

# Optimal distributed load allocation and resource utilisation in evaporation plants

Maria P. Marcos<sup>a,b,\*</sup>, Jose L. Pitarch<sup>a</sup>, Christian Jasch<sup>c</sup> and Cesar de Prada<sup>a,b</sup>

<sup>a</sup>*Systems Engineering and Automatic Control Department, EII, Universidad de Valladolid. C/ Real de Burgos S/N, Valladolid, 47011, Spain.*

<sup>b</sup>*Sustainable Processes Institute (IPS), Universidad de Valladolid.*

<sup>c</sup>*Lenzing AG. Werkstraße 2, Lenzing, 4860, Austria*

\* [maria.marcos@autom.uva.es](mailto:maria.marcos@autom.uva.es)

## Abstract

This work aims to optimise the operation of an evaporation network with shared resources in real time. The goal is minimising the resource utilisation (live steam and cooling water) while satisfying a set of operational constraints. Hence, problems of optimal load allocation of feeds to plants and cooling-water distribution among them arise. The work bases on plants surrogate models, experimentally obtained, and analyses different formulation alternatives of the optimisation from the practical point of view: centralised vs distributed approaches. In particular for the distributed approach, we propose a problem decomposition which allows us to solve the problem in two iterative ways: 1) as two independent optimisations or 2) via price-coordination schemes.

**Keywords:** decomposition methods, RTO, evaporation network, resource efficiency

## 1. Description of the case study

Manufacture of viscose fibers in Lenzing AG bases on a continuous spinning process where cellulose pulp transforms into very thin threads. This process performs in an acid bath (“spinbath” from now on) which is progressively degraded. Consequently, it must be continuously recovered. Therefore, a side evaporation process is performed in parallel to remove part of water. For this task, an evaporation network of 15 plants is available to serve in 5 different types of spinbath, where a single plant is formed by several equipment. See Pitarch et al. (2017b) for a deeper description.

Plants efficiency depends on several factors: the evaporation flow removed from the spinbath (plant load)  $EC$ , the spinbath temperature  $T$  and circulating flow  $F_s$ , the fouling state in the heat exchangers and the performance of the cooling system (see Figure 1a). In particular, the more cooling water (shared resource) is provided to the surface condenser, the less live steam consumption (utility) is needed to achieve the evaporation setpoint.

Denote the *specific steam consumption* ( $SSC$ ) by the live steam usage per unit of evaporated water (Pitarch et al., 2017a). Then, the *absolute steam consumption* ( $ASC$ ) in each plant is computed by  $ASC = SSC \cdot EC$ .

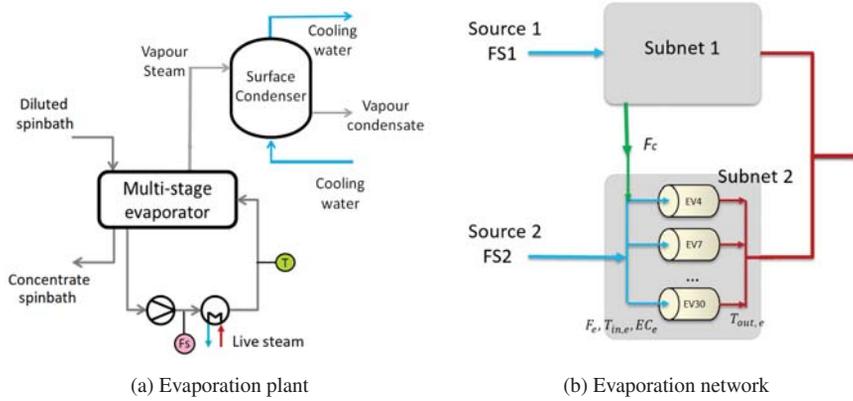


Figure 1: Systems simplified schemes.

Plant surrogate models were developed to estimate the *SSC* in Kalliski et al. (2019), which, after straightforward manipulations, is found to depend on *EC* and on the cooling capacity of the surface condenser  $C_{pow}$ , as shown in (1). The cooling system performance is also modelled experimentally: the outlet cooling-water temperature of the condensers  $T_{out}$  is estimated by a polynomial function up to degree 3 on the cooling-water flow  $F_w$  and affine in its inlet temperature  $T_{in}$ , as proposed in (Marcos et al., 2018):

$$SSC = \beta_2 \cdot EC + \beta_1 \cdot C_{pow} + \beta_0; \quad C_{pow} = 4.18 \cdot F_w (T_{out} - T_{in}) / 3600 \quad (1)$$

$$T_{out} = \alpha_0 + \alpha_1 F_w + \alpha_2 F_w^2 + \alpha_3 F_w^3 + T_{in} \quad (2)$$

Here,  $\theta = \{\beta_0, \beta_1, \beta_2, \alpha_0, \alpha_1, \alpha_2, \alpha_3\}$  are experimental regression parameters.

According to the physical layout of the evaporation network, the plants can be grouped in two subnets, each one directly supplied by an independent water source. The water distribution among plants is done in parallel and, additionally, the exceeding water in Subnet 1 ( $F_c$ ) can go to Subnet 2 but not backwards (see Figure 1b). Further details on the evaporation network can be found in Palacín et al. (2018); Marcos et al. (2018).

## 2. Problem formulation

The objective is to allocate feeds to plants and distribute the cooling-water in the network at minimum cost. This cost is a trade-off between the usage of resources: live steam and cooling water, times their respective “prices” ( $P_s$  and  $P_w$ ). Three sets of decision variables are defined:  $X_{e,p} \in \{0, 1\}$  to link product  $p$  to plant  $e$ ,  $EC_{e,p} \in \mathbb{R}$  defining the evaporation flow of  $p$  to be achieved in plant  $e$ , and  $F_{w,e} \in \mathbb{R}$  defining the water sent to plant  $e$ . Further, an additional variable  $F_c \in \mathbb{R}$  indicates the flow of water sent from Subnet 1 (SN1) to Subnet 2 (SN2). Hence, for a given set of  $e \in \mathcal{E}$  evaporation plants that have to process  $p \in \mathcal{P}$  spinbaths, the centralised formulation of the optimisation problem is:

$$\min_{X_{e,p}, EC_{e,p}, F_{w,e}, F_c} J = \sum_e^{\mathcal{E}} \left( \sum_p^{\mathcal{P}} ASC_{e,p} \cdot P_s + F_{w,e} \cdot P_w \right) \quad (3a)$$

$$\text{s. t.:} \quad \sum_p^{\mathcal{P}} X_{e,p} \leq 1 \quad \forall e \in \mathcal{E}; \quad X_{e,p} = 0 \quad (e, p) \in \mathcal{N} \quad (3b)$$

$$\sum_e^{\mathcal{E}} EC_{e,p} \geq SP_p \quad \forall p \in \mathcal{P} \quad (3c)$$

$$\sum_e^{\mathcal{E}_{SN1}} F_{w\ e} + F_c \leq F_{S1}; \quad \sum_e^{\mathcal{E}_{SN2}} F_{w\ e} - F_c \leq F_{S2}; \quad F_c \geq 0 \quad (3d)$$

$$\underline{EC}_{e,p} \cdot X_{e,p} \leq EC_{e,p} \leq \overline{EC}_{e,p} \cdot X_{e,p} \quad \forall e \in \mathcal{E}, \forall p \in \mathcal{P} \quad (3e)$$

$$\sum_p^{\mathcal{P}} X_{e,p} \cdot \underline{F}_{w\ e} \leq F_{w\ e} \leq \overline{F}_{w\ e} \cdot \sum_p^{\mathcal{P}} X_{e,p} \quad \forall e \in \mathcal{E} \quad (3f)$$

$$T_{out\ e} \leq T_{max} + (1 - \sum_p^{\mathcal{P}} X_{e,p}) \cdot M \quad \forall e \in \mathcal{E} \quad (3g)$$

Where the sets  $\mathcal{E}_{SN1}$  and  $\mathcal{E}_{SN2}$  ( $\mathcal{E} = \mathcal{E}_{SN1} \cup \mathcal{E}_{SN2}$ ) include the plants belonging to SN1 and SN2 respectively, the set  $\mathcal{N}$  represents connections between some plants and spinbaths that are forbidden due to the factory layout, and M is a big enough number.

The problem constraints are: (3b) each plant can only serve in one spinbath loop at a time; (3c) total evaporation demands per spinbath ( $SP_p$ ) have to be fulfilled; (3d) total water consumption in each subnet has to be lower than the available at the sources ( $F_s$ ), considering that exceeding water can go from SN1 to SN2 but not backwards; (3e)-(3f) evaporation and water flows are bounded and; (3g) outlet water temperatures have to be lower than a limit ( $T_{max}$ ), stated by the environmental regulation.

The presence of discrete and continuous variables as well as the nonlinear dependency of (3a) and (2) on them, makes (3) become an MINLP problem.

### 3. Problem decomposition

In order to avoid the issues that commonly arise with MINLP (slow convergence, high computational demands, etc.), we propose a suitable decomposition of (3) in two local optimisation subproblems, according to the physical layout of the evaporation network. In fact, the overall equipment can be grouped in two sets or networks: 1) the spinbath allocation one, composed by the evaporation plants and the spinbath loops themselves, and 2) the water distribution from the sources to the respective plant surface condensers. Hence, a first local problem can handle the plants load allocation whereas a second one optimises the water distribution. By this decomposition, only the magnitudes  $EC$  and  $C_{pow}$  are shared between problems 1) and 2).

1. Load-allocation problem:

$$\min_{X_{e,p}, EC_{e,p}} J_1 = \sum_e^{\mathcal{E}} \sum_p^{\mathcal{P}} SSC_{e,p} \cdot EC_{e,p} \cdot P_s \quad \text{s. t.: (3b), (3c), (3e)} \quad (4)$$

In this way, (4) is a mixed integer quadratic programming (MIQP) problem, provided that  $C_{pow}$  in (1) is assumed known.

2. Cooling-water distribution problem:

$$\min_{F_{w\ e}, F_c} J_2 = \sum_e^{\mathcal{E}} (SSC_e \cdot \sum_p^{\mathcal{P}} EC_{e,p} \cdot P_s + F_{w\ e} \cdot P_w) \quad \text{s. t.: (3d), (3f), (3g)} \quad (5)$$

Thus, assuming that the load allocation ( $EC_{e,p}$ ) is known, (5) is an non linear programming (NLP) problem.

Now, we can solve the original problem in two iterative ways: sequentially as two fully independent optimisations, or in a price-coordination fashion. See next section.

#### 4. Implementation

The above formulated optimisation problems (3), (4) and (5) are coded in Pyomo (Hart et al., 2017). The centralised version (3) uses Bonmin as MINLP solver (Bonami et al., 2008), whereas Gurobi Optimization (2018) is used to solve the MIQP (4) and Ipopt (Wächter and Biegler, 2006) to solve the NLP (5). In order to implement the decomposition approach in an iterative fashion, we propose the following.

##### 4.1. Sequential approach

Take  $C_{\text{pow } e}$  as known (computed by (1)-(2) and from given flows  $F_{w e}$ ) and solve (4). Then, from the obtained solution, take the values of  $EC_{e,p}$  and solve (5), getting a new set of values  $F_{w e}$  to compute the  $C_{\text{pow } e}$  and so on. The procedure is formalised in:

---

##### Algorithm 1 Sequential optimisation

---

- 1: Compute the  $C_{\text{pow } e}$  from current (measured)  $F_{w e}, T_{\text{out } e}, T_{\text{in}}$  and set  $\varepsilon = 1$
  - 2: **while**  $\varepsilon > 0.001$  **do**
  - 3:     Solve (4) and save the computed allocation  $EC_{e,p}$
  - 4:     Solve (5) using the  $EC_{e,p}$  from Step 3 and get a water distribution  $F_{w e}, T_{\text{out } e}$
  - 5:     Compute  $\varepsilon = \sum_e^{\mathcal{E}} \|C_{\text{pow } e} - 4.18F_{w e} \cdot (T_{\text{out } e} - T_{\text{in}})/3600\|_2^2$
  - 6:     Update  $C_{\text{pow } e} = 4.18F_{w e} \cdot (T_{\text{out } e} - T_{\text{in}})/3600$  with values from Step 4
- 

##### 4.2. Distributed approach

In a distributed fashion, we need to decouple problems (4) and (5) to allow a “true” parallel implementation. Thus, we make use of the well-known Lagrangean or price-driven decomposition approaches (Cheng et al., 2007).

Indeed, now  $C_{\text{pow } e}$  will be also *local* decision variables in (4) and  $EC_e$  will be so in (5) as well<sup>1</sup>. Then, we have to add a *coordination* layer to progressively force both sets of magnitudes to be equal in both local optimisations (shared constraints). As usual in Lagrangean decomposition, these constraints will be added as a penalty in the respective objective functions:

$$J_1 = \sum_e^{\mathcal{E}} \left( \sum_p^{\mathcal{P}} SSC_{e,p} \cdot EC_{e,p} \cdot P_s + \frac{1}{2} (p_{11e} (C_{\text{pow } e} - \mathbf{C}_{\text{pow } e}^*)^2 + p_{12e} (\sum_p^{\mathcal{P}} EC_{e,p} - \mathbf{EC}_e^*)^2) \right) \quad (6)$$

$$J_2 = \sum_e^{\mathcal{E}} \left( SSC_e \cdot EC_e \cdot P_s + F_{w e} \cdot P_w + \frac{1}{2} (p_{21e} (C_{\text{pow } e} - \mathbf{C}_{\text{pow } e}^*)^2 + p_{22e} (EC_e - \mathbf{EC}_e^*)^2) \right) \quad (7)$$

---

<sup>1</sup>Note that variables  $EC_e$  equivalently replace  $\sum_p^{\mathcal{P}} EC_{e,p}$  in (5) to reduce the problem size.

Where  $p_{ije} \in \mathbb{R}^+$  are the *shadow prices* for “resource” utilisation and  $\mathbf{R}^* := \{\mathbf{C}_{\text{pow } e}^*, \mathbf{EC}_e^*\}$  are the reference values, computed somehow from the local solutions of both optimisations in the previous iteration. In this way, both problems will be managed by a coordinator which, in each iteration  $k$ , will receive the values got for the shared variables  $R_e := \{C_{\text{pow } e}, EC_e\}$ , and it will update the prices as well as the reference values for the next iteration  $k + 1$  according to the following rules:

$$p_{ije}^{[k+1]} = p_{ije}^{[k]} + (R_{ije}^{[k]} - \mathbf{R}^{*[k]})^2 \cdot \mu^{[k]}, \quad R_{ije} \in R, \mathbf{R}^* \in R \quad (8)$$

$$\mu^{[k+1]} = \mu^{[k]} \cdot \lambda; \quad \lambda > 1 \quad (9)$$

$$\mathbf{C}_{\text{pow } e}^{*[k+1]} = C_{\text{pow } e}^{[k]}(2); \quad \mathbf{EC}_e^{*[k+1]} = EC_e^{[k]}(1) \quad (10)$$

Where notation  $R_e^{[k]}(i)$  denotes the values of variables  $R_e$ , solution of subproblem  $i$  at iteration  $k$ . Progressive hedging (Rockafellar and Wets, 1991) is used in (9) to update the factor  $\mu$  in each iteration, via the user-defined parameter  $\lambda$ . The procedure to solve the distributed optimisation is summarized below.

---

**Algorithm 2** Distributed optimisation

---

- 1: Set  $\mathbf{R}^*$  to the current values measured from the plants and  $p_{ije} = \varepsilon = 0.1$
  - 2: **while**  $\varepsilon > 0.001$  **do**
  - 3:     Solve subproblems (4) and (5), and get new values for  $R_{ije}$
  - 4:     Compute  $\varepsilon = \|R_{ije} - \mathbf{R}^*\|_2^2$
  - 5:     Update prices  $p_{ije}$  and references  $\mathbf{R}^*$  with (8)-(10)
- 

Of course, due to the non-convex nature of the problem, global optimality is not guaranteed in either of the implementations, centralised or distributed.

## 5. Results and discussion

For the sake of comparison/analysis, we have solved the problem from a particular network situation following the three presented ways. Table 1 shows a brief summary.

Table 1: Optimised cost and computational effort for the 3 approaches.

	Total CPU Time (s)	Optimal cost (€/h)	Iterations
Centralised	29.64	962.82	-
Sequential	0.89	964.81	3
Distributed	1.73	964.33	6

We can conclude that the three approaches are able to reach nearly the same (local) optimal solution<sup>2</sup>. However, as expected, the centralised problem elapses more time to reach the solution. This is not a major issue in this application (the execution period for real-time optimisation is about 30 min.) but, it will be so if the problem grows by considering fouling predictions over time or by including more parts of the factory (the water network is larger and also serves to other processes, and there is a heat-recovery network which also interacts with the evaporation one).

<sup>2</sup>Values  $p_{ije}^{[0]} = 0.1$ ,  $\mu^{[0]} = 1$  and  $\lambda = 1000$  were set for the distributed approach.

## 6. Conclusions

In this work we addressed a medium-scale real-time optimisation problem ( $\sim 200$  variables, it depends on the approach) on resource efficiency in an industrial evaporation network with shared resources. The proposed approaches will support the operators to take better decisions in real time, thus, improving the operation of the overall network: product-plant allocation and water distribution to the cooling systems.

The plant models are incorporated in three different optimisation schemes, analysing the convenience of the decomposition approaches vs the centralised one. In this case study, the three approaches solve the problem in acceptable CPU time for real-time purposes, although the iterative sequential approach has arisen as the most efficient one. The key is the proposed decomposition for the original centralised problem, which allows us to formulate two effortless independent problems (MIQP and NLP), solved efficiently by modern optimisation algorithms.

Apparently, the price-coordination approach does not have any advantage here: the optimal cost is slightly higher than the one reached by the centralised and it elapses double time than the sequential. This is because the centralised problem is not so large-scale, many constraints are affine in the decision variables, and the advantages of parallel computation cannot be exploited with just two local optimisation subproblems. Nevertheless, this approach will clearly beat the other two if the problem is extended to consider uncertainty in an explicitly way via, for instance, two-stage stochastic optimisation.

## References

- P. Bonami, L. T. Biegler, A. R. Conn, G. Cornuéjols, I. E. Grossmann, C. D. Laird, J. Lee, A. Lodi, F. Margot, N. Sawaya, A. Wächter", 2008. An algorithmic framework for convex mixed integer nonlinear programs. *Discrete Optimization* 5 (2), 186–204.
- R. Cheng, J. Forbes, W. Yip, 2007. Price-driven coordination method for solving plant-wide mpc problems. *Journal of Process Control* 17 (5), 429 – 438.
- L. L. C. Gurobi Optimization, 2018. Gurobi optimizer reference manual. <http://www.gurobi.com>.
- W. E. Hart, C. D. Laird, J.-P. Watson, D. L. Woodruff, G. A. Hackebeil, B. L. Nicholson, J. D. Sirola, 2017. *Pyomo—optimization modeling in python*, 2nd Edition. Vol. 67. Springer Science & Business Media.
- M. Kalliski, J. L. Pitarch, C. Jasch, C. de Prada, 2019. Apoyo a la toma de decisión en una red de evaporadores industriales. *Revista Iberoamericana de Automática e Informática industrial* 16 (1), 26–35.
- M. P. Marcos, J. L. Pitarch, C. de Prada, C. Jasch, Oct 2018. Modelling and real-time optimisation of an industrial cooling-water network. In: 2018 22nd International Conference on System Theory, Control and Computing (ICSTCC), pp. 591–596.
- C. G. Palacín, J. L. Pitarch, C. Jasch, C. A. Méndez, C. de Prada, 2018. Robust integrated production-maintenance scheduling for an evaporation network. *Computers & Chemical Eng.* 110, 140 – 151.
- J. L. Pitarch, C. G. Palacín, C. de Prada, B. Voglauer, G. Seyfriedsberger, 2017a. Optimisation of the resource efficiency in an industrial evaporation system. *Journal of Process Control* 56, 1–12.
- J. L. Pitarch, C. G. Palacín, A. Merino, C. de Prada, 2017b. Optimal operation of an evaporation process. In: H. G. Bock, H. X. Phu, R. Rannacher, J. P. Schlöder (Eds.), *Modeling, Simulation and Optimization of Complex Processes HPSC 2015*. Springer International Publishing, Cham, pp. 189–203.
- R. T. Rockafellar, R. J.-B. Wets, 1991. Scenarios and policy aggregation in optimization under uncertainty. *Mathematics of operations research* 16 (1), 119–147.
- A. Wächter, L. T. Biegler, Mar 2006. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming* 106 (1), 25–57.