D3.1 – Preliminary report on optimisation methods for large plants with discrete and continuous degrees of freedom

José Luis PITARCH\textsuperscript{a} (UVA)
César DE PRADA\textsuperscript{b} (UVA)
Simon WENZEL\textsuperscript{c} (TUDO)
Yannik MISZ\textsuperscript{c} (TUDO)

\textsuperscript{a}Research Associate, \textsuperscript{b}Professor, \textsuperscript{c}Research Assistant
\textsuperscript{a,b}Spain, \textsuperscript{c}Germany

April 2018

www.copro-project.eu
**PROJECT TITLE**
Improved energy and resource efficiency by better coordination of production in the process industries

**PROJECT ACRONYM**
CoPRO

**GRANT AGREEMENT NO**
723575

**INSTRUMENT**
RESEARCH AND INNOVATION ACTION

**CALL**
H2020-SPIRE-02-2016

**STARTING DATE OF THE PROJECT**
NOVEMBER 1ST, 2016

**PROJECT DURATION**
42 MONTHS

**PROJECT COORDINATOR (ORGANISATION)**
PROF. SEBASTIAN ENGELL (TUDO)

---

**THE COPRO PROJECT**

The goal of CoPro is to develop and to demonstrate methods and tools for process monitoring and optimal dynamic planning, scheduling and control of plants, industrial sites and clusters under dynamic market conditions. CoPro pays special attention to the role of operators and managers in plant-wide control solutions and to the deployment of advanced solutions in industrial sites with a heterogeneous IT environment. As the effort required for the development and maintenance of accurate plant models is the bottleneck for the development and long-term operation of advanced control and scheduling solutions, CoPro will develop methods for efficient modelling and for model quality monitoring and model adaptation.

**The CoPro Consortium**

<table>
<thead>
<tr>
<th>Participant No</th>
<th>Participant organisation name</th>
<th>Country</th>
<th>Organisation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 (Coordinator)</td>
<td>Technische Universität Dortmund (TUDO)</td>
<td>DE</td>
<td>HES</td>
</tr>
<tr>
<td>2</td>
<td>INEOS Köln GmbH (INEOS)</td>
<td>DE</td>
<td>IND</td>
</tr>
<tr>
<td>3</td>
<td>Covestro Deutschland AG (COV)</td>
<td>DE</td>
<td>IND</td>
</tr>
<tr>
<td>4</td>
<td>Procter &amp; Gamble Services Company NV (P&amp;G)</td>
<td>BE</td>
<td>IND</td>
</tr>
<tr>
<td>5</td>
<td>Lenzing Aktiengesellschaft (LENZING)</td>
<td>AU</td>
<td>IND</td>
</tr>
<tr>
<td>6</td>
<td>Frinsa del Noroeste S.A. (Frinsa)</td>
<td>ES</td>
<td>IND</td>
</tr>
<tr>
<td>7</td>
<td>Universidad de Valladolid (UVA)</td>
<td>ES</td>
<td>HES</td>
</tr>
<tr>
<td>8</td>
<td>École Polytechnique Fédérale de Lausanne (EPFL)</td>
<td>CH</td>
<td>HES</td>
</tr>
<tr>
<td>9</td>
<td>Ethniko Ktiroi Ervanas Kai Technologikis Anaptyxis (CERTH) IIM-CSIC (CSIC)</td>
<td>GR</td>
<td>RES</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>ES</td>
<td>RES</td>
</tr>
<tr>
<td>11</td>
<td>Leikon GmbH (LEIKON)</td>
<td>DE</td>
<td>SME</td>
</tr>
<tr>
<td>12</td>
<td>Process Systems Enterprise LTD (PSE)</td>
<td>UK</td>
<td>SME</td>
</tr>
<tr>
<td>13</td>
<td>Divis Intelligent Solutions GmbH (divis)</td>
<td>DE</td>
<td>SME</td>
</tr>
<tr>
<td>14</td>
<td>Argent &amp; Waugh Ltd. (Sabisu)</td>
<td>UK</td>
<td>SME</td>
</tr>
<tr>
<td>15</td>
<td>ASM Soft S.L (ASM)</td>
<td>ES</td>
<td>SME</td>
</tr>
<tr>
<td>16</td>
<td>ORSOFT GmbH (ORS)</td>
<td>DE</td>
<td>SME</td>
</tr>
<tr>
<td>17</td>
<td>Inno TSD (inno)</td>
<td>FR</td>
<td>SME</td>
</tr>
</tbody>
</table>
Abstract

This document summarizes the more recent methodologies to solve large-scale optimisation problems derived from continuous production processes, in which both continuous and discrete decisions need to be taken along a time horizon. The methods focus on the development of suitable mathematical models for scheduling which represent the actual problem accurate enough, but with affordable computational costs for execution in near-real time. Two case studies of industrial size, provided by the CoPro partners Lenzing AG and INEOS Köln, are selected as a proof of concept for the applicability of the proposed optimisation methods, reporting hopeful preliminary results.

Revision History

The following table describes the main changes done in the document since it was created.

<table>
<thead>
<tr>
<th>Revision</th>
<th>Date</th>
<th>Description</th>
<th>Author (Organisation)</th>
</tr>
</thead>
<tbody>
<tr>
<td>v0.1</td>
<td>Mar. 19, 2018</td>
<td>Creation of the document.</td>
<td>José Luis PITARCH (UVA)</td>
</tr>
<tr>
<td>v0.2</td>
<td>Mar. 27, 2018</td>
<td>Addition of the INEOS case study</td>
<td>S. WENZEL &amp; Y. MISZ (TUDO)</td>
</tr>
<tr>
<td>v0.3</td>
<td>Mar. 29, 2018</td>
<td>Addition of the introduction and problem statement. Internal review.</td>
<td>César DE PRADA (UVA)</td>
</tr>
<tr>
<td>v0.4</td>
<td>Apr. 4, 2018</td>
<td>External review and quality control.</td>
<td>Ivan KANTOR (EPFL)</td>
</tr>
<tr>
<td>v1.0</td>
<td>Apr. 5, 2018</td>
<td>Document edition and delivery to the project coordinator.</td>
<td>J.L. PITARCH (UVA); S. WENZEL (TUDO)</td>
</tr>
</tbody>
</table>

Disclaimer

THIS DOCUMENT IS PROVIDED “AS IS” WITH NO WARRANTIES WHATSOEVER, INCLUDING ANY WARRANTY OF MERCHANTABILITY, NONINFRINGEMENT, FITNESS FOR ANY PARTICULAR PURPOSE, OR ANY WARRANTY OTHERWISE ARISING OUT OF ANY PROPOSAL, SPECIFICATION OR SAMPLE. Any liability, including liability for infringement of any proprietary rights, relating to use of information in this document is disclaimed. No license, express or implied, by estoppels or otherwise, to any intellectual property rights are granted herein. The members of the project CoPro do not accept any liability for actions or omissions of CoPro members or third parties and disclaims any obligation to enforce the use of this document. This document is subject to change without notice.
# Table of Contents

1 Executive Summary .......................... 1

2 Introduction .................................. 2
   2.1 Real-Time Optimisation (RTO) ........................................................................ 2
   2.2 Discontinuous operation .............................................................................. 3
   2.3 Motivation for considering discontinuous and robustness aspects in RTO .......... 4
   2.4 Structure of the report ................................................................................... 4

3 Operation with discontinuities due to maintenance ...................................... 5
   3.1 Modelling with discrete decisions and logic .................................................. 5
   3.2 Algorithms for mixed-integer optimisation .................................................... 7

4 Methods for large-scale scheduling problems including uncertainty ............... 10
   4.1 Process State Transition Network (PSTN) ..................................................... 10
      4.1.1 Operational representation ................................................................... 11
      4.1.2 Extension to processes with decaying performance .............................. 13
   4.2 Combined discrete-continuous time scheduling ............................................ 14
   4.3 Multistage stochastic programming .............................................................. 16

5 CoPro case studies ......................................................................................... 19
   5.1 The evaporation system at Lenzing AG ......................................................... 19
      5.1.1 Plant surrogate modeling .................................................................... 19
      5.1.2 System description .............................................................................. 20
      5.1.3 Proposed modeling for scheduling ....................................................... 21
      5.1.4 Logic formulation .............................................................................. 22
      5.1.5 Two-stage stochastic schedule ............................................................. 25
      5.1.6 Illustrative results ............................................................................. 26
   5.2 The ammonia distribution network at INEOS Köln ..................................... 30
      5.2.1 Network modelling ............................................................................ 31
      5.2.2 Investigated scenario and implementation ............................................ 35
      5.2.3 Illustrative results ............................................................................. 37

6 First conclusions and ongoing work ................................................................. 40

Bibliography ................................. 41
### List of Figures

1. Hierarchy of decision layers in a process factory. ................................................. 3
2. Original and relaxed feasible sets. ................................................................. 7
3. Branching with binary values in MIP. Each arrow represents a fixed choice of a binary variable and each node a LP or NLP problem. ................................................................. 7
4. Master problem of the OA approach. ............................................................. 8
5. NLP and master subproblems in OA. .............................................................. 9
6. Example of conceptual modelling applying the PSTN approach [12]. .................... 11
7. Plant operation profiles approximated by piecewise linear segments. ............... 15
8. Plant assignment to products and load distribution. .......................................... 17
9. Exogenous scenario tree and its representation in the decision space [20]. ........ 17
10. Scheme of a single multi-effect evaporation plant. ............................................ 19
11. Evolution of the percentage increase of steam consumption due to fouling for several operation cycles. ................................................................. 20
12. Surrogate model for the steam consumption in a single plant. ......................... 21
13. Plant assignment to products and load distribution. ........................................ 21
14. Simplified scheme of the automaton. ............................................................. 22
15. Two-stage approach considering uncertainty in the weather and two products. .... 26
16. Optimal deterministic schedule. ................................................................. 28
17. Two-stage stochastic schedule for 3 plants and 2 products. ............................. 28
18. Two-stage stochastic schedule. ................................................................. 29
19. Schematic topology of the INEOS Köln network with its processing plants. .... 30
20. Modelling of plants with multiple reactors. Incoming/outgoing streams are split/added in internal node (circles), affine models for plant and individual reactors (rectangles) are used to calculate in- and outputs. ................................................................. 33
21. Scaled energy charts. The prices are given relative to the maximum price in the optimisation horizon for confidentiality reasons. ................................. 35
22. NH₃ production/consumption of the plants scaled by the maximum possible over the horizon. ................................. 38
23. Scaled tank levels over the horizon. ............................................................. 38
List of Tables

1. Connections product-plant and nominal efficiencies. ........................................ 27
2. Connections product-plant and nominal efficiencies. ........................................ 29
3. Information for the logistic constraints of the optimisation problem (day, processing hours). 37
1. Executive Summary

According to the CoPro description of work, the partner UVA is the leader for Task 3.1, with contributions from partners TUDO, PSE, INEOS and LENZING. The main objectives set up for this task were:

1. Development of architectures for the coordination between plant-wide RTO and local control layers.

2. Development of methods for the large scale optimization of dynamical systems with scheduling of cleaning and maintenance.

3. Development of robust approaches to cope with process uncertainty and process-model mismatch.

Task 3.1 considers two industrial case studies linked to these type of problems: the ones of INEOS and Lenzing. In both cases, problems related to the degradation of the equipment over time due to fouling and coking are present. They can be considered as the main disturbances affecting production as they imply periodic process units stops for maintenance, which severely affect production. Consequently, plant optimization requires production scheduling including that periods of time used for maintenance.

During the first 18 months of project, the involved partners have been working around two main tasks:

1. Review of the recent literature on methods to model production scheduling problems and algorithms for large-scale mixed-integer optimization including uncertainty, identifying the most suitable ones for the problems provided by the CoPro industrial partners, adapting them or developing novel functionalities.

2. Detailed definition of the case studies provided by the industrial partners which are chosen to serve as proof of concept for the proposed optimization schemes and production scheduling methods.

The criteria used for selecting modelling methodologies in the literature review were the ability of handling synchronization constraints (which naturally arise in continuous production processes) and the computational burden required to reach the desired level of accuracy by the industrial end users.

This report presents the result of this research, where the recently proposed Process State Transition Network (PSTN) approach for production scheduling has been considered the more suitable one, and has been extended by the CoPro partners to cope with processes with decaying performance due to equipment degradation. These approaches, the original and the extended, have been employed to model the problems of the case studies: the evaporation park in Lenzing AG and the ammonia distribution network in INEOS Köln. First tests in simulated instances show promising results in terms of accuracy and computational time, which hopefully will allow us to solve the optimisations in near-real time decision-support tools.

Additionally, a novel methodology to address production scheduling problems arisen in batch-continuous processes has been developed, where synchronization constraints in discrete time are naturally checked along plant operation continuous profiles (production or utilities demand). Briefly, the main benefit of this approach over the fully discrete-time PSTN or fully continuous-time scheduling approaches is the drastic reduction of computational time while keeping a good level of accuracy in the solutions.
2. Introduction

One of the aims of CoPro is to develop methods and algorithms for improving and optimising in real-time the operation of large process plants. The focus is not in the process units themselves, but on the way the whole plant or site is operated, considering those variables that represent the degrees of freedom of the process as well as the interaction among the different units and their state. The focus is also on plants that operate mainly with continuous production, in contrast with discrete manufacturing factories, but considering that discrete decisions have to be taken from time to time such as start and stop for cleaning and maintenance or production changes.

Process factories are normally composed of many sub-plants and process units that perform specific tasks and work together to manufacture the final products, including utilities, raw materials supply. Proper operation of these factories means coordinating the joint operation of all of them, avoiding production bottlenecks, managing shared resources, deciding the functioning and load level of the process units, deciding on the energy use, respecting safety, quality and environmental constraints and obtaining maximum efficiency, among other aims.

It is clear that even though very significant progress has been made in the optimal operational production planning for continuous processes, a highly efficient and systematic solution strategy of large-scale industrial problems is still an unresolved issue.

2.1 Real-Time Optimisation (RTO)

The complexity of the decisions made at this level is not small, as many aspects and constraints have to be taken into account. Because of this, everyone recognizes that process models are required to make decisions. Developing appropriate models that reflect the process and its operation is then a central activity in any process optimisation. Typically, once one has a model, optimisation tools are used to select the best choices respecting the process constraints and the links among variables imposed by the model. The operation decisions are to be taken in real time, collecting measurements and information from the process and deciding what to do at regular time intervals, giving rise to what is known as a Real-Time Optimisation (RTO) system. The type of models required depends on many factors such as process aims and process nature. Quite often, due to the size of the problems and the complexity of the resulting optimisation problems, which must be solved in real time, stationary models are used, looking for the ideal value of the process variables according to the process conditions and targets. This corresponds normally to a NLP (Non-Linear Programming) problem that can be solved with the corresponding algorithms.

\[
\min_{x \in \mathbb{R}} J(x) \quad \text{s.t.: } h(x) = 0, \; g(x) \leq 0
\]  

Here \(x\) denotes the vector of process variables (decisions), \(J(x)\) the objective function, \(h(x)\) are the (possible nonlinear) equations of the process model and \(g(x)\) denotes some constraints for optimisation. There are good numerical solvers available for these problems. Depending on the framework, one can mention IPOPT or KNITRO for large scale problems where the problem is solved directly in an optimisation environment, SNOPT for sequential approaches where the optimiser calls a model simulation reducing the decision variables to the degrees of freedom, or derivative-free methods that use some type of surrogate model to avoid computing gradients on the original one.

Then, the computed optimal values are passed to the underlying control layer for implementation as targets to follow and the procedure is repeated periodically according to the scheme of Figure 1. This scheme corresponds fairly well to what can be found in some industrial implementations of RTO. Of course, in practice it incorporates other elements, such as steady-state detectors, data-reconciliation modules to obtain reliable variable estimations and model updates, etc., but what is important is to notice that this scheme is oriented to obtain the optimal operation targets taking into account only the current plant state. It does not incorporate any element of the future when deciding on current operation at the RTO level. That is, how current operation affects future plant performance. At the same time, the quality of these decisions depends
Preliminary report on optimisation methods for large plants with discrete and continuous degrees of freedom

not only on the optimisation methods, but on the quality of the models and on external factors and disturbances not taken into account. In addition, notice that an NLP problem like Eq. (2.1) assumes that the process operation can be well represented by continuous models and real variables. Nevertheless, in practice, even if the production can be characterized mainly as continuous, there are many elements that do not follow that pattern.

2.2 Discontinuous operation

This is the case when, for instance:

• There are several units working in parallel and performing similar tasks, each one operating only in a range with the lower limit above zero, and the number of units in operation may depend on the plant load. In this case, discrete decisions about which ones should work must be taken, besides others affecting the operation of the plant.

• There are different modes of operation, including for instance alternative storages of products, which can be chosen depending on internal or external conditions (weather, supplies or demands...), so that some units will be out of service for some time and the mode of operation must be selected using integer variables which does not fit with problem (2.1).

• Phenomena like fouling, cooking or catalyst deactivation are present in the process units forcing them to stop for cleaning or refilling after some time of operation. So, some units will be out of service for some time, which will affect the normal functioning. Planning of the operation over time is required, which again does not fit with problem (2.1), leading to production scheduling problems in order to cope with these discontinuities.

Discontinuities like the ones above mentioned imply considering on one hand the need of incorporating binary variables and logic in the formulation of the optimisation problems in the RTO level and, on other hand, taking into account the dynamic aspects of the problem, leading to production scheduling problems. Both cases normally appear as mixed-integer nonlinear programming problems (MINLP) such as:

\[
\min_{x,y} J(x, y) \quad \text{s.t.: } h(x, y) = 0, \quad g(x, y) \leq 0, \quad x \in \mathbb{R}, \quad y \in \{0, 1\}
\]  

(2.2)

Where both vectors of variables, \(x\) and \(y\), span over certain time horizon. This mathematical problem can be closer to reality if we recognize that, likely, there are always wrong parameters or variables with a stochastic behaviour in the model, such as \(\chi\) in:

\[
\min_{x,y} J(x, y, \chi) \quad \text{s.t.: } h(x, y, \chi) = 0, \quad g(x, y, \chi) \leq 0, \quad x \in \mathbb{R}, \quad y \in \{0, 1\}
\]  

(2.3)
Or structural differences with the plant, which leads to consider the robustness of the solution as part of the problem.

### 2.3 Motivation for considering discontinuous and robustness aspects in RTO

CoPro tries to consider these elements, going beyond current practice by incorporating them into the optimal plant-wide decision making. The research has several aims:

- Expand the range of problems that can be effectively treated in the framework of RTO. As mentioned above, there are many structural decisions that require binary variables for its proper characterisation. Not including them in the RTO formulation would lead to incorrect or non-implementable solutions.

- Plan for the future, considering the long term effect of current decisions on the state or efficiency of the equipment after some time. In this way, optimum operation policies over selected time horizons can be obtained, providing improved overall efficiency.

- Avoid possible bottlenecks and improve efficiency by scheduling the use of the resources and process units over time, taking into account the discontinuities in their use imposed by maintenance works, modes of operation or other discontinuities.

- Generate the best balanced solutions by explicitly considering the uncertainty involved in the use of models or the presence of unknown disturbances in the decision making process. That way of operating involves a set of possible scenarios, instead of a single (and possibly wrong) one, providing robustness to the decisions.

### 2.4 Structure of the report

This “First report on optimisation methods for large plants with discrete and continuous degrees of freedom” is organized as follows: After the introduction, the main aspects related to the operation with discontinuities are reviewed in Section 3, considering modelling and the basic optimisation algorithms for mixed-integer optimisation. Then, Section 4 deals with methods for large-scale scheduling problems, including uncertainty, where some scheduling approaches oriented to degradation issues and periodic maintenance of process units while keeping production levels are described. This section includes also multistage stochastic programming approaches as a way to include uncertainty in the scheduling. Next, Section 5 is devoted to two case studies used as a proof of concept for the methods presented in the previous sections: The Lenzing evaporation system and the ammonia distribution network at INEOS Köln. The report ends with a first conclusions and ongoing work section and related bibliography. It will be followed by deliverables D3.2 and D3.3 in M36, that will describe more in detail the use of the methods in these two case studies.
3. Operation with discontinuities due to maintenance

The type of discontinuities considered in this report are the ones found more frequently in the continuous process industry, which refer mainly to two problems:

1. Selection of operation modes involving which units should operate or how they should operate.
2. Scheduling of the process units over time to cope with maintenance operations due to efficiency degradation.

This section focuses on formulations and methods that fits the first one, while scheduling problems are considered in Section 4. As with other optimisation problems, two key steps when incorporating discontinuities in the RTO layer are modelling and optimisation algorithms.

3.1 Modelling with discrete decisions and logic

Discrete decisions can be represented using binary variables \( y \) that can take only two values, 0 and 1. Note that when a discrete variable \( z \) can have more than two values (i.e. integer values), it can be substituted by several binary variables \( y_k \) linked by the equation

\[
z = y_0 + 2y_1 + 4y_2 + 8y_3 + 16y_4 + \cdots + 2^k y_k
\]

which can represent up to \( 2^{k+1} \) different integer values. These variables can be used to represent different operating modes, start or stop of process units, etc. In practice, the problems normally also include other real variables \( x \), so that the resulting optimisation problem is a mixed-integer one like Eq. (2.2). Nevertheless, the formulation of the problem often involves rules or conditions that are described by logic propositions, sentences which can have values true or false. A logic proposition is a set of logic expressions \( P_i \) linked by the logic operators AND, OR, NEGATION, IMPLICATION, etc. The logic propositions can be formulated as equations associating each \( P_i \) (True,False) with \( y_i \) (1,0), and \( \neg P_i \) with \( 1 - y_i \).

There are some basic modelling rules that translate elementary logic propositions into equations involving binary variables. They are illustrated next using three logic propositions \( P_i \):

\[
\begin{align*}
P_1 \lor P_2 \lor P_3 & \quad | \quad y_1 + y_2 + y_3 \geq 1 \\
P_1 \land P_2 \land P_3 & \quad | \quad y_1 \geq 1, y_2 \geq 1, y_3 \geq 1 \\
\text{One and only one among} \{P_1, P_2, P_3\} & \quad | \quad y_1 + y_2 + y_3 = 1 \\
P_1 \Rightarrow P_2 & \quad | \quad y_1 \leq y_2 \\
P_1 \text{if and only if } P_2 & \quad | \quad y_1 = y_2
\end{align*}
\]

The first one refers to selecting at least one alternative among the ones represented by \( P_1, P_2 \) and \( P_3 \) whereas Eq. (3.2b) represents that all of them must be selected. The statement (3.2c) means that one and only one must be selected. The implication (3.2d) means that if \( P_1 \) is true (that is, if the alternative 1 is selected) then \( P_2 \) must also be selected. Finally, Eq. (3.2e) represents the way of describing that \( P_1 \) is true iff \( P_2 \) is true. Using these equivalences, it is possible to convert any logic proposition \( P \) to an associated set of equations in the binary variables \( y_i \) if the logic expression is written in its normal conjunctive form:

\[
Q_1 \land Q_2 \land \cdots \land Q_n
\]

Where \( Q_i \) are logic expressions written as disjunctions (\( \lor \) operator). The systematic procedure for converting any proposition into form (3.3) follows three steps:

1. Replace the implication (if any) by its equivalent expression:

\[
P_1 \Rightarrow P_2 \iff \overline{P_1} \lor P_2
\]
2. Apply the Morgan’s laws to move inside the negations:

\[ P_1 \land P_2 \Leftrightarrow P_1 \lor (P_2 \lor P_1) \Leftrightarrow P_1 \land P_2 \]  

(3.5)

3. Use the distributive property to arrive to normal conjunctive form:

\[ (P_1 \land P_2) \lor P_3 \Leftrightarrow (P_1 \lor P_3) \land (P_2 \lor P_3) \]  

(3.6)

In this way, the logic associated to the selection of different modes of operation or alternatives can be incorporated as linear equations involving binary variables in the problem formulation. Normally, changing from one mode of operation to another or selecting one alternative, which is represented by a binary variable \( y \), implies removing or adding to the problem model a set of equations of forcing some values to zero. Of course, forcing a variable \( x \) to zero as a function of \( y \) can be done substituting it by the product \( xy \), but this should be avoided as it implies introducing non-convex and nonlinear terms into the problem. Instead, the following equations can be used, which only add linear expressions:

\[ -M \cdot (1 - y) \leq x \leq M \cdot (1 - y), \quad M \gg 0 \]  

(3.7)

Similarly, one can force the continuous variable \( q \) to have a value 0 or a positive one between lower and upper bounds \( L \) and \( U \), as a function of a logic condition represented by a binary variable \( y \) by means of:

\[ L \cdot y \leq q \leq U \cdot y \]  

(3.8)

One can activate or deactivate a set of equations \( h(x) = 0 \) and constraints \( g(x) \leq 0 \) associated to a binary variable \( y \), adding new positive slack variables \( s, v \), by means of:

\[ h(x) + s - v = 0 \]  

(3.9a)

\[ s + v \leq M_1 \cdot (1 - y), \quad M_1 \gg 0 \]  

(3.9b)

\[ g(x) - M_2 \cdot (1 - y) \leq 0, \quad M_2 \gg 0 \]  

(3.9c)

\[ s \geq 0, \quad v \geq 0 \]  

(3.9d)

A rather systematic way of modelling alternatives of operation is the so called disjunctive programming [1], in which the different alternatives are associated to different binary variables as in

\[
\begin{aligned}
\min_{x \in \mathbb{R}, \, y \in \{0, 1\}} & \quad c + J(x, y) \\
\text{s.t.:} & \quad h(x, y) = 0, \ g(x, y) \leq 0, \ \Omega(y) = 0 \\
& \quad \left[ \begin{array}{c}
\frac{y_1}{c = c_1} \\
\frac{y_2}{c = c_2} \\
\vdots
\end{array} \right] \lor \left[ \begin{array}{c}
\frac{y_2}{c = c_2} \\
\vdots
\end{array} \right] \lor \cdots \left[ \begin{array}{c}
\frac{y_n}{c = c_n} \\
\vdots
\end{array} \right] \\
& \quad g_1(x) \leq M_1 \cdot (1 - y_1) \\
& \quad g_2(x) \leq M_2 \cdot (1 - y_2) \\
& \quad \cdots \\
& \quad g_n(x) \leq M_n \cdot (1 - y_n) \\
\end{aligned}
\]  

(3.10a)

which combines the global formulation (3.10a) with alternative equations specific for every \( y_i \). One advantage of this formulation, besides simplicity, is that it can be translated more or less automatically to equation form, using, for instance, the big-M formulation:

\[
\begin{aligned}
\min_{x \in \mathbb{R}, \, y \in \{0, 1\}} & \quad \sum_{i=1}^{n} c_i y_i + J(x, y) \\
\text{s.t.:} & \quad h(x, y) = 0, \ g(x, y) \leq 0, \ \Omega(y) = 0 \\
& \quad g_i(x) \leq M \cdot (1 - y_i) \quad i = 1, \ldots, n \\
& \quad \cdots \\
& \quad \sum_{i=1}^{n} y_i = 1, \quad \Omega(y) = 0, \quad M \gg 0
\end{aligned}
\]  

(3.11a)
3.2 Algorithms for mixed-integer optimisation

Solving mixed-integer optimisation problems like Eq. (2.2) is considerably harder than their counterpart continuous variable versions. Because of this, whenever possible, one should try to formulate them in linear terms. Today, many Mixed-Integer Linear Programming (MILP) problems can be solved efficiently with state-of-the-art solvers such as CPLEX [2].

The best known approach for solving MILP or MINLP problems is Branch and Bound (B&B). This method is based on an intelligent search of the optimum combining the choice of integer combinations with relaxations and the generation of lower and upper bounds of the objective function that leads to the solution. It uses three main ideas:

- **Relaxation**, that converts integer into real variables and allows to compute bounds on the objective function.
- **Branching**, that generates alternatives of combinations of integer variables in the decision tree.
- **Fathoming**, examining the bounds allows to eliminate groups of integer combinations, thus improving the search.

A relaxation of an integer variable in a MILP or MINLP problem consists of allowing it to take any real value between its maximum and minimum range. For instance, a binary variable could take values within the interval $0 \leq y \leq 1$. So, in the relaxed problem, all variables, $x$ and $y$, are continuous and the corresponding problem is LP or NLP. Consequently, as the search space is widening, as illustrated in Figure 2, the solution of the relaxed problem is a lower bound (upper bound if the problem is a maximisation one) of the original MILP or MINLP. The relaxation is made with the purpose of obtaining such a bound.

![Figure 2: Original and relaxed feasible sets.](image)

Then, in the branching step, some binary variables are fixed to a 0 or 1 value following certain rules, and the new constrained LP or NLP problem is solved. If integer values for the $y$ variables are found, that solution is a feasible candidate for the MILP or MINLP problem, providing an upper bound of a minimisation problem. If not, the solution will generate a new lower bound of the problem, larger than the previous one.

![Figure 3: Branching with binary values in MIP. Each arrow represent a fixed choice of a binary variable and each node a LP or NLP problem.](image)

The procedure continues branching as in Figure 3, using the values of the bounds to fathom branches without the need of computing its values until the gap between the upper and lower bounds is less than a certain desired accuracy or no more branching is possible according to the values of the bounds. The best candidate
is then the problem solution and such gap measures its quality. This basic B&B algorithm has been improved in several ways. The best known approach is called Branch and Cut in which at each node, new valid linear inequalities, also called “cutting planes” or “cuts”, are added in order to improve the relaxation bound and reducing the searching space and consequently explore fewer branch-and-bound nodes.

A valid inequality should always satisfy the problem constraints. For a MINLP problem, if \( K \) points satisfying an integer solution are available, then, a lower bound of the search can also be obtained solving the MILP approximated problem (cutting plane):

\[
\min_{x \in \mathbb{R}, y \in \{0,1\}} \alpha \\
\text{s.t.: } \alpha \geq J(x_k, y_k) + \nabla J(x_k, y_k)^T \begin{bmatrix} x - x_k \\ y - y_k \end{bmatrix} \\
0 = h(x_k, y_k) + \nabla h(x_k, y_k)^T \begin{bmatrix} x - x_k \\ y - y_k \end{bmatrix} \\
0 \geq g(x_k, y_k) + \nabla g(x_k, y_k)^T \begin{bmatrix} x - x_k \\ y - y_k \end{bmatrix}
\] (3.12)

With \((x_k, y_k)\) a fixed point and the linearisation accumulating with increasing \(k\).

An alternative approach to B&B for MINLP problems is Outer Approximation (OA), where instead of exploring the integer space by branching, a dialogue among two subproblems (NLP and MILP) is performed like depicted in Figure 5.

![Diagram](image)

**Figure 4:** Master problem of the OA approach.

The NLP problem

\[
\min_{x \in \mathbb{R}, y \in \{0,1\}} J(x, y_k) \text{ s.t.: } g(x, y_k) \leq 0
\] (3.13)

is solved for a fixed vector \(y_k\) of the integer variables to obtain \(x_k\) and an upper bound of the problem.
the linearized master problem

\[
\begin{align*}
\min_{x \in \mathbb{R}, y \in \{0,1\}} & \quad \alpha \\
\text{s.t.:} & \quad \alpha \geq J(x_k, y_k) + \nabla J(x_k, y_k)' \left[ \begin{array}{c} x - x_k \\ y - y_k \end{array} \right] \\
& \quad 0 \geq g(x_k, y_k) + \nabla g(x_k, y_k)' \left[ \begin{array}{c} x - x_k \\ y - y_k \end{array} \right]
\end{align*}
\]  

(3.14a) (3.14b)

is executed to obtain \( y_k \) and a lower bound (Figure 4), iterating and adding new linear cuts to the MILP at each iteration until the gap is small enough.

Good descriptions of these methods can be found in several well-known references in literature, for instance [3]. There are also good reviews of these methods like, for instance [4].
4. Methods for large-scale scheduling problems including uncertainty

Conventionally, production scheduling problems have been represented (i.e. modelled) 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 deliverable the meaning of task is closely linked to the state in which a plant can be (processing a product $p$, standby or under a maintenance 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 (or part of it) is continuous, the production forecast may be uncertain and the best moment to schedule maintenance operations is not known.

Handling shared resources (steam, electricity, load, etc.) with a continuous-time approach requires synchronisation constraints to be fulfilled at all time $t$, which makes the problem computationally very demanding, hence usually not suitable for real-time implementations. Therefore, in order to handle such constraints in a direct way, predictive scheduling problems usually use a discrete-time approach in which the prediction horizon $H$ is split into $n$ time intervals of suitable length.

There exist several alternatives in order to formulate scheduling problems via mixed-integer and disjunctive programming [5], each of them influencing the final model structure, solver to be used and efficiency in obtaining a solution. Although implementation issues are not the main topic of this deliverable, understanding the associated advantages and drawbacks of each option is key. In this sense it must be noted that, even though current software tools can solve a complex nonlinear problem to (often local) optimality, it is of little use if the solution is not at hand when a decision needs to be made [6]. Although algorithms and computers are improving and getting faster every day, in general (unless an efficient decomposition technique can be applied [7]–[10]) still only linear approaches are able to give reasonable solutions in reasonable time for large-scale problems [11].

In the following we summarize two of the more recent scheduling formulations which have been developed to efficiently tackle large-scale problems in continuous processing plants.

4.1 Process State Transition Network (PSTN)

A conceptually different discrete-time MILP scheduling formulation is proposed in [12]. The scheduling problem determines the mode and the production amount of a plant/unit at every selected time such that a desired production demand is met at minimal energy cost. Of course, operational constraints regarding plant feasible transition modes, residence times, storage levels and production rates are also considered.

This approach is suitable for many process industries in which the production is mainly continuous but plants/units can get different operating modes, depending on external demands or market prices. Typically, the operation modes which are significant to consider are:

- **Start up**: A plant or unit which is “off” is required to reach a suitable state to start production. This mode usually involves ramp-up transitions respecting residence times, and may compress several phases.

- **Shutdown**: Contrary to the start-up procedure, after a plant or unit remains in the “on” mode longer than a given lapse of time, it can start the shutdown procedure. Ramp-down characteristics also apply in this mode.

- **Normal operation**: The plant is running and producing. Once started, normally the plant must remain in this mode for a minimum time, after which it can switch from “on” to “standby”, “off” or any other defined state.

- **Standby**: Applies when an active plant is not temporally needed and fulfils the minimum time in operation. Once in “standby”, this configuration must be held for a minimum time too. Note that this mode does not
involves any level of production, but can involve energy consumption (equipment is still empowered in order to be ready in a short time when required to produce).

- **Off operation**: The plant is entirely shut down and will remain “off” for a relative long time. Turning back the plant into operation will require time and energy consumption, according to the start-up procedure.

The main idea of this approach is the concept of *operational or transitional states* corresponding to admissible operating points of the system. This concept allows the *disaggregation of operation modes* and a more detailed modelling of the dynamic behavior arising from the transition to one mode to another. In this way, if we get a plant state with a given total duration, we could decompose it in shorter predefined sequential substates representing, for instance, initial states, intermediate states, and critical states. This disaggregation allows to formulate the scheduling problem with fewer binary variables and simpler constraints compared to previous scheduling approaches.

Figure 6b shows an example of this novel concept, where the state decomposition reflects the dynamic behavior of Plant A. Each node represents a specific transitional state of the system operation cycle according to the state graph (Figure 6a).

**Figure 6**: Example of conceptual modelling applying the PSTN approach [12].

Note that both nodes and arcs involve operational constraints which must be satisfied at all times. Note also that the PSTN approach naturally deals with systems having operating modes with different durations.

### 4.1.1 Operational representation

The scheduling formulation requires a set of constraints to represent the new state graph. The PSTN model includes constraints regarding operational decisions like production and storage levels, demand constraints, operating modes and feasible transitions, timing constraints, and energy balances. Additionally, the energy consumption is computed to minimise the total associated cost.

A set of binary transitional variables, say $W_{s,t}$, are introduced to indicate in which state $s$ the system is operating at time period $t$. 
Sequential Transition States. Transitions between states (discrete decisions) can happen provided that they are executed in the correct order, i.e., there are predefined sequences of operation that describe the behavior of the plant. Constraints (4.1)-(4.2) represent the operating sequence at on, off, or standby states. If the operating point of the plant at time $t$ is in the initial state of on, off, or standby mode, then at time $t + 1$ and $t + 2$ the plant has to operate in the corresponding states, intermediate $S_{\text{initial}}$ and critical $S_{\text{critical}}$, respectively.

\[
W_{s,t} = \sum_{s' \in S_{\text{inter}}} W_{s',t+1} \quad \forall t \in T, s \in S_{\text{initial}} \tag{4.1}
\]

\[
W_{s,t} \leq W_{s',t+1} \quad \forall t \in T, s \in L I C_s, s' \in S_{\text{critical}} \tag{4.2}
\]

Where $L I C_s$ stands for the subset of states that immediately precedes a critical state.

The start-up and shutdown procedures shouldn’t be interrupted. Consequently, during these processes the plant has to comply with given transition sequences through three different states, which have a residence time (i.e., $n_d$ time periods for shutdown and $n_s$ for start up). The feasible sequences are guaranteed by the following constraints:

\[
n_d W_{s,t} = \sum_{s' \in S_{\text{down-inter}}} \sum_{t' = t + 1}^{t + n_d} W_{s',t'} \quad \forall t \in T, s \in S_{\text{down-initial}} \tag{4.3}
\]

\[
n_s W_{s,t} = \sum_{s' \in S_{\text{up-inter}}} \sum_{t' = t + 1}^{t + n_s} W_{s',t'} \quad \forall t \in T, s \in S_{\text{up-initial}} \tag{4.4}
\]

However, additional constraints are necessary to complete the switching between the on and off states (see Figure 6b).

\[
W_{R D B, t} = W_{O F F, t+1} \quad \forall t \in T \tag{4.5}
\]

\[
W_{R U B, t} \leq W_{O N, t+1} \quad \forall t \in T \tag{4.6}
\]

Note that the ON$_i$ state has two possible previous states (SB$_n$ and RUB). This is the reason for the equality in constraint (4.5) whereas an inequality is stated for Eq. (4.6). Additionally, constraint (4.7) is also defined to represent these two feasible paths to ON$_i$ state.

\[
W_{S B n, t} + W_{R U C B} \geq W_{O N, t+1} \quad \forall t \in T \tag{4.7}
\]

Critical transition states. Once the system operates in a critical state (ON$_n$, OFF$_n$, and SB$_n$ in Fig.6b) at time $t$, it can remain in the same state or switch to other in the next period ($t + 1$). To describe possible transitions that can happen from the critical states, constraint (4.8) is formulated. The binary variable $W_{s,t}$ is used to meet the allowed switches. For example, when the plant operates in ON$_n$ state at time $t$, then in time period $t + 1$ it can operate in SB$_n$, RDA$_1$ or stay in the ON$_n$ state.

\[
W_{s,t} + W_{s',t} = \sum_{s'' \in N T S_s} W_{s'',t+1} \quad \forall t \in T, s \in S^{\text{critic}}, s' \in L I C_s \tag{4.8}
\]

Here $N T S_s$ stands for the subset of states that succeed a critical state.

Furthermore, the transition between an intermediate state and a critical state belonging to the same operating mode (on, off, or stand-by) is also modelled by Eq. (4.8). Note that the plant can operate only in one state at the same time, so only a binary variable on each side of the equality can be activated.

Residence time constraints. A unit which has both minimum and maximum stay constraints, enforces lower and upper bounds on the residence time of particular operation modes. These constraints become active when
there is a transition involved, depending on the previous state (usually after a start-up or shutdown). Hence, constraints similar to Eqs. (4.3)-(4.4) are used to restrict the minimum/maximum number of hours in which a plant is required to stay in a certain mode (see [12] for details).

**Production constraints.** The production level of each time period depends on its plant configuration and is denoted by the continuous variables $P_{s,t}$. Each operation state $s$ has a minimum ($\text{Min}P_s$) and a maximum ($\text{Max}P_s$) production limit. Hence, the binary variable $W_{s,t}$ is used in constraint (4.9) to guarantee that the production ($P_{s,t}$) will always satisfy the predefined allowed limits, taking into account the plant operation state in each time instant.

$$W_{s,t} \cdot \text{Min}P_s \leq P_{s,t} \leq W_{s,t} \cdot \text{Max}P_s \quad \forall t \in T, s \in S$$ (4.9)

Finally, as the plant has to operate in a single state/mode at each sample time, the binary variables are constrained as usual by $\sum_s W_{s,t} = 1 \quad \forall t \in T$. Of course the model is completed with the energy balances, inventory and/or storage level constraints.

A similar approach was proposed in [13], [14] (omitted for brevity). This approach has been also reported as an efficient scheduling representation for production planning problems. The authors of such references developed a discrete-time scheduling formulation to determine the production and inventory levels and the operation modes for each time period according to time-dependent electricity pricing schemes. Their MILP model was already used to evaluate an industrial case study on an air separation plant, and has been followed here in CoPro for addressing the INEOS ammonia distribution case study, see Section 5.2 of this report.

The major difference between the cited approach and the PSTN one is the way of representing the plant operation points and of capturing the transition mode behaviour: while [13], [14] represent the transitions defining the modes as a set of operating points which capture the whole transitional behaviour, the PSTN model disaggregates these modes in operational states at each time period.

### 4.1.2 Extension to processes with decaying performance

The previous formulation performs a disaggregation of the operation states such that intermediate stages can be accurately modelled and individual residence times are taken into account. In this formulation, when a plant is in operation (ON) after the first state $\text{ON}_i$, it is assumed to operate in the same state conditions (excluding the production load per time period). However, many systems suffer from performance degradation due to several factors in practice (fouling, coking, catalyst deactivation, mechanical tearing, etc.).

To model such effects in the production scheduling problem using PSTN, the underlying ideas of general precedence allocation [15] are used here to force the operation accordingly to a known time evolution of the decaying performance. Then, apart from the previous standby, maintenance, shutdown and start-up stages, the normal operation stage is split itself in as many discrete stages as needed, whose length normally depends on the time discretisation period. In this way, these additional stages are related to the time that a plant has been in operation (v.g., a plant that has just started operation will be in stage $s_0$, and when it has been working for two weeks will be in stage $s_{14}$).

Using these stages, we are able to indicate the plant time-varying performance due to the degradation via an associated cost value $K_s$ which impacts the objective function to optimise. Thus, the plant operational state will advance with time through the PSTN chart in which there will be a set of initial operation stages where no decision needs to be checked, followed by a limited number of alternatives on maintenance or standby. An illustrative schema of this modified PSTN for the Lenzing evaporation system is depicted in Figure 14.

Note that, according to this approach, once a plant has started operation it must continue in operation as many time instants as stages are defined in the initial set. This concept is analogous to the minimum running time in [16] required once a task has started.

The constraints required to enforce this modification are rather similar to the ones in the previous section and to the commonly used in general precedence approaches. For brevity, the reader is referred to Section 5.1 and [17] for a detailed formulation.

---

1 A previous analysis and maybe model identification of the degradation effect is assumed available.
4.2 Combined discrete-continuous time scheduling

This novel approach is designed for special plant-wide optimisation problems which result when continuous plants must operate together with batch ones in series interchanging materials and energy and sharing resources. In these cases, the usual static models and nonlinear programming tools of the RTO layer are useless due to the dynamic and discontinuous nature of the problems, which require coordination and scheduling of the operations over time in order to avoid creating bottlenecks, violating constraints on shared resources or risking the safe operation of the plant/unit.

To this aim, this section presents a novel and efficient formulation of the scheduling problem combining continuous and discrete-time domains. The approach looks for computational efficiency in order to be applied in real-time, and for enough flexibility to be adapted to the real process in a sensible way. This requires a MILP formulation, in spite of the presence of clear nonlinear profiles of many variables as well as large horizons that must cover at least a factory shift. With these aims in mind, the formulation of the problem combines the following three elements:

- An assignment problem, described in continuous time, to compute the scheduling of the individual plants.
- A description of the nonlinear profiles of the variables associated to the operation of the plants in terms of a set of local linear approximations.
- A description of the dynamic constraints imposed by the shared resources using a discrete time grid synchronized with the continuous one.

The reason for using two base times is linked to the fact that assignments in continuous time provide information only on the start and end times of each batch unit, but some shared continuous variables needs to check the fulfillment of its associated constraints at regular and more frequent time intervals.

Let’s think in a case where each plant will process a certain number of batches from a supply tank. We will denote by \( I \) the set of indexes \( i \), each one representing a batch or processed lot. In the same way, we will denote by \( M \) the set of batch units \( m \). The continuous inflow of material to the supply tank can be considered as composed of a series of material lots arriving to the tank to be processed later on. The size of each lot is equal to the amount of material consumed by a batch unit for a whole batch. Depending of the characteristics of the material, we can associate to each lot a processing time or duration in the plant that can be different in each one and that will be denoted as \( p_{i,m} \). In parallel, we can assign a cost associated to the processing of a lot \( i \) in a plant \( m \), denoted by \( c_{i,m} \). In addition, the scheduling problem uses the following continuous time variables:

- \( t_{s}(i) \) denotes the start time of lot number \( i \).
- \( t_{e}(i) \) denotes the earliest time in which lot \( i \) can be processed.
- \( t_{d}(i) \) denotes the latest time in which lot \( i \) should have been processed.

Note that \( t_{e} \) and \( t_{d} \) are known parameters at time zero, when the scheduling problem has to be solved over the prediction horizon. For all lots present in the supply tank according to its capacity, \( t_{e} \) will be zero, as they can be processed in parallel, while the rest will have different values of \( t_{e} \) according to their expected arrival times. As well, two binary variables are defined following a general precedence approach:

- \( y_{i,m} \) will be equal to 1 if lot \( i \) is processed in plant \( m \) and 0 otherwise
- \( z_{i,j} \) will be equal to 1 if lot \( i \) precedes lot \( j \) on the same plant and 0 otherwise

Then, the optimal allocation of lots and its sequencing within each batch unit can be formulated easily if we select as optimality criterion to minimise the overall costs by means of constraints (4.10b)-(4.10g). Here, only the cost of each effective assignment is counted in the objective function \( J \). Constraints (4.10b) and (4.10c) assure that a lot \( i \in I \) will not be assigned before the time \( t_{e}(i) \) and will be processed before the time \( t_{d}(i) \).

Equation (4.10d) guarantees that each lot will be processed in a plant and only in one of them. Next, inequality
(4.10e) serves to limit the number of lots that can be assigned to a single plant. Finally, Eq. (4.10f) assures that if two lots, \( i \) and \( j \), are assigned to the same plant \( m \), then either \( i \) precedes \( j \) or the other way around. Then, the last inequality (4.10g) establish that if \( i \) precedes \( j \) on plant \( m \), the start time \( t_s \) of \( j \) cannot be earlier than the start time of \( i \) plus its processing time.

\[
\min_{y, i, m} \quad J = \sum_{i \in I} \sum_{m \in M} c_{i,m} y_{i,m} \quad (4.10a)
\]

\[\text{s.t.:} \quad t_s(i) = t_e(i) \quad (4.10b)\]

\[t_s(i) \leq t_d(i) - \sum_{m \in M} p_{i,m} y_{i,m} \quad \forall i \in I \quad (4.10c)\]

\[\sum_{m \in M} y_{i,m} = 1 \quad \forall i \in I \quad (4.10d)\]

\[\sum_{i \in I} p_{i,m} y_{i,m} \leq \max_i \{t_d(i)\} - \min_i \{t_e(i)\} \quad \forall m \in M \quad (4.10e)\]

\[1 \geq z_{i,j} + z_{j,i} \geq y_{i,m} + y_{j,m} - 1 \quad \forall i, j \in I, i > j, m \in M \quad (4.10f)\]

\[t_s(j) \geq t_s(i) + \sum_{m \in M} p_{i,m} y_{i,m} - M \cdot (1 - z_{i,j}) \quad \forall i, j \in I, i \neq j \quad (4.10g)\]

The above formulation can be used for assignment but it does not consider neither the constraints associated to the shared resources (e.g., steam, cooling water, syrup, gas, etc.) nor the ones of the continuous elements (supply tank, receiver, etc.). Because of that, it needs to be expanded, first with a representation of the time evolution of flows associated to the shared resources and, then, with the corresponding level or range constraints. For the representation of the material demand profile of a plant over time, as well as for the demand of utilities, we define a set of time points \( t_p(n) \) covering from \(-t_N\) to \( t_N\) and approximate the continuous profile by a series of linear segments, each one with an associate binary variable \( w \), as in Figure 7. Outside the range of the operation, the profile is fixed to zero. This non-uniform discrete-time base is linked to each batch and needs to be synchronized with the overall time scale. Also notice that the times \( t_s(i) \) are only a relatively small number and likely not equally spaced over time, so that they are not adequate to check if a variable like the level in a tank is within the desired range. For these reasons, we need to define a third time base that consists of a larger set of time points \( t(k) \) equally distributed over the future horizon.

**Figure 7**: Plant operation profiles approximated by piecewise linear segments.

For the synchronisation of the different times, the following variables have been defined:
\( w_{k,i,n} \) is a binary variable equal to 1 if the time instant \( t(k) - t_s(i) \) belongs to the interval \( n \) of the corresponding profile and 0 otherwise.

\( \alpha_{k,i,n} \) is a coefficient between 0 and 1 used for linear interpolation in the \( n \) piecewise profiles for each time instant \( t(k) \) and each batch \( i \).

Hence, for example, the following set of equations can be used then for computing the total material flow \( Q(k) \), steam \( Q_s(k) \) and discharge to a receiver \( Q_d(k) \) at a given time instant \( t(k) \) by synchronizing the corresponding profiles and start times, and interpolating later on according to the particular value of the elapsed time.

\[
\sum_n w_{k,i,n} = 1, \quad w_{k,i,N} = 0 \quad (4.11a)
\]

\[
\sum_n \alpha_{k,i,n} = 1, \quad \alpha \geq 0 \quad (4.11b)
\]

\[
\alpha_{k,i,n} \leq w_{k,i,n-1} + w_{k,i,n} \quad (4.11c)
\]

\[
t(k) - t_s(i) = \sum_n \alpha_{k,i,n} t_p(n) \quad \forall i \in I, k \in K \quad (4.11d)
\]

\[
Q(k) = \sum_i \sum_n \alpha_{k,i,n} F(i,n) \quad (4.11e)
\]

\[
Q_d(k) = \sum_i \sum_n \alpha_{k,i,n} F_d(i,n) \quad (4.11f)
\]

\[
Q_s(k) = \sum_i \sum_n \alpha_{k,i,n} F_s(i,n) \quad (4.11g)
\]

Constraint (4.11a) establishes that one and only one interval \( n \) of the profile can be active in a time instant \( k \) and a batch \( i \). Equations (4.11b)-(4.11d) identify the interval \( n \) and the fraction of time \( \alpha \) of the interval that corresponds to the time relative to the beginning of the batch \( i \). The last set of equations (4.11e)-(4.11g) computes the corresponding flows interpolating with that fraction \( \alpha \) in the profiles. Note that these can be particularized according to the characteristics of the lot \( i \) being processed. Finally, once the global flows at the synchronization points \( k \) have been computed, mass balances in the tanks can be formulated and the constraints associated to levels or total steam demand are added to the model:

\[
L(k) = L(k-1) + \left( q(k) - Q(k) \right) \frac{\Delta t}{A_t} \quad (4.12a)
\]

\[
L_m(k) = L_m(k-1) + \left( Q_d(k) - q_c(k) \right) \frac{\Delta t}{A_m} \quad (4.12b)
\]

\[
L_{\text{max}} \geq L(k) \geq L_{\text{min}} \quad \forall k \in K \quad (4.12c)
\]

\[
Q_{\text{max}} \geq Q_s(k) \geq Q_{\text{min}} \quad (4.12d)
\]

Here \( L(k) \) stands for the level of material in the supply tank, \( L_m \) for the corresponding level in a receiver, \( q(k) \) represents the incoming material flow to the tank and \( q_c(k) \) the demand to the receiver.

This approach has been successfully tested in simulation with a reference process benchmark: a simplified crystallisation section of a sugar factory. For details on the example the reader is referred to [18].

### 4.3 Multistage stochastic programming

Uncertainty is a major drawback in actual implementations of RTO systems in the process industry. Several approaches have been proposed to explicitly incorporate uncertainty in mathematical optimization problems [10] but stochastic programming is probably one of the approaches with best tradeoff between robustness, conservatism and performance. It is based on optimizing the expected value of the objective function over
a set of possible uncertainty realisations [19]. Normally the realisations are characterized with a discrete set of possible values, suitably sampled from a probability distribution of the uncertainty. This leads to a finite number of scenarios \( S \), simplifying the calculation of the expected value. Accordingly, stochastic programming is often regarded as a scenario-based approach for optimisation under uncertainty. The formulation allows the consideration of alternative decisions at different stages (future time instants), according to the sequence in which uncertainty will be revealed (measurable). Therefore, the stages imply a discrete time representation of the problem and establish the information of which uncertain parameters are available at each time. The potential future paths in which discrete uncertain parameters might evolve are represented in a scenario tree as shown in Figure 8. In these trees, each node is a decision-making instance with known realisation of the uncertain parameters (up to the current time). The possible future realisations from each node are represented with branches.

The simplest stochastic programming formulation, called single stage, without recourse or robust, forces all decisions to be made before uncertainty reveals, i.e., a single solution which fulfils all the considered scenarios. This is usually very conservative, so the usual way to proceed is splitting the prediction horizon in a multi-stage fashion, where common decisions are made for all scenarios in the stage and the so-called recourse decisions adapt for the subsequent stages depending on which scenario is realised [20] (see Figure 9).

Probably, the most widely-used formulation is the MILP two-stage stochastic programming. As its name indicates, such formulation splits the decisions into two sets: robust decisions that are made before uncertainty reveals (robust horizon), and “wait-and-see” decisions that are customised for each scenario. The typical de-
Terministic formulation of a linear two-stage stochastic programming problem is:

\[
\min_{x_1, x_2[s]} J = c^T x_1 + \sum_{s \in S} p_s (d_{s[s]}^T x_2)
\]
\[
\text{s.t.: } W x_2[s] \leq h_s - T_s x_1 \quad \forall s \in S
\]

(4.13a)

Where the vector of robust \(x_1\) and recourse \(x_2\) decision variables belong to a mixed-integer polyhedral set \(X\) and \(s\) is the scenario index. An important property of the above linear formulation is that the feasible region for the robust variables \(x_1\) is a polyhedron, so all possible uncertainty realisation lying in the interior of the convex hull formed by the vertex scenarios is fully covered, even though not being explicitly considered in the tree.

The benefits of using a stochastic model can be quantified by the Value of the Stochastic Solution (VSS): the difference between the expected objective value obtained by the stochastic formulation and the one obtained by evaluating the solution of the deterministic formulation (just considering the nominal values of the uncertain parameters) in each scenario.

The formulation for the general multistage approach is omitted here for brevity, the reader is referred to the cited literature and references therein for more details. It’s worth to remark that the computational cost of multistage stochastic programming problems is considerably harder that the one for the two-stage approach. Furthermore, special care is needed to avoid anticipation of the uncertain parameters that have not been revealed yet. For these reasons, a two-stage approach is the one chosen for the Lenzing case study in Section 5.1.5 of this report.
5. CoPro case studies

The proposed optimisation methods have been applied to optimise the resource consumption and reduce costs by better tasks management and coordination in two industrial case studies: the evaporation system at Lenzing AG and the ammonia distribution network at INEOS Köln. Preliminary results obtained in simulation are presented next.

5.1 The evaporation system at Lenzing AG

The evaporation system works in continuous, processing some spinbath flows (products) in several evaporation plants. Hence, a first problem of optimal load allocation and product-plant assignment arises. The plants have different features (capacity, efficiency, etc.) and their actual performance is affected by both fouling inside the heat exchangers, load and external weather conditions.

Fouling is specially problematic, as the heat-transmission coefficient reduces with time so that the specific steam consumption (SSC) increases, hence costs too. Therefore, stopping the plant from time to time for cleaning becomes mandatory to recover nominal efficiency. Nevertheless, cleaning incurs an economic cost (manpower, cleaning products and the load switch to another plant). Hereby, the decision-support tool has to decide when maintenance operations have to be triggered in an optimal way. Therefore, a mixed production/maintenance scheduling problem arises. Furthermore, uncertainty in the weather forecast and in the production plan will be considered via the two-stage stochastic programming approach.

5.1.1 Plant surrogate modeling

Each evaporation plant is formed by a series of heat exchangers, evaporation chambers, condensers and cooling systems (see Figure 10).
Plant operating in isolation was already proposed in [22]. However, cross interactions appear when considering several plants and products in an interconnected system. Formulating such problem via MINLP becomes computationally challenging if the grey-box 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-optimising controller, we can build local surrogate plant models computed in different operating conditions [23], either from simulation with the nonlinear model or directly from process measurements. So, given a fouling state, these simulations/measurements allow to record a static map of the absolute steam consumption (ASC) as a function of the outdoor temperature $T_{\text{out}}$ 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 steam consumption around 16% can be identified between consecutive cleaning operations (see Figure 11). Therefore, a linear evolution of the dirt deposition over time can also be assumed, mainly depending on the time the evaporator has been in operation.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure11.png}
\caption{Evolution of the percentage increase of steam consumption due to fouling for several operation cycles.}
\end{figure}

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

$$\text{ASC}(v, t, p) = (K_T(v) \cdot T_{\text{out}}(t) + K_E(v)) \cdot P(v, t, p) + K_F(v, t)$$  \hspace{1cm} (5.1)

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 steam consumption due to the current state of fouling. This plant model (5.1) is supported by extensive experimental work (as can be seen in Fig.11) and it is the basis for the proposed scheduling approach in the following sections. As illustrative example, Figure 12 shows three surfaces corresponding to three different fouling states.

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

5.1.2 System description

The evaporation system consists of several 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

\[^1\text{In order to isolate the increase of steam consumption due to fouling, the plant is momentarily driven to reference conditions before taking a measurement.}\]
Preliminary report on optimisation methods for large plants with discrete and continuous degrees of freedom

Figure 12: Surrogate model for the steam consumption in a single plant.

a single product 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 13), where the operation cost differs from one plant to another due to the particular equipment efficiencies.

Figure 13: 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, because the production process is continuous.

5.1.3 Proposed modeling for scheduling

The prediction horizon $H$ has been discretized using one-day length as the shortest task unit. This choice is motivated by the fact that one day is the typical duration needed to complete a cleaning task and the assumption that the fouling state will not change significantly in one day. Then, the PSTN approach with decaying performance (Section 4.1.2) is 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.

Hence, 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 14, where the working stages are displayed as blue chevrons. These stages are related to the time that an evaporator has been in operation (e.g., 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 $K_F(v, s)$.
in Eq. (5.1). 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. 14). In addition to this, in Pitarch, Palacín, Prada, et al. [22] 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 very large (the fixed cost $K_C$ associated to a cleaning task is not amortized yet). In this way, there will be a set of initial stages where no decision about cleaning needs to be checked.

![Figure 14: Simplified scheme of the automaton.](image)

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 cleaning operations are represented in Figure 14 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, Merchan, and Maravelias [16] required once a task has started.

### 5.1.4 Logic formulation

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.
- $S$ will be the set of possible stages of an evaporation plant. As subsets it includes:
  - $S_I$ 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 shallow cleaning, e.g. of type $A$ (Fig. 14).
  - $S_A$ as stages where a decision between “keep working” or “make a cleaning task of type $A$” has to be made.
  - $S_L$ as cleaning stages, where $s_{LA}$ denotes a cleaning of type $A$. 

– \( S_P \) as the stages where the evaporation plants are stopped. Here we distinguish between two subsets: \( S_P C \) which includes stages where the plants are in standby waiting for a cleaning, e.g. \( s_{P_A} \) denotes a standby before a cleaning of type \( A \), and \( S_P L \) including stages where plants are in standby after being already cleaned, e.g. \( s_{P_L A} \) 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 realisations.

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 \( t \).
- \( P_{vtpe} \): continuous non-negative variables that assigns, in scenario \( e \), the evaporation flow of product \( p \) in plant \( v \) at time \( t \).
- \( C_{vtse} \): continuous non-negative variables that assigns, in scenario \( e \), the costs for a plant \( v \) being in stage \( s \) at time \( t \).

According to linear generalized disjunctive programming [24], the feasible transitions within stages in the proposed automaton are defined by the following positive (i.e., True) logic statements\(^2\):

1. An evaporator must be in one stage and only in one stage for each sample time.
   \[
   \bigvee_{s \in S} E_{vtse} \quad \forall v \in V, \forall t \in \mathcal{M}, \forall e \in \mathcal{E}
   \] (5.2)

2. An evaporator must be processing a single product, except if being cleaned or in standby.
   \[
   \bigvee_{p \in \mathcal{P}} \left( A_{vtpe} \bigvee_{s \in S_L,S_P} (E_{vtse}) \right) \quad \forall v \in V, \forall t \in \mathcal{M}, \forall e \in \mathcal{E}
   \] (5.3)

3. Only a single cleaning stage is allowed at any time period.
   \[
   \bigvee_{s \in S_L,v \in V} E_{vtse} \quad \forall t \in \mathcal{M}, \forall e \in \mathcal{E}
   \] (5.4)

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.
   \[
   E_{vtse} \rightarrow E_{v(t+1)(s+1)e} \quad \forall v \in V, \forall t \in \mathcal{M}\{t_F\}, \forall s \in S_I, \forall e \in \mathcal{E}
   \] (5.5)

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.
   \[
   E_{vtse} \rightarrow E_{v(t+1)(s+1)e} \lor E_{v(t+1)s_{L_A}e} \lor E_{v(t+1)s_{P_A}e} \quad \forall v \in V, \forall t \in \mathcal{M}\{t_F\}, \forall s \in S_A, \forall e \in \mathcal{E}
   \] (5.6)

6. A stopped evaporator which has not been already cleaned, must be cleaned or continue in standby.
   \[
   E_{vts_{P_A}e} \rightarrow E_{v(t+1)s_{P_A}e} \lor E_{v(t+1)s_{L_A}e} \quad \forall v \in V, \forall t \in \mathcal{M}\{t_F\}, \forall e \in \mathcal{E}
   \] (5.7)

\(^2\)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 [1] and Sawaya and Grossmann [24].
7. A clean evaporator in standby can continue in such state or begin to operate.

\[ E_{vt} \cup S_{PLAE} \rightarrow E_{vt(t+1)} \cup S_{PLAE} \lor E_{vt(t+1)} \cup s_e \quad \forall v \in V, \forall t \in M \setminus \{t_F\}, \forall e \in E \]  

(5.8)

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

\[ E_{vt} \cup S_{PLAE} \rightarrow E_{vt(t+1)} \cup S_{PLAE} \lor E_{vt(t+1)} \cup s_e \quad \forall v \in V, \forall t \in M \setminus \{t_F\}, \forall e \in E \]  

(5.9)

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

\[ A_{vtpe} \rightarrow A_{vt(t+1)pe} \lor E_{vt(t+1)se} \quad \forall v \in V, \forall t \in M \setminus \{t_F\}, \forall s \in \{S_L \cup S_{PC}\}, \forall p \in P, \forall e \in E \]  

(5.10)

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

\[ - \left( \bigvee_{s \in S_{PC}} E_{vt} \right) \quad \forall v \in V, \forall e \in E \]  

(5.11)

and by computing a terminal cost \( T_C \) if the plants end up working in an advanced fouling state\(^3\):

\[ T_C = \sum_{s \in S_A} E_{vt} \cdot K_C(v, s_{LA})/2 \quad \forall v \in V, \forall e \in E \]  

(5.12)

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

\[ \left[ \begin{array}{c} E_{vt} \\ C_{vt} = K(v, s) \end{array} \right] \lor \left[ \begin{array}{c} -E_{vt} \\ C_{vt} = 0 \end{array} \right] \quad \forall s \in S, \forall v \in V, \forall t \in M, \forall e \in E \]  

(5.13)

Where \( K(v, s) \in \{K_F(v, s_0), K_F(v, s_1), \ldots, K_F(v, s_n), K_S(v, s_{PL}), K_C(v, s_{LA})\} \) stands for the entry to a lookup table containing the fixed cost values associated to each plant stage (fouling state, standby stages defined in \( S_F \) and the available cleaning operations in \( S_L \)). Here, recalling Eq. (5.1), \( K_F = K_F \times P_{steam} \), being \( P_{steam} \) the price for steam generation (internally computed by Lenzing AG).

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 \( U_v(T_{out}) \) gets a known dependency with the ambient temperature.

\[ \left[ \begin{array}{c} A_{vtpe} \\ L_v \leq P_{vtpe} \\ P_{vtpe} \leq U_v(T_{out}) \end{array} \right] \lor \left[ \begin{array}{c} -A_{vtpe} \\ P_{vtpe} = 0 \end{array} \right] \quad \forall v \in V, \forall t \in M, \forall p \in P, \forall e \in E \]  

(5.14)

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

\[ \sum_{v \in V} (P_{vtpe}) \geq SP_{pte} \quad \forall t \in M, \forall p \in P, \forall e \in E \]  

(5.15)

The model is also constrained to the available physical connections between product circuits and evaporation plants \( A \) and to the initial (current) state of the plants \( S_0 \). Then, once the scheduling model is complete, an

---

\(^3\)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.
optimisation problem can be stated, for instance considering the usual risk-neutral economic objective [9] of minimising the normalized cost of operation for the overall evaporation network during a time horizon $H$:

$$\min J := \sum_{e \in \mathcal{E}} \sum_{t \in T} \sum_{v \in V} \sum_{p \in P} \left( K_T(v) T_{out}(t) + E_T E_{out}(v) \right) \cdot P_{vtp} \cdot P_{steam}$$

$$\text{s.t.: Eqs.}(5.2) - (5.15); A_{vtp} \in \mathcal{A} \forall t \in T, \forall v \in \mathcal{V}; E_{vtp} \in \mathcal{E}_0 \forall e \in \mathcal{E};$$

$$C_{vtp}, P_{vtp} \in \mathbb{R}^+; E_{vtp}, A_{vtp} \in \{\text{True, False}\}$$

(5.16)

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 $J$, 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 $E$, the above optimisation setup (5.16) reduces to the deterministic scheduling approach.

### 5.1.5 Two-stage stochastic schedule

The performance of the evaporation plants depends on two external factors: their load and the outdoor temperature, see Eq. (5.1). Consequently, if only the nominal scenario is considered in problem (5.16), 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 [20], [25] is adopted here to deal with such uncertainty considering that, even if at the current day there is no accurate information about the future realisations of the weather and evaporation loads, they will be known/measured at some future days. Hence, it is possible to compute many recourse scheduling actions which fit each considered uncertainty realisation. In particular, a two-stage approach is chosen because it is computationally less demanding, thus providing 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 $E$, 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 15a, where two max/min realisations for the outdoor temperature (extreme values $T_{out}$ and $T_{out}$ 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.

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 realisations make an 8-scenario tree. So, following the two-stage approach, decisions taken within the days belonging to the robust horizon $H_R$ must be unique, i.e., they must fulfil all scenarios. For instance, in Figure 15b, 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, many 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 multiplied by $2^\rho$ variations of evaporation demands (assuming $\rho$ products) make a $2^{\rho+1}$ scenario tree in $E$.

**Remark.** The robust horizon $H_R$ is usually chosen as the future time window along which getting reasonable information of the actual uncertainty realisation 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 realisations in the scenario tree. Thus, the high computational demands of sampling-based probabilistic approaches (e.g. Monte-Carlo) are avoided.
Preliminary report on optimisation methods for large plants with discrete and continuous degrees of freedom

Example of a possible scenario tree.

Robust schedule with \( H_R \) set to 8.

**Figure 15:** Two-stage approach considering uncertainty in the weather and two products.

Now, denote by \( M_U \) the subset of \( M \) 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 \( H_R \):

\[
E_{vtse} \equiv E_{vts}, \quad A_{vtpe} \equiv A_{vtp}, \quad P_{vtpe} \equiv P_{vtp} \quad \forall t \in M \setminus M_U, \quad \forall v \in V, \quad \forall e \in E, \quad \forall p \in P
\]

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

\[
\text{minimise } J \in \mathbb{R} \quad \text{subject to: Eqs.(5.2) – (5.15); (5.17);}
\]

\[
A_{vtpe} \in A \quad \forall t \in M, \forall e \in E; \quad E_{vtse} \in S_0 \quad \forall e \in E
\]

\[
C_{vtse}, P_{vtpe} \in \mathbb{R}^+; \quad E_{vtse}, A_{vtpe} \in \{\text{True, False}\}
\]

**Remark.** 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 \( M_U \), which facilitates the possibility of measuring the current uncertainty realisation in real time and adapting the schedule accordingly.

**5.1.6 Illustrative results**

In order to check the effectiveness of the proposed approach, several schedules have been computed in simulation for different instances of the problem. 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.
Deterministic solution

In this case, a system 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 are listed in Table 1.

Table 1: Connections product-plant and nominal efficiencies.

<table>
<thead>
<tr>
<th>v1</th>
<th>v2</th>
<th>v3</th>
<th>v4</th>
<th>v5</th>
<th>KE</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>0.6</td>
</tr>
<tr>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>0.7</td>
</tr>
<tr>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>0.8</td>
</tr>
<tr>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>0.9</td>
</tr>
<tr>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>1.0</td>
</tr>
<tr>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>1.1</td>
</tr>
<tr>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>1.2</td>
</tr>
<tr>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>1.3</td>
</tr>
<tr>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>1.4</td>
</tr>
<tr>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>1.5</td>
</tr>
<tr>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>1.6</td>
</tr>
<tr>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>1.7</td>
</tr>
<tr>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>1.8</td>
</tr>
<tr>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>1.9</td>
</tr>
<tr>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>2.0</td>
</tr>
<tr>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>2.1</td>
</tr>
<tr>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>2.2</td>
</tr>
<tr>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>2.3</td>
</tr>
</tbody>
</table>

Each plant cannot operate under a load of $L_v = 15 \text{T/h}$, but their maximum capacity varies with the weather condition, i.e. $U_v = 30 + f(T_{out})$, where $f(T_{out})$ is such that $U_v \leq 35 \text{T/h}$. 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 should not operate for more than 40 days without cleaning, because it is clearly suboptimal from the economic point of view [22]. So, the set $S$ is formed by $\{s_0, s_1, \ldots, s_{40}, s_{LA}, s_{LB}, s_{PA}, s_{PB}, s_{PLA}, s_{PLB}\}$.

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 $SP_5 = 68 \text{T/h}$. Hence, given a randomly fixed initial state of the network together with the above constraints, we run the economic optimisation (5.16) to provide the optimal load allocation as well as the task schedule within a prediction horizon of $H = 30$ days.

An optimal solution with relative gap less than 1% has been found for this problem (35150 binary variables, 3452 continuous ones and 42613 constraints) in about 11 minutes using up to 4 threads for concurrent optimisation in GAMS with GUROBI 7.0.2 over an Intel® i7-4510U CPU machine with 16 Gb of RAM memory. This solution is depicted in the Gantt diagram of Figure 16, 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.

The computed schedule shows how the optimiser tries to avoid using the less efficient plants when possible,

4Scaled values. Real ones are not included due to confidentiality agreements with Lenzing AG.
5See Section 3.2 for an explanation of this concept.
6Reducing 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.
either because they get higher $K_E$ or they are more fouled than others. The cleaning tasks are scheduled in a 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.

**Two-stage stochastic solution**

Now we introduce uncertainty as explained in Section 5.1.5. 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.

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 realisations, a 4-scenario tree arises. The obtained two-stage stochastic schedule computed by solving problem (5.18) with a given initial plant states is depicted in Figure 17.

Table 2: Connections product-plant and nominal efficiencies.

<table>
<thead>
<tr>
<th>Products:</th>
<th>p1</th>
<th>p2</th>
<th>p3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standby clean:</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Standby dirty:</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Small cleaning:</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Big cleaning:</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Figure 18: Two-stage stochastic schedule.

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. 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 $\sigma_{T_{\text{out}}} = 7^\circ\text{C}$ and $\sigma_{\text{p}} = 6$ 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 continuous ones and 152863 constraints. Solving problem (5.18) with CPLEX 24.8.5 in the same machine returns...
the 1%-gap optimal solution in 5 minutes. Figure 18 depicts the computed two-stage stochastic schedule for plants 2 and 8 (the rest are omitted due to space constraints).

For completeness, if Eq. (5.4) 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. The reader is referred to [17] for a more extended formulation and analysis of this case study, including a multiobjective optimisation setup and a multicriteria analysis of the resulting Pareto frontier.

5.2 The ammonia distribution network at INEOS Köln

In this section, the ammonia production plant (O7) as well as three subsequent processing plants of INEOS in Köln are considered. The topology of the setup is shown in Figure 19. The ammonia plant receives natural gas and hydrogen from an on-site cracker complex as reactant streams and fuel gas as the main energy carrier from the shared resource networks. The produced ammonia can be sent to two different storage types. One possibility is to compress the ammonia, cool it down and store it in the deep-cooled tanks T83 or T84. In the case of high demand, the stored ammonia can be reheated and used in the plants or sold to the market. The second possibility is to send the ammonia to the three spherical warm buffer tanks, which are considerably smaller than the deep-cooled tanks and which are used to buffer the production as well as the ammonia delivery and shipping facilities that transfer the ammonia over the borders of the site via ships or train vessels. The ammonia that is further processed on site is sent to three major ammonia consuming plants: The nitric acid plant (O4) and the two acrylonitrile plants (O8) and (O17). The nitric acid plant receives natural gas as its second main reactant and the two acrylonitrile plants receive propylene.

![Figure 19: Schematic topology of the INEOS Köln network with its processing plants.](image-url)
the plant models are of rather simple structure and composed of linear input-output mappings used for the
planning of the operation, the presence of numerous discrete decisions render this problem into a challenge,
both from the modelling as well as from the optimisation perspective. However, including the external influ-
ences into a site-wide ammonia distribution optimisation offers great potential to adjust the operation of this
complex system in order to have an overall energy and resource efficient operation.

5.2.1 Network modelling

In the following the modelling of the network itself, the processing plants, and the tanks is explained and a
discussion on the influence of external uncertainties follows.

Notation. The INEOS Köln ammonia network consists of several systems \( s \in S \). The systems are tanks \( t \in T \),
production plants \( p \in P \) and nodes \( n \in N \), which are connected by pipes \((c, s, s') \in \Pi\). The mass flows
through the pipes at time \( j \) are denoted by \( \dot{m}^{(j)}_{c,s,s'} \) with component \( c \in C \) going from system \( s \) to \( s' \). The
symbols and the naming convention used throughout the section are listed below.

SETS:

- \( C \): set of chemical components (index \( c \)).
- \( \Omega_f(s, m, m') \): set of transitions that require a fixed stay time.
- \( \Omega_i \): set of impossible transitions.
- \( \Omega_p(s, m, m') \): set of transitions that are associated with additional costs.
- \( IN_s, OUT_\hat{s} \): set of pipes leading to and leaving system \( s \) respectively.
- \( J \): set of time intervals in the optimisation horizon (index \( j \)).
- \( M_s \): set of modes of system \( s \).
- \( N \subset S \): set of nodes (index \( n \)).
- \( P \subset S \): set of plants (index \( p \)).
- \( PMR \subset P \): subset of plants with multiple independent reactors.
- \( \Pi(c, s, s') \): set of pipes for component \( c \) from system \( s \) to system \( s' \).
- \( \Pi_b \subset \Pi \): subset of pipes with restricted mass flows.
- \( S \): set of systems (index \( s \)).
- \( R_p \): set of reactors of plant \( p \) (index \( r \)).
- \( T \subset S \): set of tanks (index \( t \)).
- \( T_{\text{cold}} \subset T \): subset of cold storage tanks.
- \( T_{\text{sphere}} \subset T \): subset of spherical tanks.

DISCRETE VARIABLES:

- \( y^{(j)}_{s,m} \): binary which determines whether the system \( s \) is in mode \( m \) in interval \( j \).
- \( z^{(j)}_{s,m,m} \): binary which determines whether there is a transition from mode \( m \) to mode \( m' \) at system \( s \) in
  interval \( j \).
CONTINUOUS VARIABLES:
- \( m^{(j)}_{c,s,s'} \): stored mass of component \( c \) in system \( s \) in interval \( j \).
- \( \dot{m}^{(j)}_{c,s,s'} \): mass flow of component \( c \) from system \( s \) to system \( s' \) in interval \( j \).
- \( \dot{m}_{\text{prod},p}^{(j)} \): production of the main product in plant \( p \) in time step \( j \), equal to the associated \( \dot{m}^{(j)}_{c,s,s'} \).

PARAMETERS:
- \( \text{lb}_{c,s,s',s'}^{(j)} \), \( \text{ub}_{c,s,s',s'}^{(j)} \): lower and upper bounds on the mass streams \( j \).
- \( \text{lb}_{c,t}^{(j)}, \text{ub}_{c,t}^{(j)} \): lower and upper bounds on the mass the tanks.
- \( \alpha^{(j)}_{c,s,s',s'} \): price of component \( c \) in the associated stream in interval \( j \).
- \( \rho^{(j)}_{s,m,m',s'} \): price of transition of system \( s \) from mode \( m \) to mode \( m' \) in interval \( j \).
- \( \gamma_t \): prioritisation penalty of tank \( t \).
- \( \hat{\vartheta}^{(j)}_{a} \): estimated ambient temperature in degree Celsius in interval \( j \).
- \( \vartheta_{\text{LB}}, \vartheta_{\text{UB}} \): lower and upper extreme temperatures for the estimation of the shutdown or start up times.
- \( \tau_{\text{LB}}, \tau_{\text{UB}} \): lower and upper extreme times required for shutdown or start up.
- \( \Delta t \): interval length in hours.
- \( \tau_f \): temperature independent time required for a shutdown or start up.
- \( \tau_v^{(j)}(\hat{\vartheta}^{(j)}_{a}) \): temperature dependent time required for a reactor start up.

Nodes

The nodes \( N \) are the places where streams are joint or split. The mass balance of the incoming streams \( IN_n \) and the outgoing streams \( OUT_n \) has to hold in every time interval \( j \in J \) and for all nodes \( n \in N \). It can be expressed as:

\[
\sum_{(c,s,s') \in IN_n} \dot{m}^{(j)}_{c,s,s'} = \sum_{(c,s,s') \in OUT_n} \dot{m}^{(j)}_{c,s,s'} \forall n \in N, j \in J \quad (5.19)
\]

Many streams on site are bounded by technical limitations and all mass streams are modelled as positive streams, i.e., the flow direction is fixed. Thus, the following constraints are formulated:

\[
0 \leq \dot{m}^{(j)}_{c,s,s'} \forall (c,s,s') \in \Pi, j \in J \quad (5.20)
\]

\[
\text{lb}_{c,s,s'}^{(j)} \leq \dot{m}^{(j)}_{c,s,s'} \leq \text{ub}_{c,s,s'}^{(j)} \forall (c,s,s') \in \Pi^b, j \in J \quad (5.21)
\]

Plant models

The models used to plan the production of all plants are affine input-output models where the output stream is the one that determines all reactant and energy streams of the plant.

Plants with a single reactor are a non-dividable system of type plant. In plants with multiple reactors the entering mass flows are split in an internal node (see Figure 20). Each reactor \( r \in R_p \) has its own model (cf. "Reactor \( i \)" in Figure 20) and a plant subsystem (cf. "Plant" in Figure 20) with the name of the overall plant accounts for static terms independent of the number of active reactors. The subsystem outputs are added in a second internal node. Each plant type system has four discrete states on, off, startup, shutdown. It is possible
to get from on to off via the transition mode shutdown and to get back to on via startup. Since the plants or the individual reactors cannot be shut down or started up arbitrarily fast, there are fixed stay times for the states shutdown and startup, i.e., a fixed number of intervals (cf. subsection 5.2.1). Some of the streams on site vary depending on the ambient temperature. Thus, the temperature dependent (sub-)plant in- and outputs are calculated from the main product output \( \dot{m}_{\text{prod},p}(j) \) with affine models in:

\[
\dot{m}_{c,s,s'} = (a_0 + a_1 \cdot \vartheta) y_{p,on} + a_1 \cdot \dot{m}_{\text{prod},p}(c,s,s') \forall (c,s,s') \in \text{IN}_{p} \cup \text{OUT}_{p}, p \in \mathcal{P} \cup \mathcal{R}, j \in \mathcal{J} \tag{5.23}
\]

Where the product of the constant term with the binary \( y_{p,on} \) ensures that the plant does not consume energy or resources, if the complete plant is switched off.

Finally, the constraints

\[
y_{p,on} \leq \sum_{r \in \mathcal{R}_{p}} y_{r,on} \forall p \in \mathcal{PMR}, j \in \mathcal{J} \tag{5.24}
\]

\[
y_{p,on} \geq y_{r,on} \forall p \in \mathcal{PMR}, r \in \mathcal{R}_{p}, j \in \mathcal{J} \tag{5.25}
\]

guarantee that the static consumption plant model is only active if at least one of the reactors is active, i.e., if all reactors are switched off then the plant is switched off too. If at least on reactor is on then the plant is on too.

### Tanks

The tanks are modelled as difference equations. All incoming streams augment the level, all outgoing streams decrease it. The mass is balanced over the time interval \( \Delta t \):

\[
m_{c,t}^{(j+1)} = m_{c,t}^{(j)} + \Delta t \cdot \left( \sum_{(c,s,s') \in \text{IN}_{t}} \dot{m}_{c,s,s'}^{(j)} - \sum_{(c,s,s') \in \text{OUT}_{t}} \dot{m}_{c,s,s'}^{(j)} \right) \forall c \in \mathcal{C}, t \in \mathcal{T}, j \in \mathcal{J} \tag{5.26}
\]

Each of the tanks has lower and upper bounds on its content by:

\[
l_{b,c,t} \leq m_{c,t}^{(j)} \leq u_{b,c,t} \forall c \in \mathcal{C}, t \in \mathcal{T}, j \in \mathcal{J} \tag{5.27}
\]

The bounds result from technical limitations or by management decision to, e.g., to keep a certain level of inventory for the case of unforeseen shutdowns.

At the site of INEOS in Köln there are two types of tanks that differ in their operating modes:

**Cold storage tanks.** \( T_{\text{cold}} \subset \mathcal{T} \) cannot be filled and discharged at the same time. They have three modes: fill for filling, disc for discharging and idle if neither is done. A mode-dependent reformulation of Eqs. (5.21)-(5.22) for inlets and outlets of these tanks is used in:

\[
l_{b,c,s,s'} \cdot y_{t,fill}^{(j)} \leq \dot{m}_{c,s,s'}^{(j)} \leq u_{b,c,s,s'} \cdot y_{t,fill}^{(j)} \forall (c,s,s') \in \text{IN}_{t}, t \in \mathcal{T}_{\text{cold}}, j \in \mathcal{J} \tag{5.28}
\]

\[
l_{b,c,s,s'} \cdot y_{t,disc}^{(j)} \leq \dot{m}_{c,s,s'}^{(j)} \leq u_{b,c,s,s'} \cdot y_{t,disc}^{(j)} \forall (c,s,s') \in \text{OUT}_{t}, t \in \mathcal{T}_{\text{cold}}, j \in \mathcal{J} \tag{5.29}
\]
There are no restrictions on changing between the three modes.

**Spherical tanks.** $T_{\text{sphere}} \subset T$ can be filled all times. However, they have outlets to different nodes. They can discharge towards either of following processes (**discp**), ships (**discs**), train vessels (**disct**) or stay idle (**idle**). Like in Eq. (5.28), the mass flows are set to zero if the corresponding discharge mode is inactive. There are no forbidden mode changes.

### Systems operating modes

All systems except nodes have several operating modes $\mathcal{M}_s$. Following the formulation of [14], binary variables $y^{(j)}_{s,m}$ for each mode and $z^{(j)}_{s,m,m'}$ for mode transitions are used. In Eq. (5.30), the systems are restricted to one mode at a time and in Eq. (5.31) the transitions are formulated.

$$
\sum_m y^{(j)}_{s,m} = 1 \quad \forall m \in \mathcal{M}_s, s \in \mathcal{P} \cup \mathcal{T}, j \in J
$$

(5.30)

$$
\sum_{m'} z^{(j)}_{s,m,m'} - \sum_{m'} z^{(j)}_{s,m',m} = y^{(j)}_{s,m} - y^{(j-1)}_{s,m} \quad \forall (m, m') \in \mathcal{M}_s, j \in J
$$

(5.31)

The transitions that are forbidden are prohibited by forcing the respective transitions to zero:

$$
z^{(j)}_{s,m,m'} = 0 \quad \forall (s, m, m') \in IMT, j \in J
$$

(5.32)

The transitions $(s, m, m') \in \Omega_f$ require the plant to stay in the new mode for a fixed time $K_{s,m,m'}^{fix(j)}$ before the next transition. These constraints become necessary, since shutting down a plant or starting it up cannot be done instantaneously.

$$
y^{(j)}_{s,m'} = K_{s,m,m'}^{fix(j)} - 1 \sum_{\theta=0}^{K_{s,m,m'}^{fix(j)} - 1} z^{(j-\theta)}_{s,m,m'} \quad \forall (s, m, m') \in \Omega_f, j \in J
$$

(5.33)

For the different transitions from the different modes the values are either fixed values based on experience, or they are dependent on the ambient temperature. For all shutdown procedures the estimated required time is computed by

$$
K_{s,m,m'}^{fix(j)} = \left(\tau_v(\hat{\vartheta}_a) + \tau_f\right) \Delta t^{-1} \quad \forall (s, m, m') \in \Omega_{\text{reactors}} \subset \Omega_f, j \in J.
$$

(5.34)

The part of the transition time that is dependent on the ambient temperature $\vartheta_a$ needs to be estimated from either experience or from the weather forecast. Obviously, this depends on the length of the horizon, since there is no accurate weather forecast for the next months. However, it perfectly makes sense to use average temperatures for night and day times to optimise the schedule with respect to those. Based on experience, the extreme times for starting up and shutting down are known for the specific reactors. At low ambient temperature $\vartheta$ the required time is $\tau$ and at high ambient temperature $\overline{\vartheta}$ the required time is $\overline{\tau}$. Thus, the variable estimated time $\tau_v(\hat{\vartheta}_a)$ is computed by

$$
\tau_v(\hat{\vartheta}_a) = \frac{\overline{\tau} - \tau}{\overline{\vartheta} - \vartheta} (\hat{\vartheta}_a - \vartheta) + \tau \quad \forall j \in J.
$$

(5.35)

### External sinks and sources

There are several ingoing and outgoing streams crossing the system boundaries that are depicted in Figure 19. All reactant streams as well as the required steam and electricity are assumed to be available at sufficient quantity. They are provided by adjacent processes and have lower and upper bounds on the quantity per hour. In addition, ammonia is both bought and sold over the boundaries of the site via ships and trains. The filling and discharging of ships and trains at the same time is not possible and limited to specific technical limitations depending on the means of transport, i.e., ship or train. These logistic constraints add significant complexity to the overall systems, since the operation of the plants can be hindered by back propagating bottlenecks such...
as full storage tanks for ammonia in case a selling action cannot be performed. The assignment of the delivery and shipping windows and their respective mass flow rates across the boundaries of the site can either be set manually for fixed days or it can be up to the optimiser to arrange the schedule. Needless to say that this adds extra and sometimes unnecessary degrees of freedom which increases the solution time of the model by orders of magnitude.

**External uncertainties**

Apart from logistic constraints that might be uncertain, the prices for electricity and natural gas from the grid as well as the ambient temperature vary during the day or seasonal. In Figure 21, the electricity and the gas prices scaled to their maximum value in the considered interval are plotted. It can be seen that the electricity price shows besides the ordinary day-night and weekdays-weekend pattern some hours in which the price is very far away from the average. These situations can be, for instance, holidays or weekend days where renewable electricity sources are at high levels and few consumers are operating at the upper bound. During these hours, it can be very beneficial for the site to operate equipment with energy consuming parts such as the compressor shown in Figure 19.

![Figure 21: Scaled energy charts. The prices are given relative to the maximum price in the optimisation horizon for confidentiality reasons.](image)

**5.2.2 Investigated scenario and implementation**

To determine the quality of the site-wide optimisation model, a problem instance for an example scenario provided by the industrial partner INEOS in Köln is formulated and solved.

**The optimisation problem**

The optimisation problem in a compressed form is stated as:

$$\min_{m, \dot{m}, y, z} f(m, \dot{m}, y, z) \quad \text{subject to: Eqs. (5.19) – (5.35)}$$

The objective function is formulated over the scheduling horizon \( \forall j \in J \). It contains the following contributions and penalty terms:

$$f(m, \dot{m}, y, z) = - \sum_{s \in P \cup T} f_{\text{profit},s} + \sum_{s \in P \cup R \cup p} f_{\text{shutdown},s} + \sum_{t \in T} f_{\text{priorisation},t}$$
The objective value contains the economic performance of the site over the horizon and includes additional penalties to increase the computational performance and prioritise current daily operating practices. The terms contributing to the objective are discussed in the following.

**Systems profit.** For all plant and tank systems the profit is calculated as the difference of exported and imported mass weighted with the (internal) price of the specific component:

\[
f_{\text{profit},s}(m, \dot{m}, y, z) = \sum_{j \in J} \left( \sum_{(c,s,s') \in \text{OUT}_s} \dot{m}_{c,s,s'}^{(j)} \cdot \alpha_{c,s,s'}^{(j)} - \sum_{(c,s,s') \in \text{IN}_s} \dot{m}_{c,s,s'}^{(j)} \cdot \alpha_{c,s,s'}^{(j)} \right).
\]

Where \(\alpha_{c,s,s'}^{(j)}\) is the price of component \(c\) for that specific stream. It is important to note that this price is time variant for electricity and natural gas. Especially the price of the import and export of \(\text{NH}_3\) into or from cold storage tanks via the compressors or a heater uses a price where the power consumption of these processes is included. When summing up the profit terms for all tanks and plants the internal sales cancel out and only transfers across the site boundary appear in the total balance.

**Shutdown and start up cost.** As shutting down and starting up a reactor or plant does not only require a specific time (see Section 5.2.1) but is also associated with costs, the prices for these procedures are taken from experience values and are included in the cost function. This is done using the binary transition variables and individual prices \(\beta_{s,m,m'}^{(j)}\) for all transitions that are leading directly to additional costs.

\[
f_{\text{startup, shutdown},s}(m, \dot{m}, y, z) = \sum_{j \in J} \sum_{(s,m,m') \in \Omega_p} \dot{z}_{s,m,m'}^{(j)} \cdot \beta_{s,m,m'}^{(j)}.
\]

**Priorisation of tanks.** To avoid symmetric solutions of the ammonia distribution to tanks of same type, the usage of each tank is weighted in:

\[
f_{\text{priorisation},t}(m, \dot{m}, y, z) = \sum_{j \in J} \sum_{m \in M_t} \dot{y}_{t,m}^{(j)} \cdot \gamma_{t}.
\]

The weighting factors are chosen to match daily operating practice where possible.

### Formulation of an optimisation scenario

The model can be optimised with respect to different questions or target values, which depend on the choice of the industrial partner INEOS in Köln. One of the possible scenarios considers a planned shutdown of the ammonia plant \(O7\) in the next month. Consequently, all the plants have to be operated such that for the next month, a sufficiently large amount of ammonia is present on site to ensure a seamless operation of the subsequent processes. With a planned schedule of purchase and sale of ammonia, the tank levels and the operating level of the plants have to be optimised. The schedule of the logistics is given in Table 3.

For this contribution, a set of data for May of a recent year has been taken in order to be able to compare the optimised site schedule with the operation of the site that was realised in that month. Hence, an estimation of the temperature as well as a prediction of the electricity and gas price is not necessary, as they are taken from the records (see Figure 21). The schedule shown in Table 3 has been implemented, such that the external streams from and to the trains and ships have been fixed during the time that is required to process the loading or unloading. For example, a train vessel can be filled at a given maximum flow rate \(\dot{m}_{\text{train, max}}\) and a sample train requires 5.6 hours. Then during the first five hours the filling station is set to the maximum filling rate and in the sixth hour the rest is being processed, which is of course an approximation caused by the length of the optimisation interval of one hour.

Furthermore, the initial levels of the tanks as well as the initial operating states of the plants and reactors are fixed manually. The monthly targets for the subsequent processes are fixed by hard constraints on the processed ammonia quantity per month, which can be inferred from the product quantity that has to be processed during the month. For one ACN plant this can be formulated as

\[
m_{O8, \text{NH}_3, \text{target}} = \sum_{j \in J} \dot{m}_{\text{NH}_3,T13X,O8}^{(j)} \cdot \Delta t,
\]
Preliminary report on optimisation methods for large plants with discrete and continuous degrees of freedom

Table 3: Information for the logistic constraints of the optimisation problem (day, processing hours).

<table>
<thead>
<tr>
<th>Train deliveries</th>
<th>Ship deliveries</th>
<th>Train sales</th>
<th>Ship sales</th>
</tr>
</thead>
<tbody>
<tr>
<td>(15,6)</td>
<td>(1,12)</td>
<td>(2,3)</td>
<td>(16,8)</td>
</tr>
<tr>
<td>(16,8)</td>
<td>(8,12)</td>
<td>(3,9)</td>
<td>(17,19)</td>
</tr>
<tr>
<td>(17,19)</td>
<td>(13,12)</td>
<td>(4,3)</td>
<td>(18,19)</td>
</tr>
<tr>
<td>(18,19)</td>
<td>(21,12)</td>
<td>(6,9)</td>
<td>(19,6)</td>
</tr>
<tr>
<td>(19,6)</td>
<td>(27,12)</td>
<td>(9,3)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

where $n_{\text{NH3,T13X,O8}}^{(j)}$ is the mass flow of ammonia from the spherical tanks to the ACN plant O8 (Figure 19).

To compare the solution of the optimiser with what was done during the investigated period at INEOS in Köln, in addition to the initial tank levels for each individual ammonia tank the aggregated ammonia on site at the end of the optimisation horizon has been fixed by

$$m_{\text{NH3, on site}} = \sum_{t \in T} n_{\text{NH3},t}^{(j=H+1)},$$

where $n_{\text{NH3},t}^{(j=H+1)}$ is the mass of ammonia at the end of the horizon in tank $t$.

Implementation

The use case is implemented with the help of the mathematical modelling package JuMP [26], which is a package written in Julia [27]. This implementation ensures independence of the chosen solver, since JuMP interfaces many open-source and proprietary solvers that are available. For this contribution CPLEX 12.8.0.0 is used, because it shows the best performance with respect to the solution time.

5.2.3 Illustrative results

The result of the optimisation regarding operating decisions on the plants and tanks is shown in Figure 22 and Figure 23.

The ammonia plant (O7) is shut down in the beginning and restarted after a downtime of 158 hours or 6.5 days. Afterwards, it is operated at maximum capacity most of the time with some short intervals of production at lower or minimal capacity. The nitric acid plant (O4) is shut down for an interval of 33 hours about one day after almost 6 days of operation at maximum capacity and then operated at full capacity interrupted by periods of minimum capacity utilisation which are concentrated to the last eleven days of the horizon. The result shows operation at full capacity of acrylonitrile plant O17 for most intervals in the horizon. In the period of constant lowered production starting after approximately 500 hours, one reactor is inactive while the others remain at maximum production rate. Afterwards, the reactor is restarted and the plant keeps running at full capacity with only short periods of lowered production until the end of the horizon. The other acrylonitrile plant (O8) is
operated at full capacity for almost four weeks and then completely shut down. A single reactor of that plant is reactivated for a very short production interval and immediately shut down afterwards.

From Figure 23 it can be seen that the cold storage tank T83 is almost completely emptied within the first week and shortly after it is filled until the end of the horizon. The other cold storage tank T84 is kept idle almost constantly. The spherical tank T134 is kept idle at maximum filling state most of the time. The other two spherical tanks T135 and T138 are in constant use for sending the ammonia feed to following processes or to the scheduled train sales as well as for taking up the ammonia that is arriving via ship and train. The filling state shows utilisation of the whole permitted range with periods of quick net filling or discharging. In the end both are filled to the maximum.

The shutdown of the ammonia production plant in combination with the depletion of T83 shows that reaching the target quantity of stored NH$_3$ at the end of the month would have been possible with less purchase from external sources. The plant is inactive in the beginning of the month where the highest gas prices are effectual (see Figure 21). The presented solution shows that the affine models favour a maximised production rate which is applied in the intervals with the most beneficial circumstances while in the intervals in between the plant is run at minimal utilisation because a complete shutdown is too expensive for short downtimes. This behaviour leads to large differences in the set points between two adjacent intervals which in general is
problematic in large plants as reaching steady state may take several hours. Further constraints or penalties derived from operating practice may significantly improve this behaviour in the future. Additionally, with this kind of concentrated periods of full capacity utilisation, unexpected events may strongly impact the plant’s ability to supply because there is no capacity left to buffer lost outputs. The three other plants included in this optimisation show a similar behaviour with the month’s production goal scheduled to a minimised number of intervals and maximised production rates. This shows a need for the same modifications as mentioned before. However, the main difference is that especially plant O8 and O17 show that for consuming ammonia the earlier intervals of the horizon are more suitable and reduced production rates mainly appear in the second half of the month. During this time the tanks are filled with higher priority to reach the target amount of ammonia.
6. First conclusions and ongoing work

The methodologies summarised in this report have been employed to address a mixed production-maintenance scheduling problem for the evaporation system at Lenzing AG site, and the optimisation problem of the ammonia distribution network at INEOS in Köln. The evaporation plants are 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 an RTO needs to be extended to the network scheduling in a computationally tractable way, also considering uncertainty. A discretisation in days and a modification of the PSTN 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 optimisation approach. In this way, less conservative robust solutions are obtained by computing different schedules for some expected uncertainty realisations 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.

The approach is tested in simulation with several instances of the evaporation system. The results were promising so that the two-stage stochastic approach can be progressively extended to the whole system, or to other facilities of the site like the heat-recovery network, whose heat exchangers suffer similar fouling problems. 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.

The result of the proposed optimisation for the INEOS Köln site indicates beneficial strategies for the resource distribution in the investigated scenario. The distribution of stored ammonia to the tanks is highly influenced by the penalties introduced in Eq. (5.40) to avoid symmetric solutions. Still it can be seen how T138 is (partially) emptied to be able to receive a new shipment via ship or train multiple times. The same holds for T135 but in a less drastic way as only train shipments are serviced via this tank. The final state shows that warm ammonia is preferred by the current objective function as it does not generate additional costs for compressing if the electricity price is positive. Whenever O7 is running and the electricity price is negative the compressors are used to profit from this situation. However, in the interval with the lowest electricity prices in the beginning of the month, O7 is shut down since the comparably high gas prices have a stronger influence on the site wide optimisation.

It could be shown that the site scheduling at INEOS petrochemical production site can be formulated as a MILP problem that contains the plant models with switching operating modes and fixed stay times, all relevant buffer and storage tanks for ammonia, important technical equipment such as compressors as well as the adjacent logistics of the ammonia network. Depending on the complexity of the logistic constraints and the level freedom the solver is granted to have, the solution time of the problem can range from minutes to hours. The results obtained from the solution to the problem are in accordance with how the site has been operated. However, some deviations are present where technical constraints are missing or daily operating practice is not yet formulated as a constraint or limitation in the problem.

Since this proof of concept can be regarded as a success, it is possible to extend the scope of the use case to incorporate more plants on sites with additional shared resource distribution grids. It is also possible to extend the logistics of the network by flexible ship or train schedules. If the complexity of the MILP reaches a level where a centralised solution might become intractable, one might consider a decomposition of the problem by, e.g., modelling the spherical tanks as a subproblem that is linked to the overall optimisation problem. With respect to distributed (market-based) coordination, one could consider linking the ammonia network as a subsystem to other large systems on site such as the cracker complex. Part of the future work should also include a proper choice of the temperature and electricity price estimation.
Bibliography


Preliminary report on optimisation methods for large plants with discrete and continuous degrees of freedom


