ONLINE DECISION SUPPORT FOR AN EVAPORATION NETWORK

José Luis Pitarch, Carlos Gómez Palacín, César de Prada
Systems Engineering and Automatic Control Department, EII, Universidad de Valladolid.
C/ Real de Burgos s/n, 47011, Valladolid. {jose.pitarch | carlos.gomez | prada}@autom.uva.es

Marc Kalliski
Department of Biochemical and Chemical Engineering, Technische Universität Dortmund.
Emil-Figge-Str. 70, 44227, Dortmund Germany. Marc.Kalliski@bci.tu-dortmund.de

Christian Jasch
Lenzing Aktiengesellschaft, Werkstraße 2, 4860 Lenzing, Austria. c.jasch@lenzing.com

Abstract

This work presents a decision-support tool to address the model-based optimization approach for online load allocation and scheduling of cleaning operations in an evaporation network. The aim is improving the resource efficiency by supplying the optimal solution for a given production goal. The approach includes the semi-automatic update of evaporator models, which is based on historical data for minimal modelling effort. The structure of the problem is formulated via mixed-integer programming and integrated into the plant supervision systems. Production constraints, concerns about the practical implementation and visualization preferences are also taken into account in the design of the prototypical tool.

Key Words: integration, decision support, visualization, surrogate model, evaporation network, resource efficiency.

1 INTRODUCTION

The fast changes in global market conditions and increasing environmental constraints force the process industry to continuously adapt their operation to keep competitiveness. In this way, an agile plant-wise optimization of continuous and discrete decisions is required to operate as efficient as possible while fitting the new conditions [1].

To face these emerging challenges, plant managers and operators need to be provided with computer-based tools which guide them to balance production and resource consumption [2]. Special attention to the efficient development of plant models needs to be paid, as it is the basis for advanced control and coordination tasks. Moreover, there exist many coding languages and alternatives to implement optimization algorithms into software modules but, in the end, these tools must be integrated into the information technology (IT) infrastructure of the plants, e.g., via a neutral deployment platform that connects to different IT systems [3].

This paper deals with these issues in the evaporation network of Lenzing AG, a viscose fiber production factory located in Austria. A description of the approach and a prototypical tool for the optimization of the allocation of evaporators to products is presented, with the goal of minimizing the overall specific steam consumption (SSC).

In addition, evaporation plants suffer from performance degradation due to fouling inside the heat exchangers. Hence, maintenance tasks become necessary to recover efficiency, but they involve a cost. Therefore, a suitable scheduling of such cleaning tasks over time for each evaporator arises as an additional problem to the load allocation optimization. This requires the resolution of an economic optimization involving discrete and continuous values.

Our approach considers data driven plant models, able to be updated in a semi-automatic way, and efficient mixed-integer nonlinear programming (MINLP) software to solve the proposed optimizations online. The different optimizations were programmed as modules using MATLAB® and MS Excel®, and then linked to the PI System in the plant.

The paper organizes as follows. Next, a description of the application, system limitations and assumptions are given. The modelling routine is summarized in Section 3. Then, the optimization of the load allocation and the cleaning schedule is formally stated in Section 4. The interfaces design together with the system integration is presented in Section 5. Finally, a summary of the work together with indications for the next steps is given in the last section.

2 APPLICATION CASE

The production of viscose fibers is based on the renewable resource wood. The cellulose contained in the wood is chemically treated and converted to a viscose solution. The key production step is the regeneration of this solution into fibers, introduced in an acid bath. Apart from the solid fibers, this chemical treatment produces sodium sulfate (Na_2SO_4) and water as side products. This degrades the acid bath and, in consequence, the product quality. Therefore, it is necessary to constantly remove water and sodium sulfate from the bath. A network of multiple-effect evaporation plants linked to a crystallization section is used for such a task.

Figure 1 depicts a simplified schema of an evaporation plant, where the main manipulated variables are the recirculation flow F and the product temperature T after the heat exchangers. For a more detailed view of the plant the reader is referred to [4].

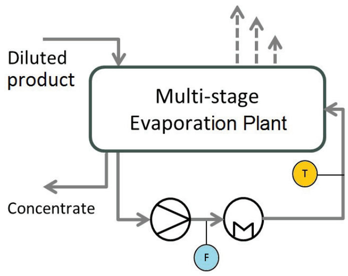

Figure 1. Simplified schema of an evaporation plant with the important control variables F and T.

2.1 NETWORK DESCRIPTION

The evaporation network comprises a total of 23 plants of different nominal capacities. This network needs to process 5 different products, so some plants can serve in more than one product, but only a single product at a time. The changeover from one product to another gets a cost and requires time.

There are several factors which affect the efficiency:

- **Plant type**: [a] Compact with small capacity, [b] 3-stage evaporators and [c] large evaporators.
- **External influences**: Ambient temperature and air humidity affect the cooling towers.
- **Fouling**: Bath impurities settle within the heat exchangers, reducing the heat transfer.
- **Operating point**: For a desired evaporation set point, the control values are not uniquely defined

The resource efficiency indicator (REI) [2] chosen for this process is the SSC, defined for each plant as the ratio of fresh steam consumed per amount of water removed from the product. The task for the plant personnel is to find an optimal allocation of plants to products that ensures the required evaporation rate per product with the lowest SSC. This optimal operation can only be achieved by considering all these significant influences on the resource efficiency in the optimization of the evaporation network. However, the size of the combinatorial problem and the amount of influence factors make the problem very challenging. Indeed, model-based optimization approaches have already improved the efficiency in the operation of an evaporation plant [4], and we still foresee more potential savings in a better coordination of the whole network.

Therefore, computer-aided decision support (DS) tools need to be provided to help operators in this task, so suitable models need to be developed. Furthermore, a suitable computational time is required to provide results in acceptable time, to avoid production delays, and to ensure operator acceptance.

2.2 PLANT MODELS

The evaporator set-up is similar for all plants, but varies in the number of stages and production capacities. The amount of evaporated water depends on the circulating flow F, the product temperature T, and cooling water temperature T_{MK} which, in turn, is limited by the outdoor temperature. A mapping of the evaporation flow achieved for different values in the manipulated variables can be recorded (Figure 2).

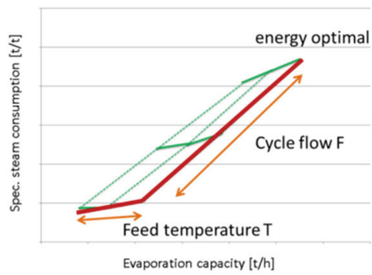

Figure 2. Specific steam consumption VS evaporation set point, obtained for different control values.

Extensive experimental tests shown that the effects of T and F on the SSC, as well as on the evaporation flow, can be described by linear relationships. Additionally, it was observed that the mapping in Figure 2 is shifted in a linear fashion with T_{MK} and the fouling state. Hence, two linear models were proposed to describe the plant behavior, one for the evaporation flow (EF) and another for the SSC, as linear functions of the inputs T, F, T_{MK} and the fouling state K_f (to be estimated):

$$EF = [a_1 \quad a_2 \quad a_3 \quad b_0] \cdot \begin{bmatrix} T \\ F \\ T_{MK} \\ 1 \end{bmatrix} + K_{f1} \tag{1}$$

$$SSC = [c_1 \quad c_2 \quad c_3 \quad d_0] \cdot \begin{bmatrix} T \\ F \\ T_{MK} \\ 1 \end{bmatrix} + K_{f2} \tag{2}$$

Where $\theta = \{a_i, b_0, c_i, d_0\}$ are constant parameters for offline regression and K_{fi} are time dependent ones to be identified online. The absolute steam consumption (ASC) for a plant is computed by multiplication of (1) and (2):

$$ASC = EF \cdot SCC \tag{3}$$

In this way, given a fouling state K_{f1} and a cooling water temperature T_{MK}, the maximum and minimum evaporation capacities, denoted by $\overline{EC}$ and $\underline{EC}$ respectively, for each plant can be computed by (1) with the acceptable operating ranges for F and T:

$$\begin{aligned} \overline{EC} &= EF\left(\overline{T}, \overline{F}, T_{MK}, K_{f1}\right) \\ \underline{EC} &= EF\left(\underline{T}, \underline{F}, T_{MK}, K_{f1}\right) \end{aligned} \tag{4}$$

The goal for the selection of the operating point is minimal specific steam consumption fulfilling the evaporation demand (red front in Figure 2). A self-optimizing controller (SOC) was implemented to ensure that operation always lies in this region. The controller maximizes the product temperature T to its upper limit and adjusts the circulating flow F to achieve the required evaporation flow [4]. Thanks to this optimal operation pattern, we are able to compute the control values corresponding to the red boundary in Figure 2 given a desired EF and an estimated state of fouling K_f: indeed note that the T is set to its upper bound and T_{MK} is set to the lower one achievable by the cooling tower, so F can be computed directly from (1) and, thus, the SSC from (2).

3 MODELLING ROUTINE

The model identification task is implemented in MATLAB and comprises a data treatment to remove inconsistent measurements, identifies step changes and performs an iterative fitting of the model parameters. The required data form the evaporators are obtained from the PI system via an OPC-connection and additional information such as the time window for identification, tag labels to the measurements in the historian, minimal number of changes in the EF, acceptable noise band in stationary operation, largest transition period during step change, or the time window for validation is provided by the operator with standardized Excel sheets. Finally, the quality of the model is assessed by a comparison of the model predictions with the measured EF and SSC.

3.1 DETECTION OF STEP CHANGES

For the modeling of the stationary part, the contribution of fouling must be removed from the training set. This is achieved during the model fitting process but requires data from operational points that are subject to the same degree of fouling. Thus, the tool identifies changes in the EF, because operation point varies enough to identify the parameters and we can assume that the fouling state does not vary significantly in one day.

In that way, the data is scanned for step changes as shown in Figure 3, providing intervals ***a*** and ***b***. Step changes are identified in the case that: data in ***a*** is at steady state (within a threshold); step change is larger than the threshold ***c*** and; the transition between the two steady states is completed within interval ***b***. Each of the identified step is recorded and translated into a data pair by averaging the measured values before and after the change. Of course, enough changes provoked by T, F and T_{MK} are required for a reliable identification of the parameters in (1) and (2).

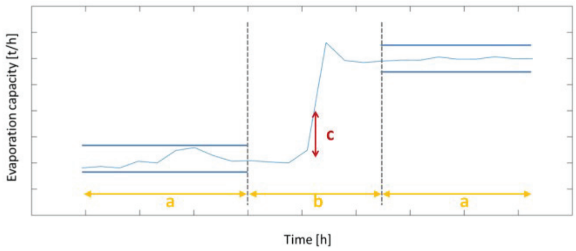

Figure 3. Stationary operation before and after the step (blue bounds), minimal step height (red arrow), maximal transition interval (dotted lines).

3.2 PARAMETER ESTIMATION

Based on the assumption that the fouling is different from step to step but remains constant during the step itself, parameters K_{f1}, K_{f2} can be estimated by comparison of the model predictions $(\widehat{SSC}, \widehat{EC})$ with the actual values (*SSC, EF*). The resulting fouling factor is subsequently used for both operating points that are considered (before and after step changes).

Hence, an iterative LS optimization over the overall data set arises, which fist yields an intermediate set of steady-state parameters θ, used afterwards to calculate new values for K_{fi}, before solving an updated LS optimization to find the next model generation. These iterations continue until the residue J (objective function) does not improve any more.

$$J(\theta, u) = \sum_{i=1}^{n} \frac{\left(EF_i - \widehat{EF}_i\right)^2}{\sigma_{1i}} + \frac{\left(SSC_i - \widehat{SSC}_i\right)^2}{\sigma_{2i}} \tag{5}$$

Here u is the set of values of the manipulated variables T, F and T_{MK}, n is the number of identified step changes and σ_1, σ_2 are normalizing factors. The identification procedure is summarized in Algorithm 1.

Algorithm 1. Parameter estimation for evaporation plants.

1. Provide an initial guess for θ and set $k = 0$.
2. Average the measured steady-state values for the SSC and EF before and after the step change.
3. Adjust fouling factors K_{fi} by comparing the SSC and EF from Step 2 with the model prediction.
4. Minimize (5) with θ as decision variables to find the best fit for all step changes.
5. If $J_k < J_{k-1}$ set $k = k + 1$ and go to Step 3, else the algorithm stops.

In practice, for a reasonable initial guess, the model parameters θ converge after a few iteration steps.

3.3 VALIDATION

The validation step in the modelling routine is performed to assess the quality of the model on the basis of an independent set of step changes that has also been obtained according to Section 3.1. For each identified step change, the fouling factor is also adjusted in the models to match the average values of recorded data before the load change. The model with the updated fouling factors is then used to simulate the plant for the same inputs applied during change. The resulting absolute error is then normalized with the height of the step change in the ASC, to yield a relative measure of the model error.

The observed relative errors are typically below 10%. These values were acceptable, since the network optimization is performed periodically and mismodelling is reduced from run to run by an online estimation of the fouling state. In some cases relative errors of up to 30% have been observed due to a poor choice of load changes (e.g. non-stationary operation falsely identified as steady state). A manual selection of step changes, choosing an alternative modeling horizon, or an adjustment of the modeling settings was sufficient to improve the model fit.

4 NETWORK OPTIMIZATION

The objective is the minimization of the ASC for the entire network, given a desired evaporation demand. The overall ASC is calculated as the sum of (3) for all evaporators. Two main factors which affect the ASC are object of optimization: the load allocation and the cleaning policy.

4.1 OPTIMAL ALLOCATION

First, given a set of $p \in \mathcal{P}$ products to be processed in $e \in \mathcal{E}$ evaporation plants, the problem is to allocate plants to products and then distribute the required total demand per product SP_p in a way that the overall ASC in the network is minimized. Two sets of decision variables are defined for this aim:

- X_{ep}: Binary variables which link the product p to the plant e.
- EF_{ep}: Real variables defining the evaporation flow to be achieved in a plant e processing the product p.

Now, recalling (4), assuming that the fouling state K_f for each plant will be estimated, T_{MK} is measured and controlled, a set of maximum and minimum capacities for each plant $e \in \mathcal{E}$ is provided. Moreover, following the optimal control pattern explained in Section 2.2 of setting T for each plant to its upper limit $\overline{T}$, from (1)-(3) we get:

$$F_{ep} = \frac{EF_{ep} - a_1\overline{T} - a_3 T_{MKe} - b_0 - K_{f1e}}{a_2} \tag{6}$$

$$\begin{aligned} ASC_{ep} = \frac{c_2}{a_2} EF_{ep}^2 + \Bigg[\left(c_1 - \frac{a_1}{a_2}\right)\overline{T} + \left(c_3 - \frac{a_3}{a_2}\right) \\ T_{MK} - \frac{b_0}{a_2} + d_0 - \frac{K_{f1e}}{a_2} + K_{f2e} \Bigg] EF_{ep} \end{aligned} \tag{7}$$

Thus, feeding this information, the optimal allocation of products to plants is found by solving the mixed integer quadratic programming problem below:

$$\min_{X_{ep}, EF_{ep}} \quad J := \sum_{e\in\mathcal{E}} \sum_{p\in\mathcal{P}} ASC_{ep} \quad \text{s.t.:} \tag{8}$$

$$\sum_{p\in\mathcal{P}} X_{ep} \leq 1 \quad \forall e \in \mathcal{E} \tag{9}$$

$$\sum_{e\in\mathcal{E}} EF_{ep} \geq SP_p \quad \forall p \in \mathcal{P} \tag{10}$$

$$EF_{ep} \leq \overline{EC}_e \cdot X_{ep} \quad \forall e \in \mathcal{E}, \forall p \in \mathcal{P} \tag{11}$$

$$EF_{ep} \geq \underline{EC}_e \cdot X_{ep} \quad \forall e \in \mathcal{E}, \forall p \in \mathcal{P} \tag{12}$$

$$X_{ep} = 0 \quad (e,p) \notin \mathcal{N} \tag{13}$$

Where X_{ep} is constrained in (13) to find feasible solutions within the set $\mathcal{N}$ of allowed connections between plants and products.

4.2 CLEANING SCHEDULE

A complementary optimization is proposed to deal with the issue of fouling, which takes advantage of the already developed decision support: once optimal evaporation set points are computed for each plant, the idea is to suggest the next cleaning cycle by balancing the costs of operation over time with the cleaning costs in an optimal fashion.

This task requires models for the evolution of the fouling over time. Extensive experimental tests have been performed measuring the SSC in the evaporators running at reference operation points between consecutive cleaning cycles. This allows isolating the effect of fouling on the SSC increase, hence measurements are comparable. In this way, approximate linear evolutions of the fouling behavior could be identified by regression, see Figure 4.

Thus, the fouling contribution K_f in (1)-(2) becomes:

$$K_f(t) = K_{f0} + \alpha \cdot t \tag{14}$$

Where t stands for the time (in days) that a plant is in operation, K_{f0} is the initial or current estimation of the fouling state, and α is the slope of the linear model. In this way, predictions of the future SSC (hence costs) can be computed given a desired EF.

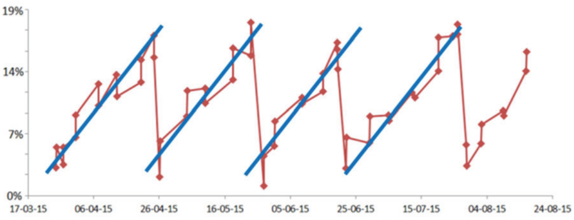

Figure 4. Measured evolution of the SSC (red) and output of the regression model (blue).

In order to lump resources of different nature (steam, manpower, cleaning products, etc.) in a single efficiency indicator, an aggregation based on currency is used. Hence, using prices and costs for utilities, the *Normalized Average Cost per Time* (NAC) is defined as an REI, and indicates the unitary cost (€/d) incurred to operate a plant between two consecutive cleaning tasks (operation cycle):

$$NAC := \Big(\sum_{t=0}^{t_f} ASC(t) \cdot P_{Th} + \Delta t_{cl} \cdot P_{Manh} + \mathrm{WstWat} \cdot P_{m^3} + \Delta t_{cl} \cdot ASC(t_f) \cdot 1.1 P_{Th}\Big)/t_f \quad (15)$$

Here t_f is the suggested future day to perform the cleaning operation, Δt_{cl} is the time required to complete a cleaning operation, and P_{Th}, P_{Manh} and P_{m^3} are the costs of the fresh steam, manpower and waste water. Note that, once the EF to each plant is set from (8)-(13), the $ASC(t)$ is computed via (7) and (14).

Note also that when an evaporator is stopped for cleaning, its load must be assumed by others, so an approximate cost factor of a 10% increase over the nominal operation cost is added in (15).

The NAC is to be minimized with respect to t_f for each evaporator to compute a periodic "individually optimal" cleaning policy, which attempts to be a "nearly optimal" one for the whole network. However an issue appears in using (15) as objective function: the cost of operation is a discrete sum which gets t_f terms, being t_f unknown a priori, as it is decision variable. To express this cost in a suitable way, we make use of the formula found by Gauss in the late 1700's for this type of arithmetic series [6]:

$$\kappa \cdot \left(1 + 2 + 3 + \cdots + T_f\right) = \kappa \cdot \frac{\left(1 + T_f\right) \cdot T_f}{2} \quad (16)$$

Moreover, there are two types of cleaning tasks, denoted by B (big) and S (small), reaching different recoveries K_{f0}, booking different times Δt_{cl} and using more or less waste water. Thus, each task will get different *fixed* costs in the NAC (15) so the optimizer must choose which option minimizes the costs.

Thus, the proposed economic optimization to predict the optimal cleaning policy for one plant reads:

$$\min_{t_f, c} \quad J := c \cdot NAC|_B + (1 - c) \cdot NAC|_S \quad (17)$$

$$\mathrm{s.t.:} \quad 0 \le c \le 1; \;\; t_f > 0 \quad (18)$$

Here notation $NAC|_B$ stands for (15) evaluated with values Δt_{cl}, K_{f0} and WstWat corresponding to a big cleaning operation ($NAC|_S$ is analogous for a small cleaning). Note that this optimization to choose between discrete alternatives can be handled via NLP because (17) is monotonous w.r.t. c, so its minimum is located in an extreme, either $c = 0$ or $c = 1$. In this way, the best cleaning (big or small) is chosen.

5 SYSTEM INTEGRATION

The modelling as well as the load allocation modules were implemented using MATLAB®. The cleaning schedule optimization was coded in directly in MS Excel. These choices are justified since the required licenses and experience of the engineering department at Lenzing AG are available for the sustainable maintenance of the decision-support solution.

Figure 5 depicts a schema of the real-time optimization (RTO) implemented to cope with the load allocation task, which is executed each 30 min. The user dashboard is included as a Process Book in the PI system. It shows the results and allows to manually trigger the optimization in case of significant changes in the evaporation SP_p. After the activation, the static information (network information, model parameters, etc) is read from an Excel interface. Production constraints change dynamically, so they are either directly supplied by the data historian or inferred from measurements. Then, an update of the fouling parameters K_{f1} and K_{f2} is performed and saved to a file. The fouling parameters of the inactive equipment are not updated in the file.

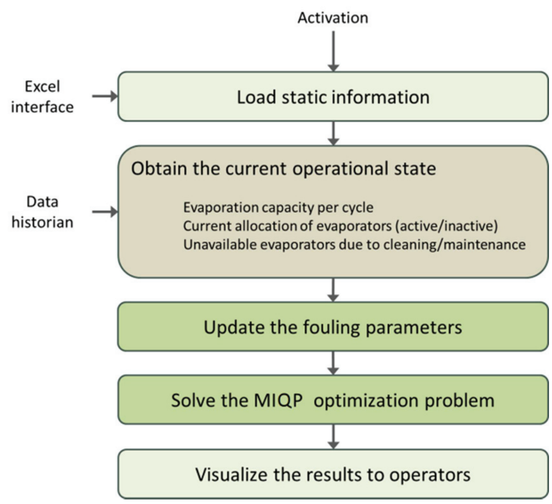

Figure 5. RTO concept for the evaporation network.

Execution of the optimization results in the allocations of evaporators to products, the load distribution, the SSC and ASC for each plant. This information is written back to the system using special PI-Tags which are displayed to the operators via dashboard application in the PI Process Book. Hence, the operators should adjust the evaporator loads and allocations accordingly.

However, the optimization of the load distribution may sometimes result in infeasibility, since approximate models (1) might slightly underestimate the evaporation capacity for some plants. This situation might lead to optimization problems that are overall infeasible based on the model prediction, even though the real plant is capable to fulfil the desired evaporation flow.

5.1 HANDLING INFEASIBILITIES

It is impossible to provide reliable decision support to the operators without a feasible solution from the optimizer, because hard constraints might be violated. To avoid these situations, a feasibility check is performed first, that evaluates whether the currently measured evaporation flow for each evaporator can be achieved with the models under the same external constraints (weather, cooling water temperatures and network availability). Then, for the identified infeasible plants, the MIQP constraints (11) are soften with slack variables $S_e \in \mathbb{R}^+$ as follows:

$$EF_{ep} - S_e \leq \overline{EC}_e \cdot X_{ep} \;\; \forall e \in \mathcal{S}, \forall p \in \mathcal{P} \qquad (19)$$

Where $\mathcal{S}$ is the set of plants which are identified infeasible after the feasibility check. Then, the sum over all slack variables is included as a penalty term into the objective function (8) as:

$$\min_{X_{ep}, EF_{ep}, S_e} \quad J := \sum_{e \in \mathcal{E}} \sum_{p \in \mathcal{P}} ASC_{ep} + M \cdot \sum_{e \in \mathcal{S}} S_e \qquad (20)$$

The weight M is roughly chosen to be greater than the largest possible value of (8), i.e., without the contribution of the slack variables. Thus, the solver will only provide the absolutely necessary amount of constraint violation. Note that the network operation will not result in constraint violations on the control inputs, since the SOC is in place for each plant, and infeasibility is only a result of a plant-model mismatch. Moreover, a warning can be passed to the supervisor (plant engineer). Thus, depending on the severity of the plant-model mismatch, corrective actions can be taken, e.g., a model update according to the procedure in Section 3.

The final optimization problem can be coded in MATLAB and solved with an MILP solver via successive linear approximations [7], or directly with a MINLP solver like BONMIN [8] via the open source OPTI-Toolbox, although this option might be less computationally efficient.

5.2 DECISION-SUPPORT INTERFACES

The visualization interface is adapted to the already existing concept that was designed to give an overview of the evaporation process during production. On the one hand, the operators are supplied with the dashboard depicted in Figure 6 that shows the computed optimal solution for the current time. The vertical columns represent the 23 plants in the network and the rows represent 5 products. Light grey boxes are the allocation possibilities of plants to products. If a plant is assigned to one of these possible combinations, the box becomes green. Plants that are currently assigned to a product but are not in operation (under maintenance or cleaning) are shown by red tiles.

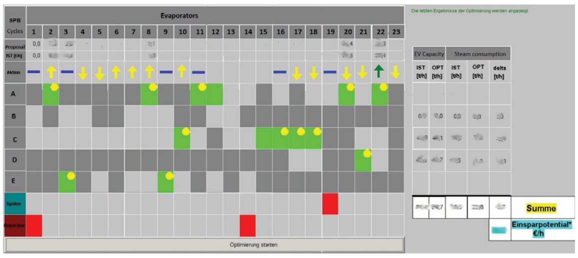

Figure 6. Interface of the prototypical tool for online optimization of the evaporation network.

The allocation plan according to the optimization results is indicated with yellow (partial load) or green circles (full load) at the corresponding position in the matrix representation. The optimal load distribution to plants is directly given next to the current value at the top of the matrix. Small pictograms show the necessary direction of the change in evaporation set points. On the right hand side, the current and optimal values for a total product evaporation flow and the ACS are listed. Finally, the predicted network-wide savings potential is shown in € saved per hour, in order to create an incentive for the operators to apply the predicted evaporation set points to plants.

On the other hand, the cleaning prediction module of Section 4.2 has been implemented by an Excel-based tool, partially coded in Visual Basic and using the OpenSolver [9] add-on, whose current version includes BONMIN as optimization engine. This tool complements the one above, by receiving the load allocation for each evaporator as input data.

The interface is formed by several sheets: one for each plant and a general overview of the network. In each plant sheet there is a set of values to be set: duration of cleaning tasks, costs of resources, energy prices, model parameters and control set points T, T_{MK} (see Figure 7). The tool provides a button in each sheet to trigger the optimization (17)-(18), displaying then when the evaporator should be cleaned and which type of operation is best, as well as the cost components and current value of the NAC for the suggested policy. Moreover, the tool serves also as a simulator for what-if analysis, because the user is allowed to manually set the next cleaning day and the type of cleaning. In this way, the tool informs the operator about the potential losses in €/d incurred with respect to the optimally computed NAC, encouraging him/her to apply the suggestions.

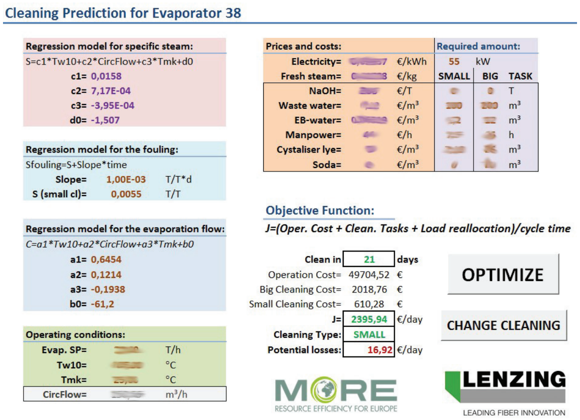

Figure 7. Interface of the prototypical tool for the improved scheduling of cleaning operations.

6 CONCLUSIONS & OUTLOOK

The modelling, optimization and visualization concepts presented in this paper support the operators to take better decisions in real time to improve the network operation. The modelling tool executes an automatized model update based on historical data and user inputs. The resulting models are incorporated in the RTO scheme that solves a MIQP problem according to the current production constraints and the plants fouling states. The results are visualized in the daily production environment, including predictions of the potential monetary savings, incentivizing thus the operators to apply the recommendations.

Models for long-term fouling effects were identified by extensive experimentation, to be then used in an economic optimization. The incorporation of such functionality allows finding the best cleaning policy for each plant. This provides additional benefits in terms of energy and costs associated to the cleaning.

The developed DS tools are currently under evaluation at Lenzing AG: the implementation into the existing systems and operational policies is performed step by step to get experience in live testing and to ensure acceptability from the plant personnel About a year of normal operation is required to assess the impact, but preliminary tests with historical data revealed around 10% ASC potential savings.

Further improvements in the modelling approach are expected if different plant models are used for the summer and winter periods. If the impact assessment shows sufficient improvement, other heuristics and decompositions of the optimization problems will be evaluated to take into account uncertainty in model parameters and/or external factors.

Acknowledgement

This research is funded by the European Union's Horizon 2020 program, under grant nº 723575, and by the MINECO/FEDER (DPI2015-70975-P).

References

[1] S. Engell and I. Harjunkoski, "Optimal operation: Scheduling, advanced control and their integration," *Computers & Chemical Engineering,* pp. 121-133, 2012.

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

[3] LeiKon, "D4.1 Requirement specification for the integrated deployment platform," Outcomes of the MORE Project, 2014.

[4] J.L. Pitarch, C.G. Palacín, C. de Prada, B. Voglauer and G. Seyfriedsberger, «Optimisation of the Resource Efficiency in an Industrial Evaporation System,» *Journal of Process Control,* vol. 56, pp. 1-12, 2017.

[5] M. Kalliski, B. Beisheim, D. Krahè, U. Enste, S. Krämer and S. Engell, "Real-time resource efficiency indicators," *atp edition - Autom. Praxis,* vol. 58, pp. 64-71, 2016.

[6] D.M. Burton, Elementary Number Theory, Boston: MA: Allyn and Bacon, 1989, pp. 80-81.

[7] C. Bliek, P. Bonami and A. Lodi, "Solving Mixed Integer Quadratic Programming problems with IBM-CPLEX: a progress report," in *Proc. of the 26th RAMP Symposium*, Tokyo, 2014.

[8] P. Bonami, L. T. Biegler, A. R. Conn, G. Cornuejols, I. E. Grossmann, C. D. Laird, J. Lee, A. Lodi, F. Margot and A. Waechter, «An Algorithmic Framework for Convex Mixed Integer Nonlinear Programs,» *Discrete Optimization,* vol. 5, nº 2, pp. 186-204, 2008.

[9] A. Mason, "OpenSolver – An Open Source Add-in to Solve Linear and Integer Progammes," in *Operations Research Proceedings 2011*, D. Klatte, H. Lathi and K. Schmedders, Eds., Springer Berlin Heidelberg, 2012, pp. 401-406, http://opensolver.org.