Cost(v,t,p) = (K_T(v) \cdot T_{\text{out}}(t) + K_E(v)) \cdot P(v,t,p) + K_F(v,t) \qquad (1)$$

Where $K_T(v)$ depends on the efficiency of each cooling system, $K_E(v)$ represents the nominal efficiency of the evaporation plant v, and $K_F(v,t)$ is the increase of cost due to the current state of fouling. This plant model (1) is supported by extensive experimental work (as can be seen in Fig.3) and it is the basis for the proposed scheduling approach in the following sections. As illustrative example, Figure 4 shows three surfaces corresponding to three different fouling states.

Remark 1. Note that this approach has not only the advantage of sensibly reducing the computational cost required to solve the optimization, but also allows an easier maintenance of plant models by the process engineers.

3. The evaporation network

The network consists of several evaporation plants and some products to be concentrated. On the one hand, each product may be processed in several evaporation plants at the same time, but a plant can only process a single product

[1] In order to isolate the increase of steam consumption due to fouling, the plant is momentarily driven to reference conditions before taking a measurement.

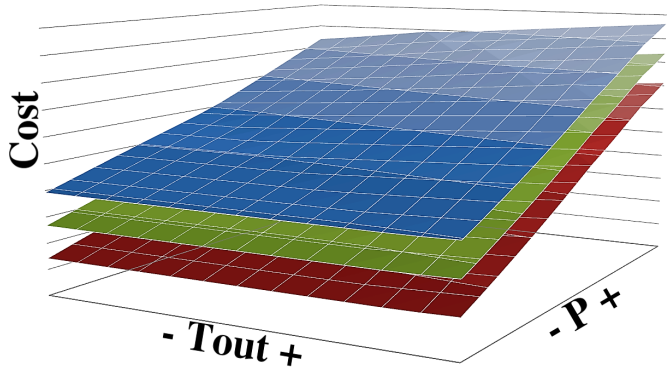


Figure 4: Surrogate model for the steam consumption in a single plant.

at a time. Therefore, given sets of p products and v evaporation plants, problems of plant assignment to products and load allocation appear (see Figure 5), where the operation cost differs from one plant to another due to the particular equipment efficiencies.

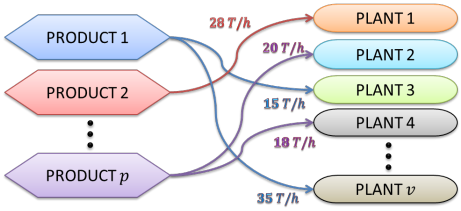


Figure 5: Plant assignment to products and load distribution.

On the other hand, the fouling, which reduces plant efficiencies, forces periodic stops for cleaning in order to recover the nominal values. There exist several aspects related to cleaning to be considered: (A) there are different cleaning types, each one with an associated cost K_C of manpower and cleaning products, and achieving different recoveries; (B) because of limitations on the available personnel, only one task can be performed at a time. Therefore, a maintenance scheduling problem appears, where we need to coordinate the plant cleaning stops (right time and type of task) while keeping the overall production and the use of resources in an optimal way. Note that, if one evaporator stops for cleaning, its load must be reassigned to the rest of operative plants.

Conventionally, production scheduling problems have been represented via network structures using *tasks* (any abstracted process operation) with several batches to schedule as basic elements. However, as here we are not deciding at the long-term planning, in this paper the meaning of task is closely linked to the state in which a plant can be (processing a product p, standby or un-

der a cleaning operation). Note that the number of equipment that can be used for operation is known, but the amount of tasks of any type that have to be performed within a time horizon is not known in advance, because the process is continuous, the future production may be uncertain and the best moment for cleaning the plants is not known. Handling common resources (total evaporation per product in this case) with a continuous-time approach requires synchronization constraints to be fulfilled at all time t, which makes the problem computationally very demanding, hence usually not suitable for real-time implementations.

3.1. Proposed modeling

To efficiently handle the synchronization constraints in the predictive scheduling problem of the evaporation network, the prediction horizon H has been discretized using one-day length as the shortest task unit. This choice is motivated by tree facts: 1) one day is the typical duration needed to complete a cleaning task and the fouling will not change significantly in one day, 2) resource-shared constraints are naturally handled in discrete time and 3) computational studies (Sundaramoorthy & Maravelias, 2011) showed that discrete-time formulations usually perform better in complex problems. Then, the underlying ideas of **general precedence allocation** (Méndez et al., 2006) are used here to force the operation accordingly to the known time evolution of the fouling effects, which must be followed by a limited number of cleaning alternatives.

Three different classes of stages are defined for the proposed automaton: working, cleaning and standby. The normal operation workflow for one evaporator is depicted in Figure 6, where the working stages are displayed as blue chevrons. These stages are related to the time that an evaporator has been in operation (v.gr., one evaporator that has started operation today will be in stage s_0, and one evaporator that has been working for two weeks will be in stage s_{14}). Using these stages, we will be able to indicate the plant performance degradation with time due to the fouling, i.e., if a plant v is in stage s, it will get an associated value $K_F(v, s)$ for the cost function (1). In this way, the state of an evaporator will advance with time through the chart stages: it will start working at stage s_0 if it is fully clean or from a more advanced one, let's name it s_c, if the cleaning was less deep (see Fig.6). In addition to this, in Pitarch et al. (2017) it was found by the authors that stopping a plant for cleaning during the first days of operation after a previous cleaning is not worthwhile, because in such case the normalized cost per time is huge (the fixed cost K_C associated to a cleaning task is not amortized yet). Hence, there will be a set of *initial stages* where no decision about cleaning needs to be checked.

Then, after leaving this initial set, the subsequent sets of stages s_A and s_B include the possibility of either continue operation, go directly to cleaning (to the small task A from s_A and to the big task B from s_B) or stop in standby awaiting for the cleaning resources to be available. Two available types of

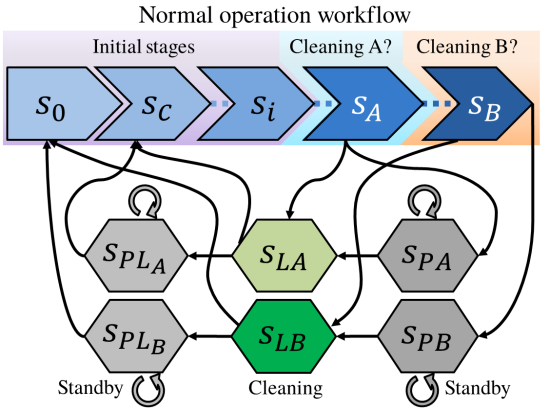


Figure 6: Simplified scheme of the automaton.

cleaning operations are represented in Figure 6 by the green hexagons, whereas the standby stages are represented by the grey ones. Also, an evaporation plant can be in standby after cleaning because it is not needed, or because it is not profitable to start working with this plant.

The option of stopping the plant in the middle of an operation cycle to continue operating without cleaning afterwards is not considered, as it is clearly suboptimal. Hence, once a plant has started operation, it must continue in operation as many days as stages are defined in the set of initial ones. This concept is analogous to the *minimum running time* in Velez et al. (2015) required once a task has started.

3.2. Logic formulation

There exist several alternatives in order to formulate scheduling problems via mixed-integer and disjunctive programming (Grossmann, 2002), each of them influencing the model structure, kind of software to be used and efficiency in obtaining a solution. Therefore, understanding the associated advantages and drawbacks of each option is key.

In this case, following the automaton proposed in the previous section, five different sets of entities are established:

- $\mathcal{V}$ denotes the set of all the evaporation plants.
- $\mathcal{S}$ will be the set of possible stages of an evaporation plant. As subsets it includes:
  - $\mathcal{S}_I$ as initial stages, defined as the ones where a stop for cleaning is not worthwhile. In particular, s_0 will be the first stage and s_c denotes a predefined stage to return operation after a less deep cleaning, e.g. of type A (Fig. 6).

  - $\mathcal{S}_A$ as stages where a decision between "keep working" or "make a cleaning task of type A" has to be made.
  - $\mathcal{S}_L$ as cleaning stages, where s_{LA} denotes a cleaning of type A.
  - $\mathcal{S}_P$ as the stages where the evaporation plants are stopped. Here we distinguish between two subsets: $\mathcal{S}_{PC}$ which includes stages where the plants are in standby waiting for a cleaning, e.g. s_{PA} denotes a standby before a cleaning of type A, and $\mathcal{S}_{PL}$ including stages where plants are in standby after being already cleaned, e.g. s_{PLA} denotes that the plant has been cleaned by type A.

- $\mathcal{M}$ denotes the set of all sample times in which the prediction horizon H is discretized. In particular:
  - t_F is the final time instant in the prediction horizon.
- $\mathcal{P}$ denotes the set of all products to be processed.
- Last, $\mathcal{E}$ denotes the set of all possible scenarios, i.e., the considered uncertainty realizations.

The variables that relate the above introduced sets are now defined:

- E_{vtse}: boolean variables which states that, in scenario e, an evaporation plant v is in stage s at time t.
- A_{vtpe}: boolean variables which, in scenario e, links a product p to a plant v at time t.
- P_{vtpe}: real non-negative variables that assigns, in scenario e, the evaporation flow of product p in plant v at time t.
- C_{vtse}: real non-negative variables that assigns, in scenario e, the costs for a plant v being in stage s at time t.

It must be noted that, even though current software tools can solve a complex nonlinear problem to global optimality, it is of little use if the solution is not at hand when a decision needs to be made (Lastusilta, 2011). Although algorithms and computers are improving and getting faster every day, in general (unless an efficient decomposition technique can be applied (Martí, 2015; Mitra et al., 2014; Zhang et al., 2016; Grossmann et al., 2016)) still only linear approaches are able to give reasonable solutions in reasonable time for large-scale problems (Klanšek, 2015). Therefore, according to linear generalized disjunctive programming (Sawaya & Grossmann, 2005), the feasible transitions within stages in the proposed automaton are defined by the following positive (i.e., True) logic statements[2]:

[2]A suitable reformulation will be required to enforce these statements via MILP. Several ways are available in the literature for this purpose, as discussed for instance in Balas (1985); Sawaya & Grossmann (2005).

1. An evaporator must be in one stage and only in one stage for each sample time.

$$\bigvee_{\overline{s\in\mathcal{S}}} E_{vtse} \;\; \forall v \in \mathcal{V},\; \forall t \in \mathcal{M},\; \forall e \in \mathcal{E} \tag{2}$$

2. An evaporator must be processing a single product, except if being cleaned or in standby.

$$\bigvee_{\overline{p\in\mathcal{P}}} (A_{vtpe}) \bigvee_{\overline{s\in\mathcal{S}_L}} (E_{vtse}) \bigvee_{\overline{s\in\mathcal{S}_P}} (E_{vtse}) \; \forall v \in \mathcal{V}, \forall t \in \mathcal{M}, \forall e \in \mathcal{E} \tag{3}$$

3. Only a single cleaning stage is allowed at any time period.

$$\bigvee_{\overline{s\in\mathcal{S}_L, v\in\mathcal{V}}} E_{vtse} \;\; \forall t \in \mathcal{M},\; \forall e \in \mathcal{E} \tag{4}$$

4. Initial stages of operation (where stopping to clean is not worthwhile) imply plants being in the next ones $(s+1)$ at the following sample time.

$$E_{vtse} \rightarrow E_{v(t+1)(s+1)e} \; \forall v \in \mathcal{V}, \forall t \in \mathcal{M}\backslash\{t_F\}, \forall s \in \mathcal{S}_I, \forall e \in \mathcal{E} \tag{5}$$

5. Accomplished a reasonable operation time, a choice can be made between continue operating, perform a suitable cleaning according to the current degree of fouling, or go to standby until cleaning.

$$E_{vtse} \rightarrow E_{v(t+1)(s+1)e} \vee E_{v(t+1)s_{LA}e} \vee E_{v(t+1)s_{PA}e} \\ \forall v \in \mathcal{V},\; \forall t \in \mathcal{M}\backslash\{t_F\},\; \forall s \in \mathcal{S}_A,\; \forall e \in \mathcal{E} \tag{6}$$

6. A stopped evaporator which has not been already cleaned, must be cleaned or continue in standby.

$$E_{vts_{PA}e} \rightarrow E_{v(t+1)s_{PA}e} \vee E_{v(t+1)s_{LA}e} \; \forall v \in \mathcal{V}, \forall t \in \mathcal{M}\backslash\{t_F\},\; \forall e \in \mathcal{E} \tag{7}$$

7. A clean evaporator in standby can continue in such state or begin to operate.

$$E_{vts_{PLA}e} \rightarrow E_{v(t+1)s_{PLA}e} \vee E_{v(t+1)s_ce} \;\; \forall v \in \mathcal{V},\; \forall t \in \mathcal{M}\backslash\{t_F\},\; \forall e \in \mathcal{E} \tag{8}$$

8. After a cleaning task, an evaporator can start operation or go to standby.

$$E_{vts_{LA}e} \rightarrow E_{v(t+1)s_{PLA}e} \vee E_{v(t+1)s_ce} \;\; \forall v \in \mathcal{V},\; \forall t \in \mathcal{M}\backslash\{t_F\},\; \forall e \in \mathcal{E} \tag{9}$$

9. When an evaporation plant is associated to a particular product, it must continue operating without product changes until it is cleaned.

$$A_{vtpe} \rightarrow A_{v(t+1)pe} \vee E_{v(t+1)se} \\ \forall v \in \mathcal{V},\; \forall t \in \mathcal{M}\backslash\{t_F\},\; \forall s \in \{\mathcal{S}_L \cup \mathcal{S}_{PC}\}, \forall p \in \mathcal{P},\; \forall e \in \mathcal{E} \tag{10}$$

10. Solutions which may reach a point of no return in the long term must be avoided, e.g., infeasibility (constraint violation) may happen in the future if several plants end up in an advanced fouling state when completing the schedule. This can be achieved by avoiding the evaporation plants to end up in a standby stage before cleaning

$$\neg \left(\bigvee_{s \in \mathcal{S}_{PC}} E_{vt_F se} \right) \quad \forall v \in \mathcal{V},\ \forall e \in \mathcal{E} \tag{11}$$

and by computing a *terminal cost* T_C if the plants end up working in an advanced fouling state[3]:

$$T_C = \sum_{s \in \mathcal{S}_A} E_{vt_F se} \cdot K_C(v, s_{LA})/2 \quad \forall v \in \mathcal{V},\ \forall e \in \mathcal{E} \tag{12}$$

In addition, each possible stage in which a plant can be gets an associated cost:

$$\left[\begin{array}{c} E_{vtse} \\ C_{vtse} = K(v,s) \end{array} \right] \vee \left[\begin{array}{c} \neg E_{vtse} \\ C_{vtse} = 0 \end{array} \right] \ \forall s \in \mathcal{S}, \forall v \in \mathcal{V},\ \forall t \in \mathcal{M},\ \forall e \in \mathcal{E} \tag{13}$$

Where, recalling (1), $K(v,s) \in \{K_F(v,s_0), K_F(v,s_1), \ldots, K_F(v,s_n), K_S(v,s_{PL}), K_C(v,s_{LA})\}$ stands for the entry to a lookup table containing the fixed cost values associated to each plant stage (fouling state, standby stages defined in $\mathcal{S}_P$ and the available cleaning operations in $\mathcal{S}_L$).

Production constraints must be also accomplished:

• The evaporation rate in each plant must be between minimum and maximum limits (or zero if the evaporator is stopped), where the upper limit $U_v(T_{\text{out}})$ gets a *known* dependency with the ambient temperature.

$$\left[\begin{array}{c} A_{vtpe} \\ L_v \leq P_{vtpe} \\ P_{vtpe} \leq U_v(T_{\text{out}}) \end{array} \right] \vee \left[\begin{array}{c} \neg A_{vtpe} \\ P_{vtpe} = 0 \end{array} \right] \quad \begin{array}{l} \forall v \in \mathcal{V},\ \forall t \in \mathcal{M}, \\ \forall p \in \mathcal{P},\ \forall e \in \mathcal{E} \end{array} \tag{14}$$

• A minimum evaporation rate must be accomplished for each product in each sample time.

$$\sum_{v \in \mathcal{V}} (P_{vtpe}) \geq SP_{pte} \quad \forall t \in \mathcal{M},\ \forall p \in \mathcal{P},\ \forall e \in \mathcal{E} \tag{15}$$

[3] A reasonable choice for this cost could be half of the cleaning cost, accounting that a cleaning task will become necessary in the next prediction period if a plant remains dirty at the current one.

The model is also constrained to the available physical connections between product circuits and evaporation plants $\mathcal{A}$ and to the initial (current) state of the plants $\mathcal{S}_0$. Then, once the scheduling model is complete, an optimization problem can be stated, for instance considering the usual risk-neutral economic objective[4] (Zhang et al., 2016) of minimizing the normalized cost of operation for the overall evaporation network during a time horizon H:

$$\begin{aligned} \min\ J_1 := & \sum_{e\in\mathcal{E}}\sum_{t\in\mathcal{M}}\sum_{v\in\mathcal{V}} \frac{\sum_{s\in\mathcal{S}} C_{vtse} + \sum_{p\in\mathcal{P}} \big(K_T(v)T_{\text{out}}(t) + K_E(v)\big)P_{vtpe}}{2^{\rho+1}\cdot H} + T_C \\ \text{s.t.: } & (2)-(15); A_{vtpe}\in\mathcal{A}\ \forall t\in\mathcal{M}, \forall e\in\mathcal{E};\ E_{v0se}\in\mathcal{S}_0\ \forall e\in\mathcal{E} \\ & C_{vtse}, P_{vtpe}\in\mathbb{R}^+;\ E_{vtse}, A_{vtpe}\in\{\text{True,False}\} \end{aligned} \tag{16}$$

The risk-neutral approach will suggest selecting scheduling policies where plants with higher production capacities are available to accommodate high-demand scenarios. Note that an aggregation based on currency is used here to form J_1, in order to lump together resources of different nature (steam, manpower, cleaning products, etc.) in a single efficiency indicator. Note also that, if only a nominal scenario is considered in $\mathcal{E}$, the above optimization (16) reduces to the deterministic scheduling approach.

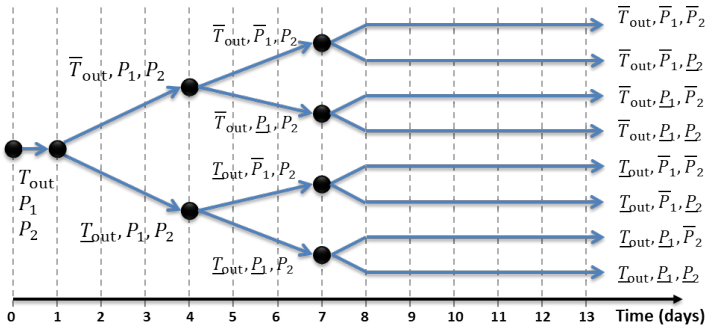
4. Two-stage stochastic schedule

The performance of the evaporation plants depends on two external factors: their load and the outdoor temperature, see (1). Consequently, if only the nominal scenario is considered in (16), any unplanned variation of the external factors during the prediction horizon may lead to a very degraded (or even infeasible) solution in practice. Therefore, production demand and weather forecasts are considered as stochastic variables supposed to take values within a certain convex region of uncertainty.

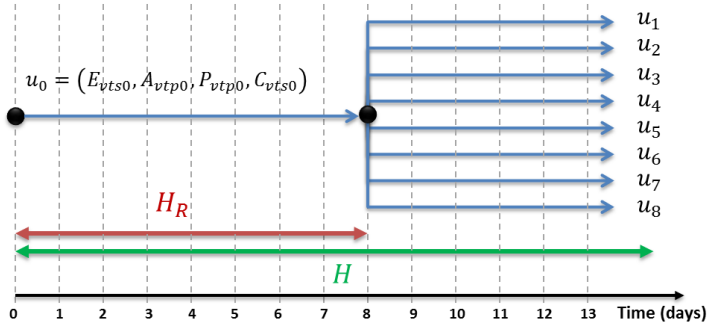
The multi-stage stochastic framework (Ruiz-Femenia et al., 2013; Grossmann et al., 2016) is adopted here to deal with such uncertainty considering that, even if at the current day there is no accurate information about the future realizations of the weather and evaporation loads, they will be known/measured at

[4]Of course risk-averse objectives weighting J_1 in (16) with the worst-case scenario or the CVaR can be considered too. However we do not expect further benefits with this approach, because more weight would be assigned to high-demand scenarios where evaporators are forced to run close to maximum total capacity, so no margin for a better coordination exists. Results in Section 6.3 are presented to support this insight. Note also that we did not approach the problem of scenario sampling from an uncertainty probability distribution, which is out of the scope of this paper. If a probability weight is assigned for each scenario in $\mathcal{E}$, it can be trivially included in J_1 to get a more realistic solution. Evidently, the more weight is assigned to low-demand scenarios, the greater is the benefit gained from the stochastic optimization.

some future days. Hence, it is possible to compute a bunch of recourse scheduling actions which fit each considered uncertainty realization. In particular, a two-stage approach is chosen because it is computationally less demanding, providing thus the possibility of recomputing the schedule in real time if needed. In this way, the prediction horizon H is split in two: a first "robust horizon" H_R computes a non-anticipative solution for all scenarios in $\mathcal{E}$, whereas the uncertainty does not grow in the remaining horizon, where individual decisions (recourse variables) are provided for each scenario. For example, a possible uncertainty tree is depicted in Figure 7a, where two max/min realizations for the outdoor temperature (extreme values $\overline{T}_{\text{out}}$ and $\underline{T}_{\text{out}}$ around the nominal prediction) are considered since the first day. Then, the evaporation demand for product P_1 may become uncertain at the fourth day, so it gets other two expected max/min values, and the demand for product P_2 might also be uncertain after day 7.



(a) Example of a possible scenario tree.



(b) Robust schedule with H_R set to 8.

Figure 7: Two-stage approach considering uncertainty in the weather and two products.

Note that, for instance, creating scenarios for the production of P_1 at day 4 does not necessarily imply that the evaporation demand is expected to change

at such day, but only that the master production plan is already fixed for the first 4 days. In fact, this scenario description mainly covers against production variations for P_1 until day 8, as we cannot certainly know the actual demand for day 8 until we reach day 4.

In the end, these considered uncertainty realizations make a 8 scenario tree. So, following the two-stage approach, decisions taken within the days belonging to a the robust horizon H_R must be unique, i.e., they must fulfil all the scenarios. See for instance the Figure 7b, where the scheduling decision variables u do not depend on scenario e until day 8. Then, from day 8 onwards, different decisions can be made for different scenarios. In a general case, a bunch of scenarios appear defined by all combinations between the considered values for each stochastic input. In particular for this case study, considering the two expected largest deviations for outdoor temperature times 2^{ρ} variations of evaporation demands (assuming ρ products) make a $2^{\rho+1}$ scenario tree in $\mathcal{E}$.

Remark 2. The robust horizon H_R is usually chosen as the future time window along which getting reasonable information of the actual uncertainty realization is not possible, e.g., typically a batch process where concentrations cannot be measured until the batch is finished. However, in continuous processes, H_R can be seen as a *user-defined* parameter to balance the approach. Indeed, note that if $H_R = H$, the linear formulation presented in the above section provides full guarantee of covering the entire region of uncertainty by just considering the reduced set of vertex realizations in the scenario tree. Thus, the high computational demands of probabilistic sampling approaches (Monte-Carlo based or similar) are avoided.

Now, denote by $\mathcal{M}_U$ the subset of $\mathcal{M}$ whose elements (days) do not belong to the robust horizon H_R. Thus, based in the above discussion, the nonanticipativity requirement must be enforced in H_R:

$$E_{vtse} \equiv E_{vts},\ A_{vtpe} \equiv A_{vtp},\ P_{vtpe} \equiv P_{vtp} \qquad \forall t \in \mathcal{M}\backslash\mathcal{M}_U,\ \forall v \in \mathcal{V},\ \forall e \in \mathcal{E},\ \forall p \in \mathcal{P} \quad (17)$$

In the end, the two-stage optimization problem reads as follows:

$$\begin{aligned} \text{minimize } J_1 \in \mathbb{R} \quad \text{subject to: } & (2)-(15);\ (17); \\ & A_{vtpe} \in \mathcal{A}\ \forall t \in \mathcal{M}, \forall e \in \mathcal{E};\ E_{v0se} \in \mathcal{S}_0\ \forall e \in \mathcal{E} \\ & C_{vtse}, P_{vtpe} \in \mathbb{R}^+;\ E_{vtse}, A_{vtpe} \in \{\text{True,False}\} \end{aligned} \quad (18)$$

Remark 3. Note that, for any particular future day, uncertainty in the weather forecast and in the production demands reduces as time advances (predictions become more reliable). Hence, the two-stage scheduling provides less conservative solutions through the recourse variables obtained within $\mathcal{M}_U$, which facilitates the possibility of measuring the current uncertainty realization in real time and adapting the schedule accordingly.

5. Multi-criteria decision making

In the above section, only an economic optimization is considered as an aim. However, an eventual decision maker must also consider additional aspects than economic ones in practice. Hence, other goals such as productivity or robustness, possibly conflicting with J_1 in (16), may be of interest for optimization too.

The two-stage stochastic approach already provides a tradeoff between robustness against the disturbances considered in the scenario tree and economic performance. However, there is always a small possibility that the suggested optimal schedule cannot be fully applied when the uncertainty realization is not explicitly considered in the scenario tree, even if the formulation is linear and the realization belongs to the convex region discussed in the previous section[5].

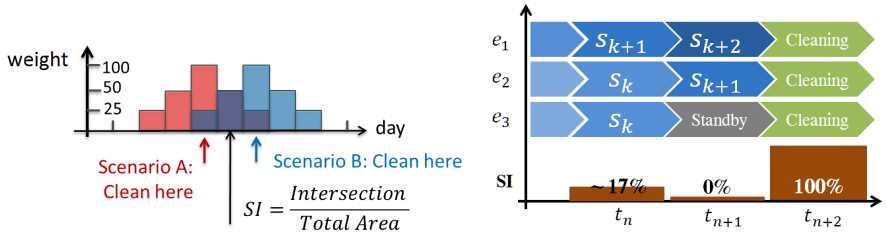
This drawback is inherent to problems involving discrete decisions. Theoretically, with a linear formulation, the optimal schedule which covers a realization belonging to the considered uncertainty set is a convex combination between the solutions corresponding to the vertex scenarios. However this optimal schedule cannot be computed except in, perhaps, a very few cases by sheer luck, because a convex combination of discrete values does not usually return another discrete value. Therefore, some values must be "rounded" to the nearest or the most probable discrete one in order to apply a schedule in practice. This runs the *risk* of being in the actual performance far from the predicted one or, in the worst case, it may lead to infeasibility.

A straightforward way to minimize such risk is either considering more scenarios or enlarging H_R (Remark 2). The first option is discarded for practical implementations (it increases the computational burden considerably). Thus, how to choose the length of H_R to achieve a desired risk reduction without being too conservative would be the question to answer. In this last case, an index to measure robustness could be defined as $J_2 := H_R/H$. However, this way may become conservative, as the worst-case combination of uncertainty realizations may happen at the beginning of the prediction horizon H so, when H_R includes such day, the conservatism-reduction advantages of the two-stage approach vanish.

To overcome this drawback and to force a desired robustness level without explicitly varying H_R, we introduce the concept of *similarity* between schedules (Palacín et al., 2017). A similarity index (SI) will be defined in order to indicate whether a suggested schedule is closer to the more risky one obtained by the standard two-stage approach (Section 4), or to the risk averse single schedule ($H_R = H$). The idea is inspired in the concept of *minimum agreement index*

[5]If the formulation is nonlinear in decision variables or the the actual realization does not belong to the convex hull formed by the considered region of uncertainty, no guarantees can be ensured. The only option in such cases is reschedule including that realization as an additional scenario in the tree, although no feasibility guarantees can be provided either.

for fuzzy duedate or fuzzy completion time (Sakawa & Kubota, 2000) and is as follows. First, discrete binary decisions taken each day for a particular evaporator and scenario are *fuzzified* along the surrounding days, e.g., a decision takes a value of 100 at the suggested day but it also influences the before and following days with a decreasing value, proportional to the distance from the current day. Then, the SI is defined as the intersection between the fuzzified schedules computed for all the scenarios, see Figure 8.



(a) Fuzzification of a discrete decision.

(b) SI values per day for one plant.

Figure 8: Illustrative SI calculation.

Then, the values computed for all days, stages and plants must be aggregated together in a single SI indicator. Thus, using just the two closest days to the current one $(t-1, t+1)$ for fuzzification, and abusing notation setting True=1 and False=0 for E_{vtse}, the SI index reads

$$S_I := \sum_{v\in\mathcal{V}}\sum_{s\in\mathcal{S}}\sum_{t\in\mathcal{M}_U\setminus t_F} \frac{\min\left(100E_{vtse}, 50E_{v(t+1)se}, 50E_{v(t-1)se}\right)}{n_v(200(n_u-1)+150)} \tag{19}$$

where n_v is the number of evaporation plants and n_u is the number of days in $\mathcal{M}_U$. Hence, a $S_I = 100\%$ means that the schedules coincide for all scenarios, so there is just a single schedule: the risk-averse solution.

The reader may realize that the SI as defined in (19) is nonlinear in decision variables. However, a lower bound for it can be set introducing slack variables $S_{vtp} \in \mathbb{R}^+$ jointly with the following additional linear disjunctions:

$$\begin{bmatrix} E_{vtse} \\ S_{vts} \le 100 \end{bmatrix} \vee \begin{bmatrix} E_{v(t+1)se} \\ S_{vts} \le 50 \end{bmatrix} \vee \begin{bmatrix} E_{v(t-1)se} \\ S_{vts} \le 50 \end{bmatrix} \vee \begin{bmatrix} \neg(E_{vtse} \vee E_{v(t+1)se} \vee E_{v(t-1)tse}) \\ S_{vts} = 0 \end{bmatrix} \quad \begin{matrix} \forall t \in \mathcal{M}_U\setminus t_F, \\ \forall v \in \mathcal{V}, \forall s \in \mathcal{S} \end{matrix} \tag{20}$$

In this way, the SI can be bounded by:

$$J_3 := \sum_{v\in\mathcal{V}}\sum_{s\in\mathcal{S}}\sum_{t\in\mathcal{M}_U\setminus t_F} \frac{S_{vts}}{n_v(200(n_u-1)+150)} \le S_I \tag{21}$$

Last, the plant manager might eventually be interested in analyzing how varying the overall *productivity* affects the other objectives, e.g., to fit market conditions,

to maximize the profit (i.e., minimize the cost per ton of produced fibre) or to decide over future equipment investments. In order to approach this, note that the evaporation demands for each product are set by parameters SP_{pte} in (15). So, the lowest demand δ for all scenarios, products and days is:

$$J_4 := \delta = \min\left(SP_{pte}\right) \quad \forall e \in \mathcal{E},\ \forall t \in \mathcal{M},\ \forall p \in \mathcal{P} \tag{22}$$

Hence, the gaps $\Delta P_{pte} := SP_{pte} - \delta$ can be computed too. Now, if δ becomes decision variable, a way to uniformly decide over the evaporation demands via (15) is varying δ, adding constraints (23) to compute new set points with the already fixed ΔP_{pte}.

$$SP_{pte} = \Delta P_{pte} + \delta \quad \forall e \in \mathcal{E},\ \forall t \in \mathcal{M},\ \forall p \in \mathcal{P} \tag{23}$$

In order to simultaneously analyze the interactions between these opposite objectives (cost, robustness and productivity) and to provide the manager with the significant information at a glance, we come up with a multi-objective optimization problem (MOOP) formulated as follows:

$$\begin{aligned} \text{minimize } & J = [J_1, -J_3, -J_4] \in \mathbb{R}^3 \quad \text{subject to: } (2)-(17); (20); (23); \\ & A_{vtpe} \in \mathcal{A}\ \forall t \in \mathcal{M}, \forall e \in \mathcal{E};\ E_{v0se} \in \mathcal{S}_0\ \forall e \in \mathcal{E} \\ & C_{vtse}, P_{vtpe}, S_{vts}, \delta \in \mathbb{R}^+;\ E_{vtse}, A_{vtpe} \in \{\text{True,False}\} \end{aligned} \tag{24}$$

In order to solve (24) using efficient MILP software, we set additional constraints with bounds in J_3 and J_4, denoted by $\mathbf{J}_3$ and $\mathbf{J}_4$, defining a grid within the pertinency region[6] (Reynoso-Meza et al., 2014) so that only J_1 is in the objective function. In this way, the original MOOP is cast as a set of single-objective optimizations:

$$\begin{aligned} \text{minimize } & J_1 \in \mathbb{R} \quad \text{subject to: } (2)-(17); (20); (23);\ J_3 \geq \mathbf{J}_3;\ \delta \geq \mathbf{J}_4; \\ & A_{vtpe} \in \mathcal{A}\ \forall t \in \mathcal{M}, \forall e \in \mathcal{E};\ E_{v0se} \in \mathcal{S}_0\ \forall e \in \mathcal{E} \\ & C_{vtse}, P_{vtpe}, S_{vts}, \delta \in \mathbb{R}^+;\ E_{vtse}, A_{vtpe} \in \{\text{True,False}\} \end{aligned} \tag{25}$$

Finally, a Pareto Front can be computed offline and its analysis will result in valuable information for the decision maker to choose which level of risk to assume depending on the permissible slack to vary production demands.

6. Illustrative results

In order to check the effectiveness of the proposed approach, several schedules have been computed from simulated data. As a first trial, we do not consider uncertainty, so we look for an efficient deterministic solution for the actual network. Then, uncertainty is introduced in two smaller instances of the problem and the obtained stochastic solutions are discussed.

[6] Range where productivity and robustness is considered acceptable.

6.1. Deterministic solution

In this case, the whole network of 23 plants to process 5 products (A,B,C,D,E) is considered. The allowed physical connections between products and plants as well as nominal efficiencies[7] are listed in Table 1.

	p_1	p_2	p_3	p_4	p_5	K_E
v_1	✓	✗	✗	✓	✓	0.6
v_2	✓	✓	✗	✓	✓	0.7
v_3	✓	✗	✗	✓	✓	0.8
v_4	✓	✗	✗	✓	✓	0.9
v_5	✓	✓	✗	✓	✗	1
v_6	✓	✓	✗	✓	✓	1.1
v_7	✓	✗	✗	✓	✗	1.2
v_8	✓	✗	✗	✓	✗	1.3
v_9	✓	✗	✗	✓	✓	1.4
v_{10}	✓	✓	✓	✓	✗	1.1
v_{11}	✓	✓	✗	✓	✓	1.11
v_{12}	✓	✗	✓	✓	✗	1.12
v_{13}	✗	✗	✗	✓	✓	1.13
v_{14}	✗	✓	✗	✓	✗	1.14
v_{15}	✓	✓	✓	✓	✓	1.15
v_{16}	✗	✓	✓	✓	✓	0.9
v_{17}	✓	✓	✓	✗	✓	1
v_{18}	✗	✓	✓	✗	✓	1.1
v_{19}	✗	✓	✓	✓	✓	1.2
v_{20}	✓	✓	✓	✓	✓	1.3
v_{21}	✗	✓	✗	✓	✓	1.4
v_{22}	✓	✓	✗	✗	✓	0.7
v_{23}	✓	✓	✓	✓	✓	0.8

Table 1: Connections product-plant and nominal efficiencies.

Each plant cannot operate under a load of $L_v = 15$ T/h, but their maximum capacity varies with the weather condition, i.e. $U_v = 30 + f(T_{\text{out}})$, where $f(\cdot)$ is such that $U_v \leq 35$ T/h. Two types of cleaning tasks have been considered, small (A) and big (B), with their corresponding associated costs $K_C(v, s_{LA})$ and $K_C(v, s_{LB})$ of manpower and chemical products. Marginal costs $K_S(v, s_{PA})$ and $K_S(v, s_{PB})$ have been assigned to the waiting stages before cleaning to avoid persistent situations in time where evaporators are not used but remain dirty, which may lead to an overall loss of efficiency when they will be needed (for instance against unexpected production increments). Also, analyzing the plant historian, it has been observed that an evaporator cannot operate during more than 40 days without cleaning because it is clearly suboptimal. So, the set $\mathcal{S}$ is formed by $\{s_0, s_1, \ldots, s_{40}, s_{LA}, s_{LB}, s_{PA}, s_{PB}, s_{PL_A}, s_{PL_B}\}$.

For this test, the desired set points of evaporated water per product are set to $SP_1 = 120$, $SP_2 = 76$, $SP_3 = 64$, $SP_4 = 146$ and $SP_5 = 68$ T/h. Hence, given a randomly fixed initial state of the network together with the above constraints, we run the economic optimization (16) to provide the optimal load allocation as well as the task schedule within a prediction horizon of $H = 30$ days.

[7]Scaled values. Real ones are not included due to confidentiality agreements with Lenzing AG.

An optimal solution with relative gap less than 1% has been found for this problem (35150 binary variables, 3452 real ones and 42613 constraints) in about 11 min using up to 4 threads for concurrent optimization in GAMS with GUROBI 7.0.2 over an Intel® i7-4510U CPU machine with 16 Gb of RAM memory[8]. This solution is depicted in the Gantt diagram of Figure 9, where the state evolution for each evaporator is shown over the horizontal axis. Columns represent the days, and each cell shows the product load which has to be processed in each plant, i.e., the value of P_{etp}. The product type is represented by the background color, whereas darkness indicates the plant fouling state.

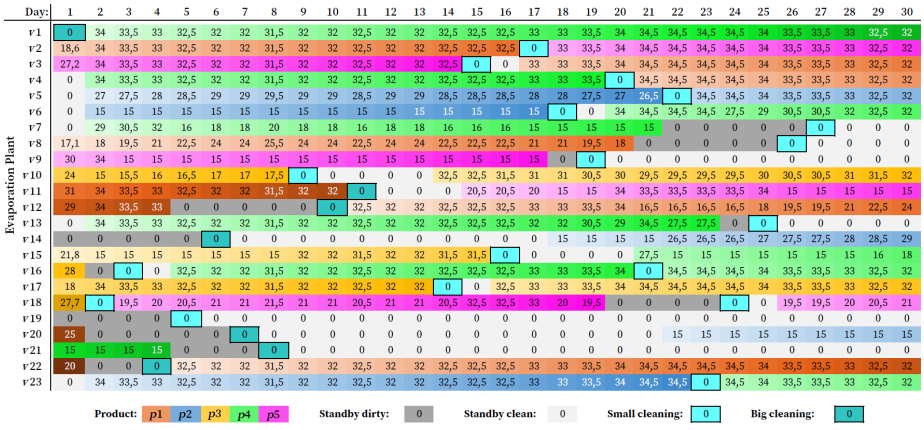


Figure 9: Optimal deterministic schedule.

The computed schedule shows how the optimizer tries to avoid using the less efficient plants when possible, either because they get higher K_E or they are dirtier than others. The cleaning tasks are scheduled in the better way, involving switching to a different product as long as overall efficiency is achieved. Finally, only 4 plants are in a relatively dirty state at the end of the prediction horizon, so the final network state guarantees feasibility in future runs.

6.2. Two-stage stochastic solution

Now we introduce uncertainty as explained in Section 4. First, a handy example with 3 plants and 2 products is provided for a better understanding of the further results. In this example, plant v_1 can work with both products whereas plant v_2 is assigned to p_1 and v_3 to p_2. Plant efficiencies are set to $K_{Ev_1} = 0.6$, $K_{Ev_2} = 0.7$ and $K_{Ev_3} = 0.8$. The set points of evaporated water are initially set to $SP_1 = 32$ and $SP_2 = 25$ T/h.

[8]Reducing the gap to 0.5% elapses 20 min and the proven optimal solution (zero gap) is got in about one hour, but this extra computational effort is not worthwhile in practice.

Then, for simplicity only uncertainty in the production is introduced, setting $H_R = 7$. The considered largest deviations from the set points are $\sigma_{p_1} = 6$ and $\sigma_{p_2} = 4$ T/h. Hence, just considering the max/min vertex values for the uncertainty realizations, a 4-scenario tree arises. The obtained two-stage stochastic schedule computed by solving (18) with a given initial plant states is depicted in Figure (10).

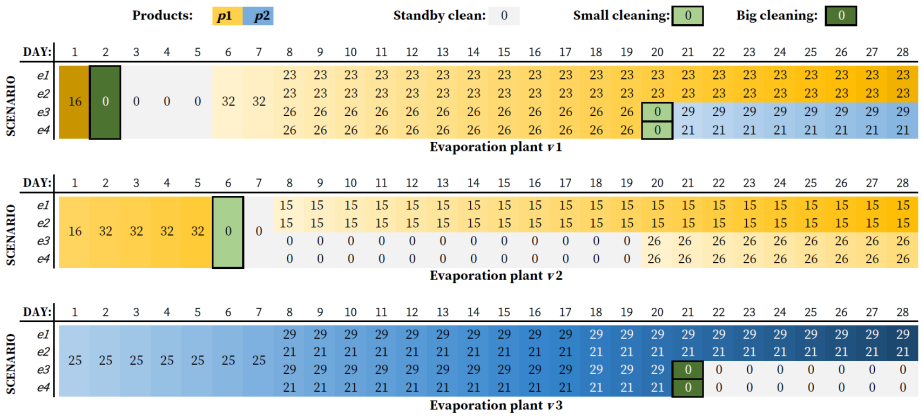


Figure 10: Two-stage stochastic schedule for 3 plants and 2 products.

Now, a more complex network subset considering 3 products and 9 plants is set up. The set point of evaporated water is set to 40 T/h per product and the physical connections between products and plants for this case are shown in Table 2.

	V_1	V_2	V_3	V_4	V_5	V_6	V_7	V_8	V_9
P_1	✗	✓	✓	✓	✓	✗	✓	✓	✗
P_2	✓	✓	✗	✗	✓	✓	✓	✓	✓
P_3	✓	✓	✓	✓	✓	✓	✓	✗	✗
K_E	1	0.88	1.1	1.01	0.77	0.95	1.2	1	1.05

Table 2: Connections product-plant and nominal efficiencies.

Here we introduce uncertainty in both the weather and in the production plan for each product. The expected largest deviations for these sources of uncertainty are $\sigma_{T_{\text{out}}} = 7$°C and $\sigma_p = 6$ T/h respectively. Hence, with 3 products, a 16-scenario tree arises. In this case, in order to get proven optimal solutions in acceptable times, the prediction horizon has been reduced to $H = 25$ days. Problem size is 130536 binary variables, 7994 real ones and 152863 constraints. Solving (18) with CPLEX 24.8.5 in the same machine returns the 1%-gap optimal solution in 5 minutes. Figure (11) depicts the computed two-stage stochastic schedule for plants 2 and 8 (the rest are omitted due to space constraints).

For completeness, if (4) is relaxed to allow cleaning several plants at the same day, only a relative cost improvement around 0.02% is achieved. So, the option of hiring more personnel for cleaning does not seem potentially worthwhile.

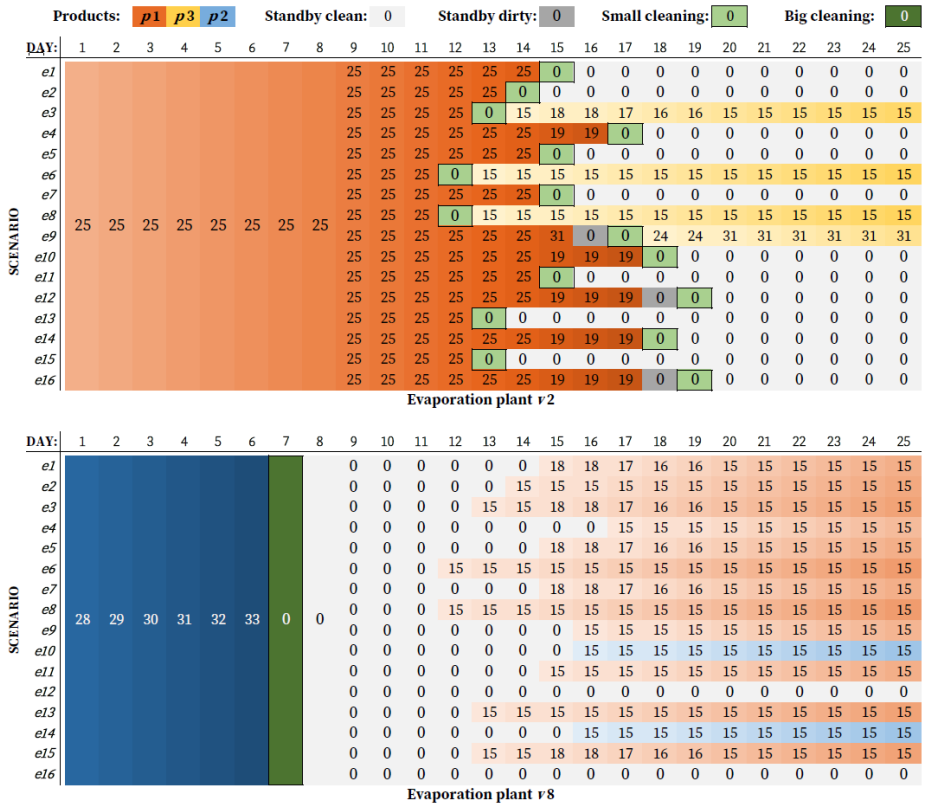


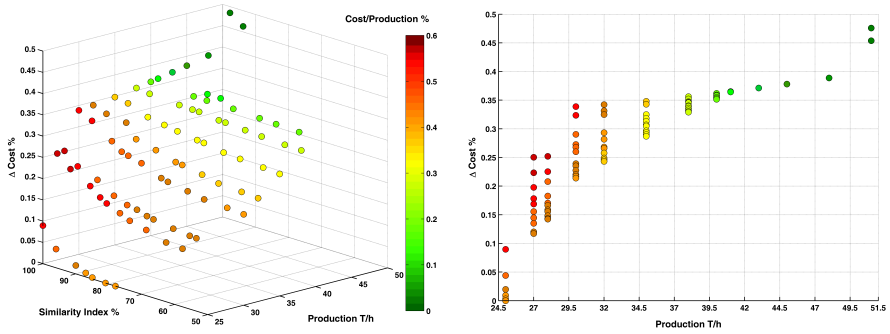
Figure 11: Two-stage stochastic schedule.

6.3. Multi-objective analysis

Now, defining a well distributed grid of points within the pertinency range for J_3 and J_4, optimal solutions in the Pareto sense can be computed by solving (25) offline. A set of points approximating the Pareto front is depicted in Figure 12. Some interesting conclusions can be extracted from its shape:

- Evidently, the absolute cost increases with the production. However, the sensitivity is higher for low productions, see Figure 12b.
- Sensitivity from cost to robustness is also higher at low productions. Indeed, the amount of different solutions reduces as production increases (see again Fig.12), tending to the single one with SI=100%, which suggests that the two-stage stochastic approach is a waste of computational resources when production is constrained to be high.
- Finally, if we look at the specific cost per amount of production (represented by the colormap) instead of the absolute cost, the lowest overall efficiency is achieved for low productions. However, this result is due to

the fact that all plants in operation must be cleaned after some time, despite of whether they are working at low load, so the fixed costs of cleaning tasks make the overall specific cost increase.



(a) 3D view. Color indicates specific cost. (b) 2D view from objectives J_1 and J_3.

Figure 12: Approximation of the Pareto front.

7. Conclusion

This paper addresses a scheduling problem for an industrial evaporation network affected by long-term fouling effects and uncertainty in external factors. Consequently, the equipment cannot operate forever without stopping to perform maintenance tasks in order to recover efficiency. The main feature which makes the problem singular from the formulation side is that the RTO needs to be extended to the network scheduling in a computationally tractable way, also considering uncertainty. A discretization in days and a modification of the general precedence allocation method have been proposed to efficiently tackle this problem.

Uncertainty has been introduced in the weather prediction and in the production plan via a two-stage stochastic optimization approach. In this way, less conservative robust solutions are obtained by computing different schedules for some expected uncertainty realizations in the future. Moreover, the proposed approach gives solutions in acceptable time, so we could also take advantage of periodically measuring the actual external factors and reschedule accordingly if needed.

Furthermore, a multi-criteria optimization is proposed by adding other objectives of interest to the economic one, in order to provide plant managers with significant information about the possibility varying the production or the assumed risk against unconsidered scenarios. A similarity index between scenario-based solutions has been proposed as a measure of robustness in order to give the scheduler the possibility of reducing such risk at the price of increased conservatism. The shape analysis of the obtained Pareto front, although dependent on

the current state of the network, provides interesting conclusions which provide plant engineers with valuable information to design decision-support systems.

Finally, the approach is tested in simulation with several instances of the evaporation network. The results were promising so that the two-stage stochastic approach can be progressively extended to the whole network. Nevertheless, eventually we will face larger problems when including more facilities so, in order to keep the resolution times within feasible ranges, our future work will explore decomposition methods for the overall problem.

Acknowledgments

These results are part of the CoPro project which has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 723575 and from the Spanish Government with project INOPTCON (MINECO/FEDER DPI2015-70975-P).

References

Adamson, R., Hobbs, M., Silcock, A., & Willis, M. J. (2017). Integrated real-time production scheduling of a multiple cryogenic air separation unit and compressor plant. *Computers & Chemical Engineering*, *104*, 25 – 37.

Balas, E. (1985). Disjunctive programming and a hierarchy of relaxations for discrete optimization problems. *SIAM Journal on Algebraic Discrete Methods*, *6*, 466–486.

Bartz-Beielstein, T., & Zaefferer, M. (2017). Model-based methods for continuous and discrete global optimization. *Applied Soft Computing*, *55*, 154 – 167.

Biondi, M., Sand, G., & Harjunkoski, I. (2017). Optimization of multipurpose process plant operations: A multi-time-scale maintenance and production scheduling approach. *Computers & Chemical Engineering*, *99*, 325 – 339.

Casas-Liza, J., Pinto, J., & Papageorgiou, L. (2005). Mixed integer optimization for cyclic scheduling of multiproduct plants under exponential performance decay. *Chemical Engineering Research and Design*, *83*, 1208–1217.

Engell, S., & Harjunkoski, I. (2012). Optimal operation: Scheduling, advanced control and their integration. *Computers & Chemical Engineering*, *47*, 121 – 133.

Floudas, C. A. (1995). *Nonlinear and mixed-integer optimization: fundamentals and applications*. Oxford University Press on Demand.

Grossmann, I. E. (2002). Review of nonlinear mixed-integer and disjunctive programming techniques. *Optimization and engineering*, *3*, 227–252.

Grossmann, I. E., Apap, R. M., Calfa, B. A., García-Herreros, P., & Zhang, Q. (2016). Recent advances in mathematical programming techniques for the optimization of process systems under uncertainty. *Computers & Chemical Engineering*, *91*, 3 – 14.

Harjunkoski, I. (2016). Deploying scheduling solutions in an industrial environment. *Computers & Chemical Engineering*, *91*, 127 – 135.

Harjunkoski, I., Maravelias, C. T., Bongers, P., Castro, P. M., Engell, S., Grossmann, I. E., Hooker, J., Méndez, C., Sand, G., & Wassick, J. (2014). Scope for industrial applications of production scheduling models and solution methods. *Computers & Chemical Engineering*, *62*, 161–193.

Klanšek, U. (2015). A comparison between MILP and MINLP approaches to optimal solution of nonlinear discrete transportation problem. *Transport*, *30*, 135–144.

Kouvelis, P., Daniels, R. L., & Vairaktarakis, G. (2000). Robust scheduling of a two-machine flow shop with uncertain processing times. *IIE Transactions*, *32*, 421–432.

Krämer, S., & Engell, S. (2017). *Resource Efficiency of Processing Plants: Monitoring and Improvement*. Wiley. (In press).

Lastusilta, T. (2011). *GAMS MINLP solver comparisons and some improvements to the AlphaECP algorithm*. Ph.D. thesis Åbo Akademi University Finland.

Lavaja, J. H., & Bagajewicz, M. J. (2004). On a new milp model for the planning of heat-exchanger network cleaning. *Industrial & engineering chemistry research*, *43*, 3924–3938.

Martí, R. (2015). *Price Coordination Strategies in Large-Scale Process Control*. Ph.D. thesis Escuela de Ingenierías Industriales, Universidad de Valladolid Spain.

Méndez, C. A., Cerdá, J., Grossmann, I. E., Harjunkoski, I., & Fahl, M. (2006). State-of-the-art review of optimization methods for short-term scheduling of batch processes. *Computers & Chemical Engineering*, *30*, 913–946.

Mitra, S., Pinto, J. M., & Grossmann, I. E. (2014). Optimal multi-scale capacity planning for power-intensive continuous processes under time-sensitive electricity prices and demand uncertainty. part ii: Enhanced hybrid bi-level decomposition. *Computers & Chemical Engineering*, *65*, 102 – 111.

Palacín, C. G., Pitarch, J. L., & de Prada, C. (2015). Efficient modelling and real-time optimisation of stationary systems: Application to an evaporation process. In *Proceedings of SIMUL 2015: The 7th Intern. Conf. on Advances in System Simulation* (pp. 6–11). IARIA XPS Press.

Palacín, C. G., Pitarch, J. L., de Prada, C., & Méndez, C. A. (2017). Robust multi-objective scheduling in an evaporation network. In *2017 25th Mediterranean Conference on Control and Automation (MED)* (pp. 666–671).

Pitarch, J. L., Palacín, C. G., de Prada, C., Voglauer, B., & Seyfriedsberger, G. (2017). Optimisation of the resource efficiency in an industrial evaporation system. *Journal of Process Control*, *56*, 1–12.

Pogiatzis, T. A., Wilson, D. I., & Vassiliadis, V. S. (2012). Scheduling the cleaning actions for a fouled heat exchanger subject to ageing: Minlp formulation. *Computers & Chemical Engineering*, *39*, 179–185.

Reynoso-Meza, G., Blasco, X., Sanchis, J., & Herrero, J. M. (2013). Comparison of design concepts in multi-criteria decision-making using level diagrams. *Information Sciences*, *221*, 124 – 141.

Reynoso-Meza, G., Sanchis, J., Blasco, X., & García-Nieto, S. (2014). Physical programming for preference driven evolutionary multi-objective optimization. *Applied Soft Computing*, *24*, 341 – 362.

Ruiz-Femenia, R., Guillén-Gosálbez, G., Jiménez, L., & Caballero, J. A. (2013). Multi-objective optimization of environmentally conscious chemical supply chains under demand uncertainty. *Chemical Engineering Science*, *95*, 1 – 11.

Sakawa, M., & Kubota, R. (2000). Fuzzy programming for multiobjective job shop scheduling with fuzzy processing time and fuzzy duedate through genetic algorithms. *European Journal of Operational Research*, *120*, 393 – 407.

Sawaya, N. W., & Grossmann, I. E. (2005). A cutting plane method for solving linear generalized disjunctive programming problems. *Computers & Chemical Engineering*, *29*, 1891 – 1913.

Smaïli, F., Angadi, D., Hatch, C., Herbert, O., Vassiliadis, V., & Wilson, D. (1999). Optimization of scheduling of cleaning in heat exchanger networks subject to fouling: Sugar industry case study. *Food and bioproducts processing*, *77*, 159–164.

Sundaramoorthy, A., & Maravelias, C. T. (2011). Computational study of network-based mixed-integer programming approaches for chemical production scheduling. *Industrial & Engineering Chemistry Research*, *50*, 5023–5040.

Velez, S., Merchan, A. F., & Maravelias, C. T. (2015). On the solution of large-scale mixed integer programming scheduling models. *Chemical Engineering Science*, *136*, 139 – 157. Control and Optimization of Smart Plant Operations.

Yenisey, M. M., & Yagmahan, B. (2014). Multi-objective permutation flow shop scheduling problem: Literature review, classification and current trends. *Omega*, *45*, 119 – 135.

Zhang, Q., Cremer, J. L., Grossmann, I. E., Sundaramoorthy, A., & Pinto, J. M. (2016). Risk-based integrated production scheduling and electricity procurement for continuous power-intensive processes. *Computers & Chemical Engineering*, *86*, 90 – 105.

Appendix A. MILP reformulation

Next, an example of a possible way to reformulate the model from GLDP constraints to MILP is given. For instance, (10) is equivalent to

$$(1 - A_{vtpe}) + A_{v(t+1)pe} + E_{v(t+1)(s_{LA})e} + E_{v(t+1)(s_{LB})e} + E_{v(t+1)(s_{PA})e} + E_{v(t+1)(s_{PB})e} \geq 1 \quad \forall v \in \mathcal{V},\ \forall t \in \mathcal{M}\backslash\{t_F\}, \forall p \in \mathcal{P},\ \forall e \in \mathcal{E} \quad \text{(A.1)}$$

and (14) is enforced by the following two MILP constraints:

$$P_{vtpe} \leq U_v(T_{\text{out}}) \cdot A_{vtpe} \quad \forall v \in \mathcal{V}, \forall t \in \mathcal{M}, \forall p \in \mathcal{P}, \forall e \in \mathcal{E} \quad \text{(A.2)}$$

$$P_{vtpe} \geq L_v \cdot A_{vtpe} \quad \forall v \in \mathcal{V}, \forall t \in \mathcal{M}, \forall p \in \mathcal{P}, \forall e \in \mathcal{E} \quad \text{(A.3)}$$

Here **True**=1 and **False**=0 are assumed to convert boolean variables (A_{vtpe}, E_{vtse}) into binary ones.