



Universidad de Valladolid

Programa de Doctorado en Ingeniería Industrial

Efficient scheduling of batch processes in continuous processing lines

Presentada por **Carlos Gómez Palacín**
para optar al grado de Doctor por la
Universidad de Valladolid

Dirigida por:
César de Prada, catedrático de universidad

*A mi mujer, Lidia,
sin la cual esta tesis se habría
terminado dos años antes*

Acknowledge

First and foremost, I would like to express my gratitude to my advisor, César de Prada, full professor. It has been a real privilege to be able to work with him. His passion for teaching, and his unwavering willingness to keep learning and improving, suppose a true motivation. He has definitely left a lasting impression on me as a researcher and, more important, as a person. I'm really grateful for his guidance and for sharing his vast knowledge in the field of automatic control and systems engineering, which I have enjoyed learning as much as I could during the several discussions we had. This thesis would never had been completed without his support, his enthusiasm, and his incredible patience.

Special thanks to Dr. José Luis Pitarch, with whom I have shared office since my first day, for his nice patience and warm friendship, and his focus on progression. I cannot thank him enough for his unselfish and unfailing desire to help. For the many hours he has spent discussing problems (in the office and also having a drink) and helping me tackling every hurdle I faced. His comments were significance for completing this thesis. It has been a true privilege to share office with him.

Further, I would like to thank all the invited lecturers, that have increased my knowledge in different topics related to the process industry. It has been an honor to assist to their seminars and I have learned something from each and every one of them. Specially, to professor Carlos A. Méndez, that opened the door for me to the challenging field of scheduling, and had helped me since. And to Antonio A. Alonso, full professor, and Dr. Carlos Vilas, who had helped me during the last part

of the thesis.

Thanks to the industrial partners, who helped me to implement the developments of the thesis with real plant data. In particular to Chris Jäsch (with whom I have a couple of beers to spare), Daniel Adrián, Miguel A. López and Agustín Vilas.

Finally, I would like to thank my family, i.e., my parents, my sister-in-law, my in-laws, *et cetera*. My brother, Dr. Luis Gómez Palacín, that likes so much to share his miscellaneous knowledge in a vast set of fields, as for example the plot of thrillers. And my lovely and beloved wife, Lidia Padierna, who has encouraged me during hard times with an endless support.

Summary

English

This thesis focuses mainly on the development of efficient formulations for scheduling in industrial environments. Likewise, decisions over the processes more related to advanced process control or production planning are included in the scheduling; in this way, the schedule obtained will be more efficient than it would be if the additional restrictions were not considered. The formulations have to emphasize obtaining online implementations, as they are planned to be used in real plants.

The most common scheduling problems handled in the industrial environments are: the assignment of tasks to units, the distribution of production among parallel units and the distribution of shared resources among concurrent processes. Most advances in this work are the result of a collaborative work.

Two industrial case studies are presented: the first is the optimization of the use of resources in a network of industrial evaporation plants where the workload is divided between parallel evaporation plants; and the second is the optimal schedule of batch processes that share a necessary resource during processing.

In the case of the evaporator network, it is modeled following a methodology for obtaining gray-models using real plant data and data reconciliation techniques. Although the plant works in a steady state, the degradation of the equipment is included entailing its dynamics. This reduction in equipment efficiency is added to the problem of optimizing the work point. When degradation is excessive, a maintenance job is required to recover efficiency, this is achieved by performing

cleaning tasks. So, the scheduling of these cleaning tasks is also included in the optimization problem, deciding not only when to perform them, but also the type of cleaning task.

Once the operation of one evaporation plant has been optimized, the problem of scheduling the whole evaporation network arises. The load to every evaporation plant must be defined, in terms of flows and type of product. Each evaporation plant has a different efficiency, and some combinations between products and units are not permitted. In addition, the problem of degradation in equipment is maintained, so that the cleaning tasks scheduling must also be carried out, which includes the selection of type and selecting when to perform them. It should be taken into account that the productivity of the section should not be affected when a plant is undergoing maintenance, so its workload must be distributed among the rest of allowed evaporation plants.

The problem is handled using mathematical programming, and modeling the system as a mixed-integer linear programming problem, which allows to obtain optimal solutions in a reasonable time. There are factors that can affect the productivity of the units, such as the outside temperature or the workloads planned by the management departments. These external factors present uncertainty that is considered into the scheduling problem using a technique commonly used in the advance control field but not frequently applied in scheduling problems, as is stochastic two-stage approach. It allows to tackle uncertainty problems with a less conservative way than the classic robust approach.

Finally, this thesis faces the problem of organizing batch processes that share certain resources, and have to be performed between continuous production processing lines. First, the problem of batching the products that arrive in a discrete flux, and the subsequent scheduling of the processes has been solved to minimize overall manufacturing time (makespan). Then, restrictions on the use of resources have been incorporated.

Two formulations have been developed to deal with these types of problems. In the first one resource availability is limited. Real consumption behavior during its

processing time is modeled, and global restrictions are added to avoid exceeding a maximum bound set by the plant architecture. In the second case, instead of adding a restriction on consumption, the effect of concurrent consumption is considered. If several processes are executed concurrently consuming the same resource, both duration will be affected. In this case, the duration of the processes is what should be modeled to be included in the task scheduling problem.

In both options, a tool with a closed-loop approach has been developed, similar to that used in advanced control techniques, which allows to react to deviations from the planning and reduces the resolution times, allowing the implementation in plants.

Español

Esta tesis se enfoca, principalmente, en el desarrollo de formulaciones eficientes para la programación de las tareas de procesamiento en entornos industriales. Así mismo, se busca la integración de la toma de decisiones más relacionadas con el control avanzado de los procesos o con la planificación de la producción; de esta manera, la programación obtenida será más eficiente de lo que sería si no se tuvieran en cuenta las restricciones adicionales. Las metodologías diseñadas enfatizan la resolución rápida de los problemas, dada su prevista implementación en plantas reales.

Se tratan los problemas de programación de tareas más comunes en las industrias: la asignación de tareas a unidades, la distribución de la producción entre equipos paralelos y el reparto de recursos compartidos entre procesos concurrentes, principalmente. Los avances obtenidos durante el desarrollo de esta tesis son fruto de trabajo colaborativo, y no deberían entenderse como un mérito individual del autor.

Se presentan dos casos de estudio industriales donde se implementan las metodologías propuestas para comprobar su efectividad: el primero consiste en la optimización del uso de recursos en una red de evaporadores industriales donde se reparte la carga de trabajo entre plantas de evaporación paralelas; y el segundo consiste en la organización óptima de procesos de esterilizado que comparten un recurso necesario durante el procesamiento.

En el caso de la red de evaporadores, estos son modelados siguiendo una metodología de obtención de modelos grises usando datos reales de planta y reconciliación de datos. Aunque la planta trabaja en estado estacionario, se incluye la degradación de los equipos con el uso que conlleva una dinámica. Esta reducción en la eficiencia de los equipos se añade al problema de optimización del punto de trabajo. Cuando la degradación es excesiva se requiere la realización de tareas de mantenimiento para recuperar eficiencia (distintas tareas de limpieza). En un segundo punto se realiza la optimización de la programación de estas tareas de limpieza, donde se debe decidir el momento de realizarlas y el tipo de las mismas.

Con el trabajo de una planta de evaporación optimizado, se plantea el problema de la organización de toda la sección de evaporación, donde se debe asignar la carga a cada planta de evaporación con respecto al tipo de producto y la cantidad de flujo. Cada planta de evaporación tiene una eficiencia diferente, y no todas las asignaciones producto-equipo se pueden realizar. Además, se mantiene el problema de la degradación en los equipos, con lo que la organización de las tareas de limpieza también debe realizarse, lo que incluye la selección del tipo de limpieza y la organización temporal de las mismas. Se debe tener en cuenta que la productividad de la sección no se debe ver afectada cuando una planta está siendo sometida a tareas de mantenimiento, por lo que su carga de trabajo se debe repartir entre el resto de unidades de evaporación accesibles.

El problema se plantea usando programación matemática como un problema de programación mixta-entera lineal, lo que nos permite obtener soluciones óptimas en un tiempo razonable. Así mismo, existen factores que pueden afectar la productividad de las unidades, como la temperatura exterior o las cargas de trabajo planificadas por los departamentos de gerencia. Estos factores externos presentan incertidumbre en cuanto a sus valores que se incorpora en el problema de programación usando una técnica común en el control, pero no aplicada en la programación, como es el enfoque estocástico en dos etapas, que permite afrontar problemas de incertidumbre de una manera menos conservadora que el enfoque robusto clásico.

Por último, esta tesis se enfrenta al problema de organizar líneas de producción continua y líneas de procesado discreto o por lotes que comparten ciertos recursos. Primero se ha resuelto el problema del loteo de los productos y la subsecuente programación de las tareas de producción para minimizar el tiempo de fabricación global. Después, se han incorporado las restricciones en cuanto al uso de recursos.

Se han desarrollado dos formulaciones para enfrentarse a este tipo de problemas. En la primera opción, la disponibilidad de recurso es limitada, por lo que la sincronización de los arranques de las unidades debe respetar restricciones globales de consumo. Se modela el consumo de los equipos durante su tiempo de proceso y se añaden restricciones globales a dicho consumo para evitar que se supere un

límite máximo fijado por la arquitectura de la planta. En el segundo caso, en lugar de añadir una restricción en el consumo, se tiene en cuenta que si varios procesos se ejecutan de manera concurrente consumiendo el mismo recurso, la duración de ambos se verá afectada. En este caso, la duración de los procesos es la que debe modelarse para incluirse en el problema de programación de tareas.

En ambas opciones se ha desarrollado una herramienta con un enfoque en lazo cerrado, similar al que se emplea en las técnicas de control avanzado, que permite reaccionar ante desviaciones frente a la planificación y reduce los tiempos de resolución, permitiendo la implementación en las plantas buscada.

Contents

| | |
|---|------------|
| List of Figures | xiv |
| 1 Introduction | 1 |
| 1.1 State of the art | 5 |
| 1.1.1 Mixed-integer linear programming | 10 |
| 1.2 Open fields | 13 |
| 1.3 Aims of the thesis | 14 |
| 1.4 Outline | 14 |
| 2 Maintenance scheduling of continuous processing lines | 16 |
| 2.1 Viscose plant | 17 |
| 2.1.1 Operation problems | 20 |
| 2.2 Objectives | 22 |
| 2.3 Methodology | 23 |
| 2.4 Evaporation process optimization | 28 |
| 2.4.1 Real-time optimization of one evaporation plant | 35 |
| 2.5 Evaporation network scheduling model | 38 |
| 2.5.1 Scheduling results | 43 |
| 2.6 Stochastic approach | 46 |
| 2.6.1 Stochastic scheduling solution | 49 |
| 2.7 Conclusions | 53 |

| | | |
|----------|--|------------|
| 3 | Real-time scheduling of batch processes | 55 |
| 3.1 | Industrial site | 57 |
| 3.2 | Objectives | 62 |
| 3.3 | Operation simulation | 63 |
| 3.4 | Methodology | 67 |
| 3.5 | Mathematical Model | 68 |
| 3.5.1 | Ordering the slots | 75 |
| 3.6 | Examples of execution instances | 77 |
| 3.6.1 | Implementation details | 77 |
| 3.6.2 | Gathering the carts | 78 |
| 3.7 | Conclusions | 80 |
| 4 | Scheduling of batch processes with shared resources | 82 |
| 4.1 | Case study | 84 |
| 4.2 | Objectives | 86 |
| 4.3 | Methodology | 86 |
| 4.4 | Mathematical Model | 87 |
| 4.4.1 | Consumption bound | 88 |
| 4.4.2 | Variable processing time | 96 |
| 4.4.3 | Reducing quality loss due to altered profiles | 100 |
| 4.4.4 | Integrating control in the scheduling | 102 |
| 4.4.5 | Considering quality loss | 106 |
| 4.5 | Results | 107 |
| 4.5.1 | Variable duration | 111 |
| 4.5.2 | Integrating control parameters | 115 |
| 4.5.3 | Uncertainty in the cart supply | 118 |
| 4.6 | Conclusions | 119 |
| | Main contributions and conclusions | 121 |
| | Bibliography | 127 |

List of Figures

| | | |
|------|---|----|
| 1.1 | Evolution from the automation pyramid | 3 |
| 1.2 | Main scheduling problems structures | 8 |
| 2.1 | Diagram of viscose production | 18 |
| 2.2 | Diagram of an evaporation plant | 20 |
| 2.3 | Recovery station connections | 21 |
| 2.4 | Connections between spinning units and evaporation plants | 22 |
| 2.5 | Diagram of cooling tower | 31 |
| 2.6 | Graphs representing the regression models | 34 |
| 2.7 | Estimated heat transmission coefficient | 35 |
| 2.8 | Control evolution applying dynamic RTO | 37 |
| 2.9 | Surrogate model representation | 39 |
| 2.10 | Automata representation of the active states | 40 |
| 2.11 | Optimal deterministic schedule example | 46 |
| 2.12 | Representation of the two-stage decision variables | 47 |
| 2.13 | Illustrative calculations for the similarity index | 49 |
| 2.14 | Uncertainty evolution example | 50 |
| 2.15 | Two-stage stochastic schedule example | 52 |
| 2.16 | Representation of the efficiency with respect to robustness | 53 |
| 3.1 | Allocation problem scheme | 57 |
| 3.2 | Estimated temperature inside the cans | 60 |
| 3.3 | Autoclave deployment scheme | 61 |

| | | |
|------|--|-----|
| 3.4 | Screenshots of SIMIO | 65 |
| 3.5 | Close-up screenshots of SIMIO | 66 |
| 3.6 | Gantt chart of gathering solution | 79 |
| 4.1 | Allocation problem scheme with shared resources | 83 |
| 4.2 | Comparison of profiles relative to shared resources availability | 85 |
| 4.3 | Duration and consumption profiles vs. temperature set point | 85 |
| 4.4 | Piece-wise linear function example | 91 |
| 4.5 | Time basis relation | 93 |
| 4.6 | Continuous consumption computation example | 95 |
| 4.7 | Graphical representation of the influence among four slots | 97 |
| 4.8 | Profiles with fixed duration of maintenance stage | 100 |
| 4.9 | Profiles with fixed final lethality | 101 |
| 4.10 | Feasible area approximation for different references | 104 |
| 4.11 | Piece-wise approximation to real consumption example | 107 |
| 4.12 | Schedule solution with maximum consumption boundary | 110 |
| 4.13 | Example of variable heating stage duration | 112 |
| 4.14 | Schedule solution with variable duration | 114 |
| 4.15 | Feasible regions approximation | 115 |
| 4.16 | Schedule solution integrating control | 117 |
| 4.17 | Solution times with different uncertainty distributions | 118 |
| 4.18 | Solution times using different solvers | 119 |

Chapter 1

Introduction

Nowadays the industrial sector is immersed in a deep renovation. The incursion of new technologies has opened the door to the implantation of a large number of advanced sensors, promoting the insertion of the Internet of Things, becoming what has been called Industrial Internet of Things (IIoT)(Gilchrist, 2016).

Even though, this greater capacity for data collection represents an advance in terms of the knowledge that can be obtained from an industrial process, it has to come accompanied by tools that transform this raw data into practical information, as it is being done within the contemporary branch of knowledge around Big Data (McAfee et al., 2012), or using different modeling tools combined with fitting methods as data reconciliation (Narasimhan and Jordache, 1999; Albuquerque and Lorenz T. Biegler, 1996). But having better plant information is not enough, and the core of the improvement resides in using the data and models in decision making for control and plant optimization at all levels.

All these advances are giving rise to an industrial revolution, predicted even before it has been fully implemented, the Industry 4.0, which has the purpose of automate the decision making procedures by a better knowledge of the plant status in the models (*Industria Conectada 4.0* 2019).

Typically, the decision making system of a plant is divided in four different levels, each one more complex than the previous one and operating on larger time scales. The bottom layer contains the systems in charge of maintain the observed variables in their range or set point, by modifying the controlled ones, as it can be PIDs, SCADAs, PLCs, *et cetera*.

In the next level, there are the procedures that seek for suitable set points or ranges, as model predictive control (MPC) among others (Burns, 2001; Cesar de Prada, Daniel Sarabia, et al., 2017; Cesar de Prada and Pitarch, 2018; Santos et al., 2020).

Over those, in the supervision layer, the manufacturing execution system (MES) provides the operation points for long time horizons, usually around few days, and the scheduling that has to be fulfilled in order to serve the clients reducing costs due to delays or storage.

And, on top, the enterprise resource planning (ERP) which focuses on the economic planning of the factory or enterprise. It makes decisions about what products to produce, what resources to invest in the production and when to start it.

Generally, referring to what is known as process industry, the bottom layers, that are controlled by the operators in order to fulfill the demands set by the MES and ERP systems, work in a continuous way; meanwhile the top layers have to choose between different working policies.

The lower layers of this pyramid have been the focus of several studies since more than fifty years (Skogestad, 2007; Åström et al., 2006; Boyd and Vandenberghe, 2004).

The more we climb the pyramid, the bigger and more complex the systems are; therefore, they are less automated, and the human presence is a must.

Furthermore, the industrial sites in Europe are forcing themselves to reduce their environmental footprint. However, the final production cost cannot be increased within the actual market conditions, with a fierce rivalry presented by the eastern industrial sectors. The efficiency in the use of natural resources must be performed without losing competitiveness. The objective function in the advanced control

techniques begin to include some economic cost terms (D. Sarabia et al., 2012), v.gr., economic model predictive control (eMPC).

Nevertheless, the fact that the units of one site work in their individual optimal operation points, does not necessarily make the whole plant to work optimally (Grossmann, 2012). The links between the different processing sections, treated as boundary values, have to be included in the global optimization problem. The goal becomes to obtain the enterprise-wide optimization. Thus, the previously rigid limits of the pyramid layers, become blurred and horizontal and vertical integration of layers, besides automation of the higher levels, becomes an important target. The well defined and isolated control departments are no longer a model to follow.

Furthermore, new IIoT technologies will replace the old control equipment. New controllers will no longer be physically delimited, and will become dispersed through the plant, or even implemented in the cloud. The old bounds between layers are becoming difficult to outline.

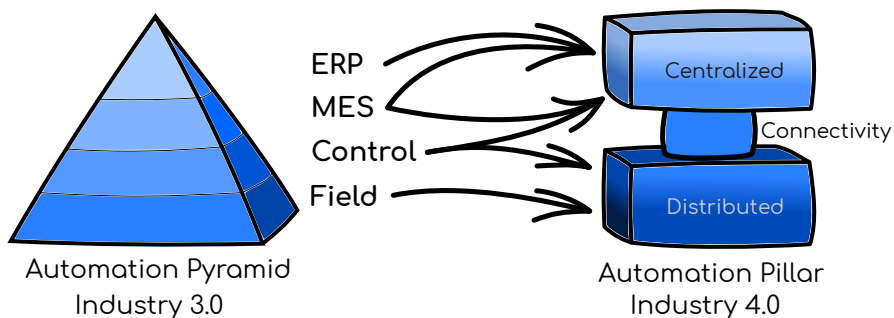


Figure 1.1: Evolution from the industry 3.0 pyramid based architecture (left) to the new industry 4.0 pillar deployment (right). In the new structure, the number of field devices has increased, the control is divided into distributed control devices and some centralized advance control systems. The supervision tasks are integrated in the ERP

The Industry4.0 aims on automating supervision and management tasks, previous third level (MES), and incorporates the advance control models in the overall optimization. The control pyramid becomes more similar to a control pillar, see

figure 1.1.

Supervision and management tasks are, in general, scheduling problems. The job orders determined by the superior ERP system, to fulfill the external demands, have to be organized. To perform the optimal distribution of the work orders, some details from the control layer should be considered. This global problem becomes more complex to solve, however the potential benefits encourage the study.

Scheduling problems have to answer four principal objectives: the selection of the jobs to perform; the assignment of the jobs to equipment; the sequencing of the orders; and finally, the start time for every job (Pinedo, 2008; Brucker, 2007).

There is no one unique approach to solve scheduling problems, which is very dependent on the type of problem considered and the environment the purpose of the scheduling. These days, ERP and MES systems are focused in obtaining a feasible solution in a short time. To do so, the formulation can be performed using graph theory (Birand et al., 2012), heuristics and metaheuristics (Umetani et al., 2017; Green et al., 2016), timed automata (Behrmann et al., 2005), constraint programming (Rossi et al., 2006), *et cetera*.

However, the progress that mathematical solvers for mixed-integer linear programming (MILP) problems have achieved, has encouraged the use of mathematical programming to solve scheduling problems. Mathematical programming techniques allow the user to include in a rigorous way cost functions to weight the possible solutions and all constraints affecting the problem.

This thesis aims to provide efficient solutions to problems associated to the implementation of real-time optimization (RTO) and scheduling in the process industry.

To focus the problems, two industrial case studies are considered:

The first one deals with the operation of a large evaporation section in a viscose-fiber factory located in Austria, in which the process is basically continuous, but the optimal economic operation requires integration with discontinuous maintenance tasks.

The second case study deals with a tuna canning factory located in Galicia, where continuous product flows and sterilizers operating in batch mode interact and real-time scheduling must be implemented in order to optimize production flows.

Both case studies bring problems that have not received extensive treatment in the literature.

1.1 State of the art

We could follow the roots of scheduling theory to the years previous to the second world war. In the 1950s this field gained recognition, and first important textbooks date from the 1960s (Baker and Trietsch, 2009). A collection of relevant papers can be found in Muth and Thompson (1963).

Nevertheless, it is worth mentioning George B. Dantzig, whom can be seen as the father of linear programming (George Bernard Dantzig, 1983). In the early stages of his academic career, Dantzig formulated the simplex method to solve linear problems (George B Dantzig, 1951). Instead of linear programming, the problem structure was called initially *Programming in a Linear Structure*.

This new solving algorithm and structure was widely applied in the production, planning and allocation field, i.e. the scheduling; since, as he said: "most practical planning problems could be reformulated as a system of linear inequalities" (Albers and Reid, 1986). After the simplex method, a second important achievement by Dantzig was to replace "ground rules for selecting good plans by general objective functions".

Therefore, in the beginning, scheduling was widely attached to mathematical programming, and has been widely study in the operational research field (Winston and Goldberg, 2004).

Typical scheduling problems consider a set of equipment, where a set of different tasks can be performed. Which equipment to use, how to assign tasks to equipment, in which order and when are the most common decisions to be taken. The solution provides answers to these questions minimizing a certain cost functions, and respecting a set of constraints.

In the process industry, the scheduling problems can be organized taking into account different features (Méndez et al., 2006):

- Considering the topology of the processing section, the problems can be divided in: scheduling problem with tasks that only require one stage, scheduling problems with tasks that require several stages, and scheduling problems that require several stages but with different routes between the processes depending on the product.
- Considering the assignment of units, this can be fixed or variable, depending on the existence of multipurpose units.
- Considering the possible routes between units, the section can present a fixed architecture where some units only led to other units, or variable sequences where the job can execute the the different required processes in a free order.

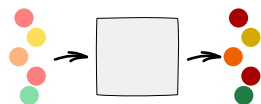
In summary, taking all this possibilities into account, the scheduling problems can be divided into:

- single machine problems, all task have to be performed on the same unit, the scheduling problem has to organized the sequence
- problems with machines in parallel, every task can be performed in one of many parallel units. The schedule algorithm has to assign the tasks to the units and then proceed with the sequence
- flow-shop problems, where there are several stages and the tasks have to be processed on all of them, following a predefined order
- job-shop problems, similar to the previous approach, but the order is related to the task and can differ one from another, also the routes do not have to incorporate all the units
- open-shop problems, these problems consist on job-shop problems but the routes do not have a fixed sequence of units

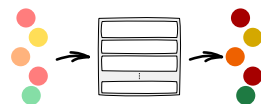
The shop problems can be increased by considering parallel machines, which arises flexible shop problems. In figure 1.2 the different structures are shown.

In addition, when formulating the scheduling problem, it has to be taken into account the following points:

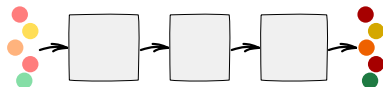
- required storage policies. That can be:
 - unlimited storage, where both the final and the intermediate storage are suppose infinite, and therefore is not included in the model as a constraint
 - limited storage related to tanks or warehouses, one processing task cannot be started if the respective tank is full. The storage can be shared between products, or devoted to a specific product
 - limited storage in the units, again one processing task cannot be started is the unit has not been emptied
 - without storage, where the product after being processed have to be emptied of the unit immediately and sent to another unit, tank or warehouse
- product transfer policies, i.e., the duration of the transfer of products between units and/or tanks can be:



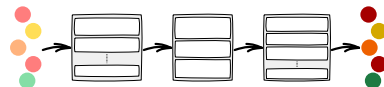
(a) Single machine. All the products have to be submitted to one process.



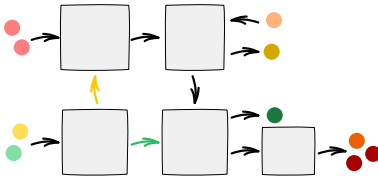
(b) Machines in parallel. All the products have to be submitted to one process, which can be performed by several parallel machines.



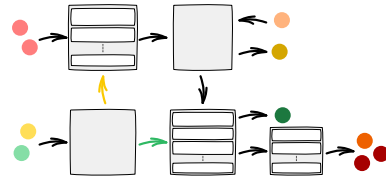
(c) Flow shop architecture. The products have to be submitted to a sequence of processes.



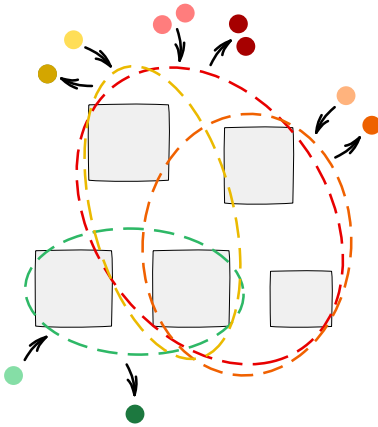
(d) Flexible flow shop. The products have to be submitted to a sequence of processes. Some of the processes can be performed by parallel machines.



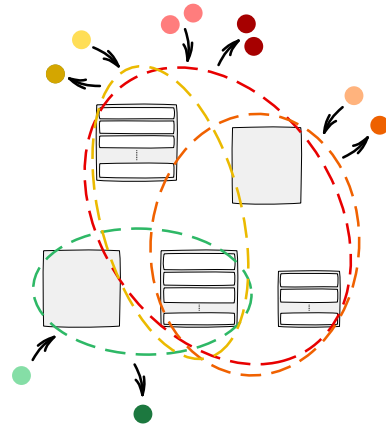
(e) Job shop structure. The different products have to be submitted to a sequence of processes depending on the product.



(f) Flexible job shop. The different products have to be submitted to a sequence of processes depending on the product. Some of the processes can be performed by parallel machines.



(g) Open shop. The products have to be submitted to a series of processes but without a required sequence.



(h) Flexible open shop. The products have to be submitted to a series of processes but without a required sequence. Some of the processes can be performed by parallel machines.

Figure 1.2: Main scheduling problems structures

- approximated as immediate, this approximation can be done when the time require to perform the different tasks takes much more than the time to move the products
- modeled in the scheduling problem
- size of the batch:

-
- the size of the batches is fixed and known *a priori*
 - it is modeled in the scheduling problem, the optimizer decides the size of every task
 - duration of the batch process:
 - the duration can be fixed and related to the type of task
 - the duration can depend on the size of the batch
 - the duration can depend on the unit that performs the task
 - deadline policies. The production orders can have:
 - a fixed due date per order or product
 - a time window to deliver the products
 - changeovers required by the system. The set-up times can be:
 - nonexistent, if the units can work with the different products equally
 - predefined by the unit
 - defined by the tasks sequence
 - defined by the tasks sequence but related to each unit
 - resources constraints. Which depends on the type of resources:
 - discrete resources, as personal or special equipment. The demand can be fixed or variable over time
 - continuous resources, as steam or electrical power. The demand can be variable over time, or fixed through the whole tasks
 - working day constraints. This constraints can consider:
 - non-working days during the weekend
 - already scheduled maintenance tasks
 - the different shifts

- that all the days are equal
- costs associated with different items. When the scheduling problem is formulated with an economical objective function, there can be costs related to:
 - the use of processing units
 - the amount of resource consumed
 - storage use
 - changeovers
- uncertainties present in the system. With respect to the uncertainty in the data, the scheduling model can be formulated as a:
 - deterministic model, where it is assumed that the most probable scenario will occur
 - stochastic model, where all the possible scenarios are considered

The combination of all this factors and the specificity of every process plant, create an enormous variety of problems, so that no single solution pattern can be applied to all of them, but they may share the fact that it is possible to formulate the problems mathematically in terms of mixed-integer programming.

1.1.1 Mixed-integer linear programming

When the scheduling problem is modeled using mathematical programming, the previous features are included using binary and/or real variables using different formulations according to the type of problem considered.

Depending on the general structure of the problem, there are two main formulations: based on precedence or based on networks. In the case of precedence, binary variables assign tasks to units or to slots, which are related to the units and other binary variables define the order between slots.

Regarding network topologies, there are two main formulations, state-task network (STN) described in Kondili et al. (1993) and resource-task network (RTN) explained in Pantelides (1994). Both maintain a continuous control on the amount of available resources, the main difference is how each one model the units.

If the scheduling problem has a particular sequence for the tasks as in flow-shop problems, the precedence formulations are preferred; meanwhile, the network formulations should be use for general scheduling problems.

Both approaches can be modeled using either discrete time grids or continuous slots. With the discrete time grids the time is divided in a sequence of time instants or samples. Events only take place on them. Meanwhile, the continuous formulation depends on the inclusion of variables to represent time events and may be more natural, but the synchronization of events and the management of shared resources is more difficult and sometimes the numerical solution is more difficult.

Whenever possible, due to the numerical difficulties associated to the solution of non-linear problems (MINLP), the formulations of practical scheduling problems tend to be linear:

$$\begin{aligned}
 & \min c^T \cdot \mathbf{x} + d^T \cdot \mathbf{y} \\
 & \text{s.t. } A \cdot \mathbf{x} + B \cdot \mathbf{y} \leq b \\
 & \quad \mathbf{x} \geq 0, \mathbf{x} \in \mathbf{X} \subseteq \mathbb{R}^n \\
 & \quad \mathbf{y} \in \{0, 1\}^q
 \end{aligned} \tag{1.1}$$

where \mathbf{x} and \mathbf{y} are vectors of continuous and binary variables respectively, c , d and b are vector of parameters, and A and B are matrices.

There are several approaches to deal with MILP problems (Floudas, 1995). The most common are:

- cutting planes methods
- branch-and-bound methods

- decomposition methods
- logic-based methods

The cutting planes methods are based on a method to solve integer linear programming problems (Gomory, 1960), using the simplex method. These methods are based on the relaxation of the binary variables. The problem is first solved, if the solution has binary values in all the binary variables, the solution has been found. If some binary variable have real values, new inequalities are added that constraint the feasible region without cutting the integer solutions. Then the optimization problem is solved again, and the process is repeated until the solution gives binary values to all the binary variables.

The branch-and-bound methodology (Dakin, 1965) explore the possible solutions. The solutions can be represented as a binary tree of all the possible combinations of the binary variables using relaxations of binary variables and fathoming to stop exploration of branches. The branch divide the feasible solution region into feasible sub-regions. Each division is compare independently. The bound refers to find lower bounds for the divisions. If a lower bound for one division is larger than a previous solution in a different division, the first one can be disregarded.

Both techniques can be merged in the branch-and-cut method, where cutting planes are used to obtain the lower bounds for every division.

The decomposition methods are based in the *Lagrangean relaxation method* (George B. Dantzig and Wolfe, 1961; Held and Karp, 1970), where the dual problem is exploited.

Last, the logic-based methods (Balas, 1975) uses disjunctive constraints and inference techniques in terms of binary variables.

1.2 Open fields

Even though scheduling is becoming widely implemented in the process industry, there are yet many open problems in the industrial sector (Castro et al., 2018). Scheduling problems arise in very different sections, most of the solutions developed are solutions *ad hoc*.

Actually, in a real industrial environment, scheduling problems are usually solved manually by expert operators using data sheets or similar tools. To obtain formulations as general as possible is a must. Current formulation tend to scale badly. When there is a huge number of tasks to organize, the problem becomes unapproachable using regular formulations. Efficient formulations have to reduce the scaling factors to apply scheduling over a whole processing section and with a long enough prediction horizon.

On the other hand, when the prediction horizon exceeds a few days, the confidence on the forecast shall be reduced. The uncertainty in the external factors or the model is a field that is being study in the advance control field (Zhou and Doyle, 1998), however there is lack in the scheduling applied to the process industry (Baldea and Harjunkski, 2014). With the new market environment, where the demands are higher in number but smaller in size and allowed deadlines, tackling uncertainty in the scheduling is gaining attention, and will become a priority (Grossmann, Apap, et al., 2016a).

When parallel batch units are concurrently consuming from one shared resource, the demand has to be well modeled to prevent the system from blocking. This problem also arise when batch units produce the same final product. The level of the storage has to be taken into account, to prevent overflows in the system.

In addition, the schedule of batch processes where the duration depends on the amount of shared resource or on the schedule itself, is an open problem that is widely present in the industry.

1.3 Aims of the thesis

In this thesis we tried to contribute to the solution of these problems studying specific industrial cases where they are present and providing novel, efficient and industrially feasible solutions to them.

In particular, we aimed at:

- To study the problems of incorporating maintenance due to degradation in continuous operating plants in RTO
- To study the problems of optimally scheduling the operation of a set of continuous plants, incorporating maintenance operations
- To find industrially implementable solutions to these problems incorporating an explicit consideration of the effect of uncertain variables
- To study the problems of operation of plants with continuous lines followed by parallel batch units and the use of real-time scheduling
- To find efficient scheduling formulations for dealing with limited shared resources
- To develop scheduling formulations to tackle variable batch times created by the synchronization itself
- To integrate the operation of the scheduling and control of batch units to improve the overall performance of the system
- To validate the methods and algorithms with industrial data and study its industrial implementation.

1.4 Outline

The rest of the thesis is organized as follows: first, the state of the art in the scheduling area is presented. The most common methodologies to deal with scheduling

problems are organized, and a brief summary on mathematical programming techniques are described.

In chapter three, an scheduling problem about maintenance tasks in continuous lines is presented. The solution is implemented in an evaporation network. First, the optimal operation point for each evaporation plants is computed. Then, maintenance policy for one evaporation is modeled. Last, the whole network is treated, and the scheduling for the maintenance tasks and the distribution of load among the different units is performed. Results on the case study are presented.

In chapter four, the scheduling of a high amount of tasks in multipurpose units is treated. The tasks have to be gathered in slots, and then assigned to the units. The duration of the slots depends on the type of tasks gathered. To prove the performance of the solutions, the formulations are implemented for a canning plant. The cans are produced in a continuous way, but have to be submitted to a batch sterilization process. A new precedence methodology is developed to increase the solution time. The result of using this methodology are shown.

In chapter five, the previous gathering problem is increased by shared resources constraints. The slots have to be synchronized to avoid an excessive consumption. This problem is also presented using the previous industrial case study. The sterilization procedures share steam as heat source. The steam consumption have to be bounded to the maximum capacity of the plant steam supply. Two new approaches to the system are implemented. The solution are implemented and some test are performed and showed.

At last, the main contributions of the thesis are listed, and a brief discussion is presented.

Chapter 2

Maintenance scheduling of continuous processing lines

In the process industry, processing lines tend to work in steady-state, while managers and operators aim to find the best operating point that satisfies the process constraints.

Following the pyramid of control scheme (see figure 1.1 (left)), the optimal operation point can be computed using RTO formulations considering the production targets provided by the economic planning.

The set points obtained are then passed to the control layer, where advanced techniques, as model predicted control (MPC), can be applied to guide the system to the required operation point. This approach is well known and is well described in Camacho and Alba (2013).

Nevertheless, optimal operation points for separate units, may not be the optimal operation point for the global industrial site (Zhang and Grossmann, 2016).

Therefore, integration and coordination among the different elements of a layer and between the different layers of the control pyramid is required. In particular, scheduling may be used to determine the execution order of different activities and its assignment to different units.

In addition, equipment suffers degradation due to its continuous use. The initial optimal operation point may not be optimal anymore when the features of the units change.

Furthermore, the optimal operation point for steady state could increase the degradation effect if this is not considered. The benefits are reduced, and maintenance tasks to recover the lost efficiency become necessary. Moreover, the cost for this maintenance tasks also have to be taken into account in the global operation cost.

This chapter will focus on the development of the optimal control for one large-scale continuous industrial system, and the integration of this control in the optimal scheduling of production and maintenance of the overall network.

First the optimal steady state operation point for one unit will be calculated, and then compared with the optimal policy when considering degradation.

Later, the scheduling of a whole section for several units working concurrently will be obtained. To increase the potential of the solution, a uncertainty study is carried on. A stochastic solution is applied to the schedule.

Next, a real industrial plant section will be described. This industrial site will be use as a reference in the formulation of the problems and as a test site of the methods developed. Then, the main operation problems of the system will be explained. Later, the main objectives of the thesis in this particular case will be listed. In Section 2.3 the methodology applied will be described. After that, the mathematical models obtained to solve the problem and the algorithms and methods developed are described. In the last section, some results are presented.

2.1 Viscose plant

The industrial system considered consists of an evaporation network, owned by Lenzing AG GmbH, the biggest European man-made cellulose fiber plant sited in the small town of Lenzing (Austria). This plant produces several type of either textile fibers or non-woven ones, using wood as raw material. First, the pulp of the

wood that has been matured, is dissolved in a multi-stage physical-chemical process to obtain a viscous fluid. After that, this fluid is pressed through micro-fine nozzles sunken in an acid bath obtaining the fibers by a chemical reaction and then spinning them. The fibers are subject to different treatments, washed and stretched; finally, once they are dried and cut, and in order to obtain a nice and soft final product, the fibers are equipped with a finish made of soap-like saponaceous substances.

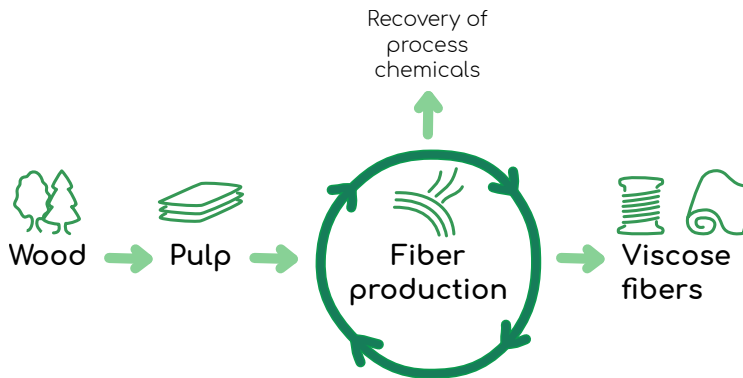


Figure 2.1: Diagram of viscose production (Lenzing AG, 2018)

The chemical processes performed need several chemical compounds, as hydrogen sulfide (H_2S), sulfuric acid (H_2SO_4) and carbon disulfide (CS_2) among others. In order to reduce the wastewater and the carbon footprint this compounds are recovered with different procedures, see figure 2.1. Particularly for the H_2SO_4 , which can be found in the bath used to spin the fibers, or spin-bath in short, it is recovered increasing the concentration of the worn-out spin-bath by evaporating the solvent, in this case water. The plants where this procedure is carried out are multiple-effect evaporation stations, or evaporation plants, (C. de Prada et al., 1987). A typical evaporation plant consists of several separation chambers, each one working at low pressure to ease the boiling, connected in series, as can be seen in figure 2.2. The last evaporation chamber is connected to a barometric steam condenser with a cooling tower. The solution is preheated in heat exchangers arranged in series using the vapor of the evaporation chambers but the last ones, that are heated using

fresh stream. The outcome of the last group of evaporation chamber is mostly recirculated to the heat exchangers. The spin-bath is fed in the station in the last evaporation chambers, meanwhile the concentrate is obtained by overflow in a previous one, see figure 2.2 for a simplified scheme of one evaporation plant.

There are 29 evaporation plants in the evaporation network working concurrently. They are connected to the recovery system of the spinning section, as shown in figure 2.3. These evaporation plants are not equivalent. Each evaporation plant has different evaporation bounds and efficiency.

Even though the evaporation network is a small department in only one step of the fibers production, it means about four fifths of the whole factory steam consumption. Consequently, even a minor reduction on the steam demands of each evaporation station will have a big impact on the overall expenditure. Therefore, the aim of this work is to developed a system able to fulfill the steam requirements in the evaporation network in an optimal way, considering the maintenance operations associated to continuous production

Lenzing AG GmbH produces several types of man-made fibers. Each type uses a particular spin-bath to obtain different final features. Every spin-bath has different requirements regarding the amount of water evaporated. However, the behavior of the evaporation plants does not differ with respect to the spin-bath treated.

Each evaporation plant can only be fed by one type of spin-bath at a time. The spin-baths are divided among the available evaporation plants to obtain the required concentration for each. Due to structural restrictions, there are combinations spin-bath to evaporation plant that cannot be accomplished. See figure 2.4 for a graphical explanation, where different spin-baths are assigned to several evaporation plants, some are forbidden because of possible maintenance scheduled tasks or failures, and each spin-bath has a different production load required.

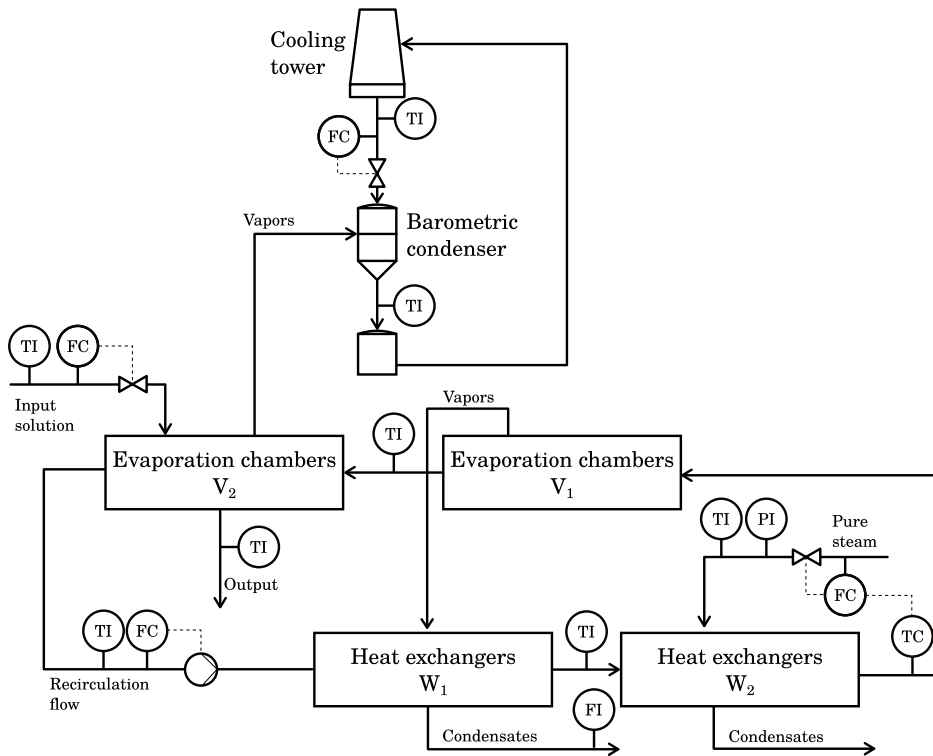


Figure 2.2: Diagram of an evaporating plant, where several evaporation plants and heat exchangers have been grouped in V_1 , V_2 and W_1 , W_2 for simplicity

2.1.1 Operation problems

The evaporation network is the main steam consumer of the site. The amount of steam required per evaporation plant has to be minimized, while the concentration degree for each spin-bath is a target that must be reached and maintained.

On the other hand, there is a shortage of sensors in the system, and hence measurements. Therefore, there is a lack of awareness on the evolution of the system and the actual operation state. Besides, the measurements acquired could be affected by noise or caused by a faulty sensors, requiring a previous treatment to assure the information recovered is trustworthy.

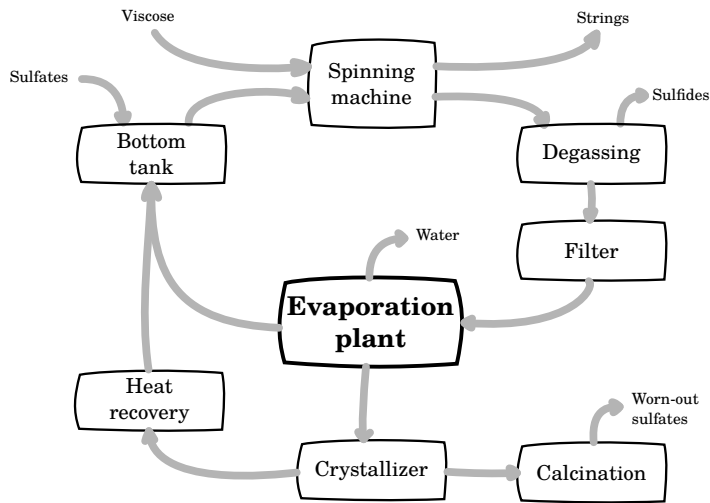


Figure 2.3: Recovery station connections

Industries which have fluids flowing through pipes usually have to deal with fouling problems inside them that can produce a loss of efficiency. In the case of heat exchangers, this deposit formation is related mainly to the fluid velocity, the compounds of the dissolution and the temperature. The fouling layer affects the heat-transmission coefficient and can block the conduct. Performing regular cleaning tasks becomes mandatory.

There are several types of cleanings, each with different recovery degree, required cleaning time and material and personnel cost.

The evaporation plants have different efficiency. The spin-bath fluids have to be distributed properly among the evaporation plants. Meanwhile, the aggregation of water evaporated per spin-bath is constrained.

All the production processes in the plant are continuous, which means the spin-bath fluid has to be concentrated also continuously. When one evaporation plant have to be stopped to be cleaned, the rest of the evaporation plants connected to the same spin-bath, or the ones stopped and already cleaned, have to take care of

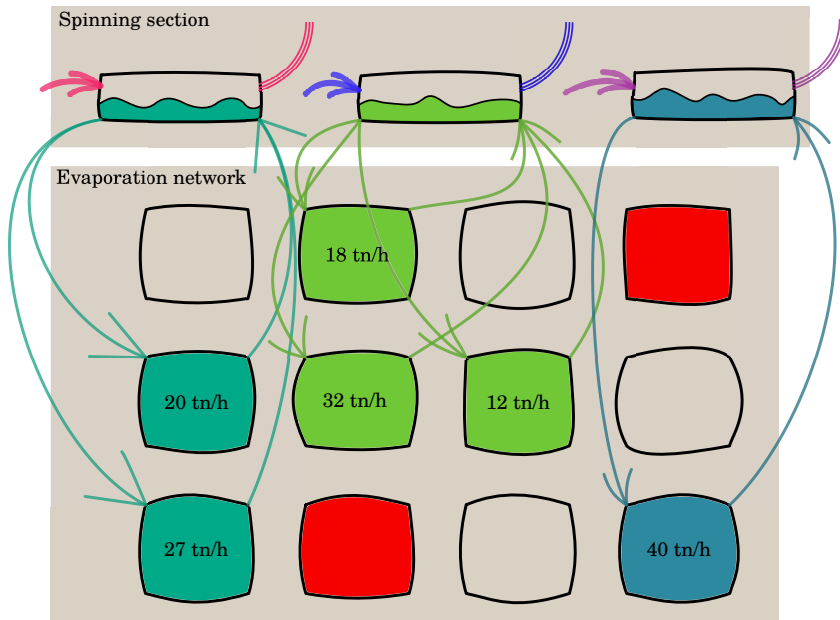


Figure 2.4: Example of connections between spinning units and evaporation plants and the distribution of production load among them

the load assigned to this one. Nevertheless, they can switch between some spin-baths for each evaporation plant, provided that a cleaning task is done in between.

Yet, only one cleaning task can take place at the same time due to limitations on the available personnel.

2.2 Objectives

The global aim is to improve the efficiency of the evaporation network in the industrial site explained. This implies targets at the plant level and targets at the network level. Initially, the optimal operation point for every evaporation plant will be computed. The cleaning tasks frequency will be calculated.

Then, we will analyze how to optimally distribute the spin-bath loads among the different evaporation plants. And then, the fouling effect will be added to the

global problem, computing the scheduling of the use of the evaporation plants and its maintenance.

The evaporation plants will be modeled to simulate the behavior of the system. A reliable model will be obtained to be able to estimate the status of the plant.

The optimal operation point that minimizes the steam consumption will be computed. Then, the fouling effect has to be added, and the optimal cleaning policies will be obtained.

The model to distribute the spin-bath among the evaporation plants will be implemented. After the optimal distribution, the fouling effect in all the evaporation stations will be included, and so will the cleaning policies for all the evaporation network.

In summary:

- To obtain a reliable model to simulate the behavior of the system
- To calculate the optimal operation point for one evaporation plant with and without fouling dynamics
- To compute the optimal sort of the cleaning tasks and when to perform it in order to maximize the benefits
- To conduct the optimal load allocation among several evaporation plants
- To schedule the cleaning tasks for the whole evaporation network

2.3 Methodology

First, one evaporation plant has to be studied alone. As production changes are rare, and the dynamic of the system is fast enough, the model focuses on representing the steady state reliably. Its kernel is based on first principles, i.e., energy and mass

balances (including heat transmissions), density relationship between volumetric and mass flows, and chemical equilibrium within the evaporation chambers.

Due to a lack of measurements and controllers, the model of the real station had to be somehow simplified in the number of devices present, as it can be seen in figure 2.2, where heat exchangers and evaporation chambers have been gathered in two sets each. Nevertheless, this increases the uncertainties in the model and forces to apply different techniques to reduce them and have a valid model.

Real behavior of the processes is difficult to formulate, moreover with the simplifications done. To overcome this problem, some parts are modeled as black-box where relations between variables are identified from real data including empirical constraints, giving rise to a gray-box model. Grey-box models have been proved as a reliable approach to simulate the state of a system (Pitarch, Sala, et al., 2019).

The approach starts by recognizing that not all plant measurements are reliable, and good estimates of them, besides unknown parameters, must be obtained before any optimization can take place. The different techniques to approximate the models through the data from the real plant, depend on the existing redundancy in the initial model due to additional algebraic constraints and duplicated sensors.

In this methodology to obtain the gray-box models, instead of including relations between the not clearly determined variables in the model, their values are constrained within a sensible region. Then, the model is approximated by applying data reconciliation methods (Leibman et al., 1992). This process estimates the plant status by solving an optimization problem that finds the values of the model variables and unknown parameters that better fit the plant measurements:

$$\begin{aligned} \min_{\theta} \eta^2 \cdot \sum_J \in J \left[\frac{|\varepsilon_j|}{\eta} - \log \left(1 + \frac{|\varepsilon_j|}{\eta} \right) \right] \\ [\theta] \leq \theta \leq [\theta] \\ h(\theta) = 0 \end{aligned} \quad (2.1)$$

where θ is a set containing the estimated value of the measures, the value of internal parameters and the value of the status; and the error ε_j is computed as

$(\theta_j - y_j) / \sigma_j \forall j \in J$, where the set J is a subset of the θ elements that corresponds to the measurements $(y_{j \in J})$, and σ represents the variance. The vector $h(\theta)$ is a vector of non-linear functions of class C^2 , i.e., the model constraints.

The objective function in this case is the so-called *Fair function*, however data reconciliation can be performed with any usual error measurement technique, v.gr., least squares. Fair function is a robust estimator (Huber, 2014), which prevents weighting gross errors. The slope of the function to reduce the weight of large residuals is tuned using the parameter η .

After solving (2.1), a set of reliable values are obtained for all variables, according to the model $h(\theta)$, which in our case is the simplified first-principles model.

Using the new values, unknown patterns or relations between variables can be identified in order to complete the gray-box model. These relations are then modeled using regression techniques, and the new equations obtained are incorporated to the model.

Additionally, the new model can be again adjusted, for example applying again data reconciliation. Therefore, the methodology to obtain precise gray-model models consists in three main steps: first, developing a first-principle model; then, adjusting the system and uncertain states through data reconciliation; and finally, obtaining using regression techniques over the new data the relation between the unknown variables. The process is well explained in Pitarch, Palacín, Merino, et al. (2017).

To enhance the knowledge about the process, some indicators that represent the relation between the production and the resources required in real time are implemented, the so called Resource Efficiency Indicators (REI) (Krämer and Engell, 2018). These REIs give information to the operators about the favorable current operation state. The developed REIs that relate the steam consumption (resource) to the water evaporated (efficiency) are used as cost function in RTO problems.

Once a reliable process model is available, next step is to formulate the optimal plant operation in terms of mathematical programming.

Current optimization algorithms can perform efficiently if accurate enough gradients for the cost function and constraints are provided and, in some cases, the Hessian is also needed (Lorenz T. Biegler, 2018), v.gr., Ipopt (Wächter and Lorenz T Biegler, 2006).

On the other hand, automatic differentiation technique has been proved more powerful than symbolic differentiation in terms of speed, and than numerical differentiation regarding accuracy, (Griewank, 1988). Hence, the model is coded using CppAD that performs automatic differentiation, (Bell and Burke, 2008). The gradients can be computed as the derivatives of the cost function and constraints with respect to the decision variables.

In this way, an optimization problem can be formulated with the REIs set as the cost function, i.e., minimizing the resource use preserving the load demand. This solution will give optimal set points to operate the plant within the actual state. Using automatic differentiation and current NLP solving algorithms, the solution can be obtained in real-time.

Subsequently, the dynamic of the fouling growth is modeled. The steady-state model previously developed has to include the slow dynamics of the fouling growth inside the pipes. This is implemented by estimating the heat transfer coefficient over a large period of time by means of several independent steady-state models. In this way, we can obtain the evolution of the heat transmission coefficient, so that a model relating this heat transfer coefficient with other process variables can be developed. The model can then be simulated from the present state until a cleaning task takes place.

The RTO problem is also modified, replacing the cost function based in REIs with an economic target. This optimization problem takes into account also the cost of the operation in the future with its decreasing efficiency and including the cost of the cleaning task. In this dynamic scenario the control actions don't have to remain constant, which can be achieved splitting the prediction horizon in different

intervals. This economic RTO gives a prediction about the behavior of the plant with the expected control actions, and estimate the best cleaning type and the moment to apply it in order to maximize the economic benefit.

Once the optimization problem for one evaporation plant has been solved, the goal now becomes to compute the optimal policy for the whole evaporation network comprising 29 evaporation plants. This includes the load allocation for all the evaporation plants, meaning the flow assigned to each evaporation plant and the type of spin-bath while maintaining the total flow of each of them, and the schedule of cleaning tasks. Including discrete decisions, the problem becomes mixed-integer, which increases the computational weight of the optimization problem. To reduce the resolution times in order to being able to keep real-time implementation, the model is implemented as a MILP problem.

The non-linearity of the model are linearized. A surrogate model is estimated using the previous rigorous model. This new model relates the operation cost with the load of the evaporation plant, the fouling state and other external factors that can't be modified but has to be taken into account, as outdoor temperature.

Then, the evolution of the fouling state is modeled in the scheduling problem to obtain the optimal cleaning policy.

Once the deterministic standpoint has been solved, the uncertainty in the external factors mentioned is included as an obstacle that the model has to deal with. To deal with the uncertainties, the stochastic methods have become one important tool, (Heyman and Sobel, 2004; Constantinescu et al., 2011). As the model is linear, a two-stage approach can assure feasibility by creating scenarios to all the possible combinations of the bounds of the variables, (Ruiz-Femenia et al., 2013; Grossmann, Apap, et al., 2016b).

Using stochastic techniques in scheduling has been proved useful in Palacín, Pitarch, Jasch, et al. (2018).

2.4 Evaporation process optimization

The evaporation plant is modeled as follows. First, the heat exchangers are merged into only two devices. One using fresh steam as heat source (W_2), and one using the evaporation chambers vapor (W_1). And, the same process is applied to the evaporation chambers, where also only two devices remain, one which vapors are used to heat the flow (V_1), and a second one (V_2) connected to the barometric condenser, which actually is formed by two parallel condensers, one big (BC_1) and one small (BC_2). The numerical order of the exchangers and chambers has been set due to the spin-bath flow direction.

Inside the heat exchangers, evaporation chambers and barometric condensers, mass and energy balances are arranged. Defining as D the set of these devices, i.e., $D = \{V_1, V_2, W_1, W_2, BC_1, BC_2\}$, and as K the set of chemical compounds present in the spin-bath, v.gr., H_2O :

$$\mathbf{f}_{id}^T \cdot \mathbf{C}_{id} = \mathbf{f}_{od}^T \cdot \mathbf{C}_{od} \quad \forall d \in D \quad (2.2)$$

$$\mathbf{f}_{id}^T \cdot \mathbf{enthw}(T_{id}, \mathbf{p}_{id}) = \mathbf{f}_{od}^T \cdot \mathbf{enthw}(T_{od}, \mathbf{p}_{od}) \quad \forall d \in \{BC_1, BC_2\} \quad (2.3)$$

$$\mathbf{f}_{id}^T \cdot \mathbf{enth}(T_{id}, \mathbf{C}_{id}) = \mathbf{f}_{od}^T \cdot \mathbf{enth}(T_{od}, \mathbf{C}_{od}) + \kappa_d \cdot ([T]_d - T_{amb}) \quad \forall d \in \{V_1, V_2\} \quad (2.4)$$

$$Q_d + \mathbf{f}_{id}^T \cdot \mathbf{enth}(T_{id}, \mathbf{C}_{id}) = \mathbf{f}_{od}^T \cdot \mathbf{enth}(T_{od}, \mathbf{C}_{od}) + \kappa_d \cdot ([T]_d - T_{amb}) \quad \forall d \in \{W_1, W_2\} \quad (2.5)$$

Where \mathbf{f}_{id} and \mathbf{f}_{od} are vectors representing the incoming mass flows to and outgoing mass flows from a device respectively; and \mathbf{C}_{id} and \mathbf{C}_{od} are matrices of $|K|$ columns representing the content in the incoming flows and outgoing flows respectively for each chemical compound $k \in K$. Note that vectors will be defined as column vectors, as long as the opposite is not specified.

The enthalpy of the flows with the temperatures T and the concentrations C is calculated by $enth(t, C)$, using the specific heat of each compound previously computed; and, for water or steam $enthw$ is used, that computes the water specific enthalpy using the flow temperature or pressure. The parameter κ_d is the coefficient of heat loss to the ambient in the device d , $[T]_d$ is the maximum temperature in the equipment d , and T_{amb} is the ambient temperature. For the heat exchangers (W_1, W_2), the flows considered (f_{id}, f_{od}) are the spin-bath flows and, assuming that there are no loss of material, they are equivalent, i.e., $f_{id} \equiv f_{od}, \forall d \in \{W_1, W_2\}$; however, in order to preserve the same notation, the sub indexes i and o for incoming and outgoing flows are maintained. Q_d represents the heat gained from the steam.

$$Q_d = UA_d \cdot \Delta T_{LMd} \quad \forall d \in \{W_1, W_2\} \quad (2.6)$$

$$\Delta T_{LMd} = \frac{\Delta T_{ind} - \Delta T_{outd}}{\ln\left(\frac{\Delta T_{ind}}{\Delta T_{outd}}\right)} \approx \left(\Delta T_{ind} \cdot \Delta T_{outd} \cdot \frac{\Delta T_{ind} - \Delta T_{outd}}{2}\right)^{\frac{1}{3}} \quad \forall d \in \{W_1, W_2\} \quad (2.7)$$

$$Q_d = f_{s_{id}} \cdot lat(T_{s_{id}}) \quad \forall d \in \{W_1, W_2\} \quad (2.8)$$

The variable UA_d is the heat transmission coefficient times the exchange surface for the device d , and ΔT_{LMd} is the logarithmic mean temperature difference, approximated using Chen formulation (Chen, 1987) as it can be seen in equation 2.7, where ΔT_{in} and ΔT_{out} are the temperature differences between the flows at the inlet and at the outlet respectively. $f_{s_{id}}$ is the incoming steam flow into the equipment d , and $lat(T_{s_{id}})$ is used to compute the condensation latent heat of the incoming saturated steam flows at temperature T_s in the device d .

The, phase equilibrium between pressure, concentration and temperature inside the evaporation chambers, i.e., V_1, V_2 , is formulated.

$$\bar{P}_d = pres(\bar{T}_d, \bar{C}_d) \quad \forall d \in \{V_1, V_2\} \quad (2.9)$$

$$\bar{T}_{S_d} = temp(\bar{P}_d) \quad \forall d \in \{V_1, V_2\} \quad (2.10)$$

Where \bar{P}_d is the mean pressure inside an evaporation chamber d , computed using the mean temperature of all the flows (\bar{T}_d) and the mean concentrations for each compound of these flows (\bar{C}_d). With the pressure estimation, the mean temperature of the saturated steam (\bar{T}_{S_d}) can be calculated.

The cooling tower, CT is modeled using previous studies found in the literature, as the ones by Wallace and Hobbs (2006). H_{S_x} and $H_x : x \in \{i, o\}$ are the saturated and current specific humidity respectively of the incoming and outgoing air flows; and H_R represents the ambient relative humidity. The partial pressure of the saturated steam is calculated using $ppres$ that relate the pressure and temperature. The flows f_a and f_{iCT} and f_{oCT} represent the incoming air flow to the cooling tower, and the inlet and outlet water flows respectively, being t_a , t_{iCT} and t_{oCT} their temperatures. The water specific heat (SH) is equalized to $4.184 \text{ kJ kg}^{-1} \text{ K}^{-1}$. See figure 2.5 for clarifications.

$$H_{S_i} = 0.622 \cdot \frac{ppres(T_{amb})}{1.01325 - ppres(T_{amb})} \quad (2.11)$$

$$H_{S_o} = 0.622 \cdot \frac{ppres(T_{ao})}{1.01325 - ppres(T_{ao})} \quad (2.12)$$

$$H_R = 100 \cdot H_i / H_{S_i} \quad (2.13)$$

$$H_o = \begin{cases} H_{S_o} & \text{if } (T_{ao} > T_{amb}) \\ H_{S_i} & \text{if } (T_{ao} \leq T_{amb}) \text{ and } (\bar{H}_o > H_{S_i}) \\ [H_o] & \text{otherwise} \end{cases} \quad (2.14)$$

$$[H_o] = SH \cdot (T_{iCT} - T_{iBC}[\text{water}]) \cdot \frac{f_{oCT}}{f_a \cdot lat(T_{amb})} + H_i \quad (2.15)$$

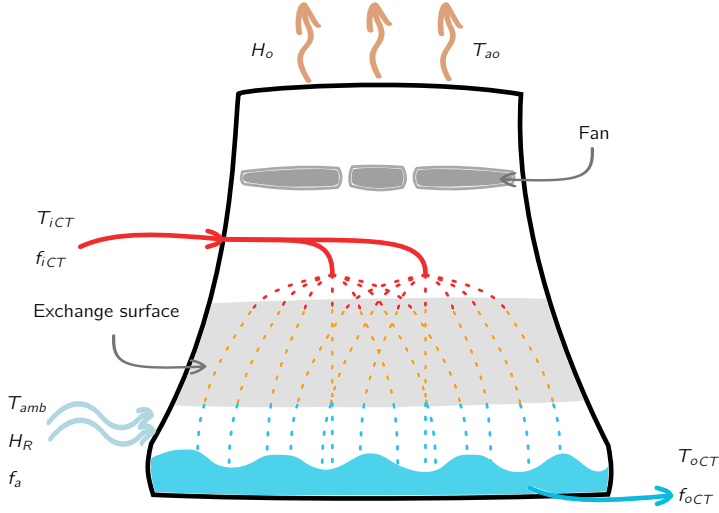


Figure 2.5: Diagram of cooling tower

Finally, flow behavior equations are implemented, where the features of the flows are related. Represented by v_{id} the volumetric incoming flows for each device $d \in D$, and calculated the density of each flow as a function of its temperature, concentration of each component and pressure ($dens(\mathbf{T}_{id}, \mathbf{C}_{id}, \mathbf{p}_{id})$). Note that water and steam flows only have one compound, i.e., $\mathbf{C}_{id}[i']$ is a row of the matrix with the elements equal to zero but for water or steam column respectively; and that they are represent as different compounds due to the different density formulas used for a water flow or a super heated steam flow. The parameters vector \mathbf{MW} contains the molecular weights of the different chemical compounds.

$$\mathbf{f}_{id}[i'] = \mathbf{v}_{id}[i'] \cdot dens(\mathbf{T}_{id}[i'], \mathbf{C}_{id}[i'], \mathbf{p}_{id}[i']) \quad (2.16)$$

$$\forall d \in D, \forall i' \in i$$

$$dens(\mathbf{T}_{id}[i'], \mathbf{C}_{id}[i'], \mathbf{p}_{id}[i']) = \mathbf{X}_{id}[i'] \cdot \mathbf{den_comp}(\mathbf{T}_{id}[i'], \mathbf{p}_{id}[i']) \quad (2.17)$$

$$\forall d \in D, \forall i' \in i$$

$$X_{id}[i', k] = \frac{C_{id}[i', k]}{MW[k]} \quad (2.18)$$

$$\sum_{k' \in K} \frac{C_{id}[i', k']}{MW[k']}$$

$$\forall d \in D, \forall k \in K, \forall i' \in i$$

The flow through the valve that controls the pure steam is calculated using the valve constant C_v as follows, where p_{in} represent the pressure of the incoming steam flow to the steam saturator. Also, the vapors flow that leaves the second evaporation chamber route to the steam condenser is modeled similarly, where κ_f is a parameter that has to be estimated.

$$f_{iW_2}[steam] = a \cdot C_v \cdot \sqrt{p_{in}^2 - p_{od}^2} : d = \text{Steam saturator} \quad (2.19)$$

$$f_{ov_2}[steam] = \kappa_f \cdot \sqrt{pres(T_{ow1}[steam])^2 - pres(T_{iBC}[water])^2} \quad (2.20)$$

This model represents almost completely the evaporation network, however there are still some dependencies and some behaviors that cannot be modeled using first principles. Therefore additional equations and mathematical relations, based on expert knowledge and regression analysis, have to be included. In order to do so, real data from the plant is first filtered, excluding outliers and transients evolution, and then a data reconciliation process is run.

Once the data reconciliation process has been executed, some relations between variables can be identified using the smoother data obtained, and then added to the model. The coefficient κ_f , which determines the steam flow from the evaporation chamber V2 to the condenser in (2.20), can be approximated by an equation of $T_{iBC}[water]$ (T_{oCT} to shorten). And the cooling tower performance, $\eta = (T_{iCT} - T_{iBC}[water]) / (T_{iCT} - T_{ao})$, is related to the temperatures of water and air, see figure 2.6.

$$\kappa_f = \begin{cases} 14.67 & \text{if } (T_{oCT} \leq 12.1) \\ 0.0121 \cdot T_{oCT}^2 - 0.9772 \cdot T_{oCT} + 24.247 & \text{otherwise} \end{cases} \quad (2.21)$$

$$\eta = \begin{cases} 1 & \text{if } (\Delta T < 5) \\ 0.387 & \text{if } (\Delta T \geq 25) \\ 126'807 \cdot \exp -0.0475 \cdot \Delta T & \text{otherwise} \end{cases} \quad (2.22)$$

where $\Delta T = T_{oCT} - T_{amb}$.

In addition, the incoming air flow to the cooling tower, f_a , is determined as the sum of the air flow produced by the fan of the cooling tower (f_{afan}), and the flow produced by the convection effect (f_{aconv}). Data from cold seasons, when the fan is stopped, feeds the data reconciliation to approximate the convection effect, an Eckert-like number is identify ($E_c = 18900$). And data from hot seasons, when the convection effect is negligible, is used to identify the maximum flow speed produced by the fan, $[f_{afan}] = 21 \times 10^5 \text{ m}^3 \text{ h}^{-1}$, that is obtained when the fan runs at maximum speed, $[\omega] = 1500 \text{ rpm}$. Then, the incoming air flow is modeled as:

$$f_a = f_{afan} + f_{aconv} \quad (2.23)$$

$$f_{afan} = [f_{afan}] \cdot \frac{\omega}{[\omega]} \quad (2.24)$$

$$f_{aconv} = E_c \cdot \left(\frac{T_{iCT} + T_{oCT}}{2} - \frac{T_{ao} - T_{amb}}{2} \right) \quad (2.25)$$

Further there is a slow dynamic on the system due to the fouling on the pipes. This dirt layers growth cause a loss in the heat transmission coefficient (UA_d). It

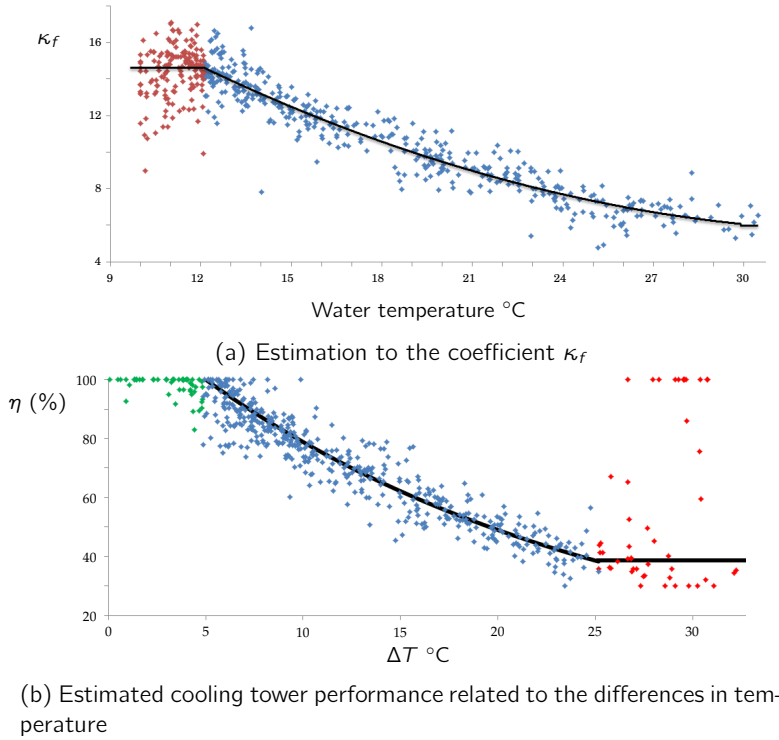


Figure 2.6: Graphs representing the reconciliated data and the regression models

is approximated as:

$$\begin{aligned}
 UA_d(m, \mathbf{T}, \mathbf{v}) = & UA_{d0} - \frac{\alpha \cdot T[d]^4 + \beta \cdot T[d]^3 + \gamma \cdot T[d]^2 + \delta \cdot T[d] + \zeta}{\mathbf{v}[d]} \\
 & \cdot \left(1 - \exp\left(-\tau \cdot \frac{t}{\mathbf{v}[d]^2}\right) \right) \\
 & \forall d \in \{W_1, W_2\}
 \end{aligned} \tag{2.26}$$

where the values for α , β , γ , δ and ζ are obtained using model fitting to the values of U , acquired by data reconciliation.

The accuracy of the solution obtained can be see in the figure 2.7.

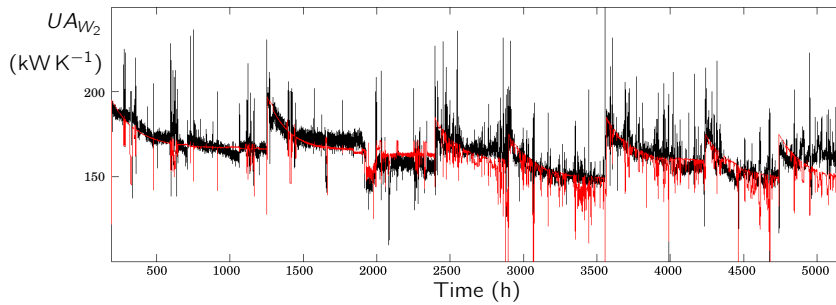


Figure 2.7: Approximation and value obtained from real data comparison for the heat transmission coefficient in one group of heat exchanger chambers

2.4.1 Real-time optimization of one evaporation plant

The optimizer has to solve the NLP problem:

$$\begin{aligned} \min_{f_{iW_1}, T_{oW_2}} \quad & f_{iW_2}[\text{steam}] \\ \text{s.t.} \quad & (2.2) - (2.22) \end{aligned} \quad (2.27)$$

Initially, the steady state system is optimized using CppAD to model the evaporation plant, and Ipopt to solve the RTO problem. The appropriate REI related to the steam consumption is used as cost function. The optimizer gave a solution that consist in maximizing the utilization of the cooling tower, and maximizing the temperature of the income product to the evaporation chambers and then adjust the recirculating flow to assure the required evaporation rate. As the income product to the evaporation chambers is heated as much as possible, their vapors will heat the spin-bath in the heat exchangers, reducing the total steam flow needed.

This solution reduces the amount of steam required without regarding the efficiency decrease over time. It is implemented in the industrial site by a self-optimizing control strategy and has reduced sensibly the steam consumption.

RTO considering maintenance

In this implementation, the dynamic of the fouling growth is included in the problem. Considering two types of cleaning tasks, one cheaper but with a lower recovery degree A , and one more efficient but more expensive B (which can be defined as small and big types), the relative dynamic RTO is defined as:

$$\begin{aligned} \min_{c, t_F, \mathbf{u}_A, \mathbf{u}_B} \quad & c \cdot \frac{\int_{t_0}^{t_F} \text{SteamFlow}(t) \cdot P_{th} \cdot dt + \text{cleaningCost}_A}{t_F - t_0} \\ & + (1 - c) \cdot \frac{\int_{t_0}^{t_F} \text{SteamFlow}(t) \cdot P_{th} \cdot dt + \text{cleaningCost}_B}{t_F - t_0} \quad (2.28) \\ \text{s.t.} \quad & h(\mathbf{u}_A) = 0 \\ & h(\mathbf{u}_B) = 0 \end{aligned}$$

where $c \in [0, 1]$ is a real variable to select between the types of cleaning A and B , t_F is the time when the cleaning task takes place, and the vectors \mathbf{u}_A and \mathbf{u}_B are the decision variables for the case when the system performs a cleaning task of type A or of type B respectively, and P_{th} is the price for fresh steam production.

The model of the evaporation plant including the evolution of the dirt layer, (2.2)-(2.26), is represented as $h(\mathbf{u}) = 0$. Notice that the equation for the time evolution of UA is included, which provides a dynamic nature to the model. The optimizer will obtain the optimal decision variables evolution for both cleaning types, and will decide which one is less expensive in terms of steam consumption and cleaning cost.

Note that (2.28) is monotonous with respect to c . Therefore, the minimum value will be either at $c = 0$ or at $c = 1$.

To implement the solution, the problem (2.28) is discretized as can be seen in (2.29). Then, the main RTO problem consists of as many independent RTO problems as time samples, using as objective function a cost aggregation. The independent models for every time sample will be linked by the slow fouling dynamic modeled in (2.26), and the time samples length that will be optimized by the NLP solver.

$$\begin{aligned}
\min_{c, t_F, \mathbf{u}_A, \mathbf{u}_B} & c \cdot \frac{\sum_{t=t_0}^{t_F} (\text{SteamFlow}[t] \cdot P_{th}) + \text{cleaningCost}_A}{t_F - t_0} \\
& + (1 - c) \cdot \frac{\sum_{t=t_0}^{t_F} (\text{SteamFlow}[t] \cdot P_{th}) + \text{cleaningCost}_B}{t_F - t_0} \quad (2.29) \\
\text{s.t. } & h(\mathbf{u}_A, t) = 0 \quad \forall t \in [t_0, t_F] \\
& h(\mathbf{u}_B, t) = 0 \quad \forall t \in [t_0, t_F]
\end{aligned}$$

The solution for the dynamic RTO including the fouling dynamics and the steady state solution has been compared in figure 2.8. As it can be seen, when the evaporation plant is cleaned, the steady state solution is similar to the one including the dynamics. However, when the dirt layer grows, this solution tries to reduce the growth by increasing the speed of the flow.

Both solutions compared can be seen in figure 2.8, where the final t_F has been forced to 600 h. The optimizer took around one minute to obtain the solution, therefore this model is suitable to be applied in real-time.

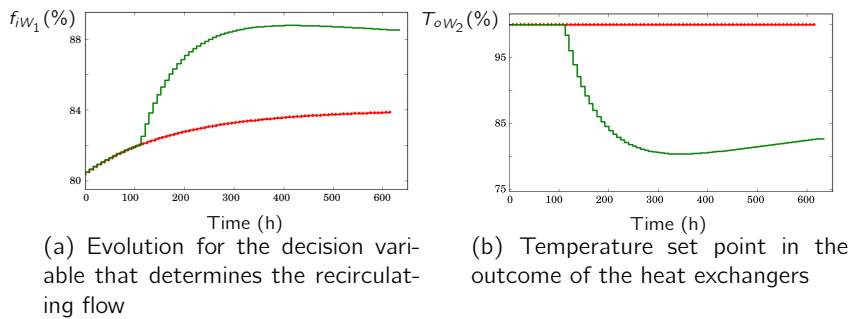


Figure 2.8: Optimal control evolution for an evaporation plant using dynamic RTO (green) compared with a steady state RTO solution (red)

2.5 Evaporation network scheduling model

It must be mentioned that, even though a non-linear approximation of the fouling evolution has been obtained, which relates the dirt layer evolution with the working set points, in practice it has been noted that it mostly depends on the time that the evaporation plant has been running since the last cleaning, and there is no much loss of accuracy by making this approximation.

Therefore, a surrogate model that relates the cost associated to the specific steam consumption to the production load, the ambient temperature and the fouling growth degree can be obtained, see figure 2.9. This model is obtained by performing a linear regression from simulated working points. The points are equally spaced through the three dimensions (production, temperature and fouling), setting up a regular grid. With this surrogate models, the cost function value for one evaporation plant becomes linear with respect to the decision variables and the external factors.

The evolution over time of the evaporation plant efficiency is discretized, using one day as the length of each sample. This duration is chosen because it corresponds to the average time required to perform one cleaning task. The evolution of the state of each evaporation plant is represented by a finite state machine (FSM), where each state is the actual condition of the plant, see figure 2.10, (Basán et al., 2017). The scheduler has to organize the tasks to perform, by deciding the evolution over time of the state of the evaporation plants.

One evaporation plant, if completely clean, starts on the first working state S_0 . As time passes, the evaporation plant evolves through the working states. The evolution through the states let us keep track of the fouling evolution as well. As we can see in figure 2.10, after a certain number of days in operation, it is possible to choose between keep working or perform a cleaning task of one specific type. The cleaning type available depends on the fouling growth. However, as it may not be any cleaning team accessible, there is one state that represent one plant stopped but waiting to be cleaned. As the cleaning type depends on the working state the flow comes from, there is one dirty stand-by state per cleaning type. After

one cleaning task is performed, the next state can be again a working state, or the evaporation plant can wait stopped until needed. The working state where the plant resumes depends again on the type of cleaning, as a less expensive clean cannot fully remove the dirt layer.

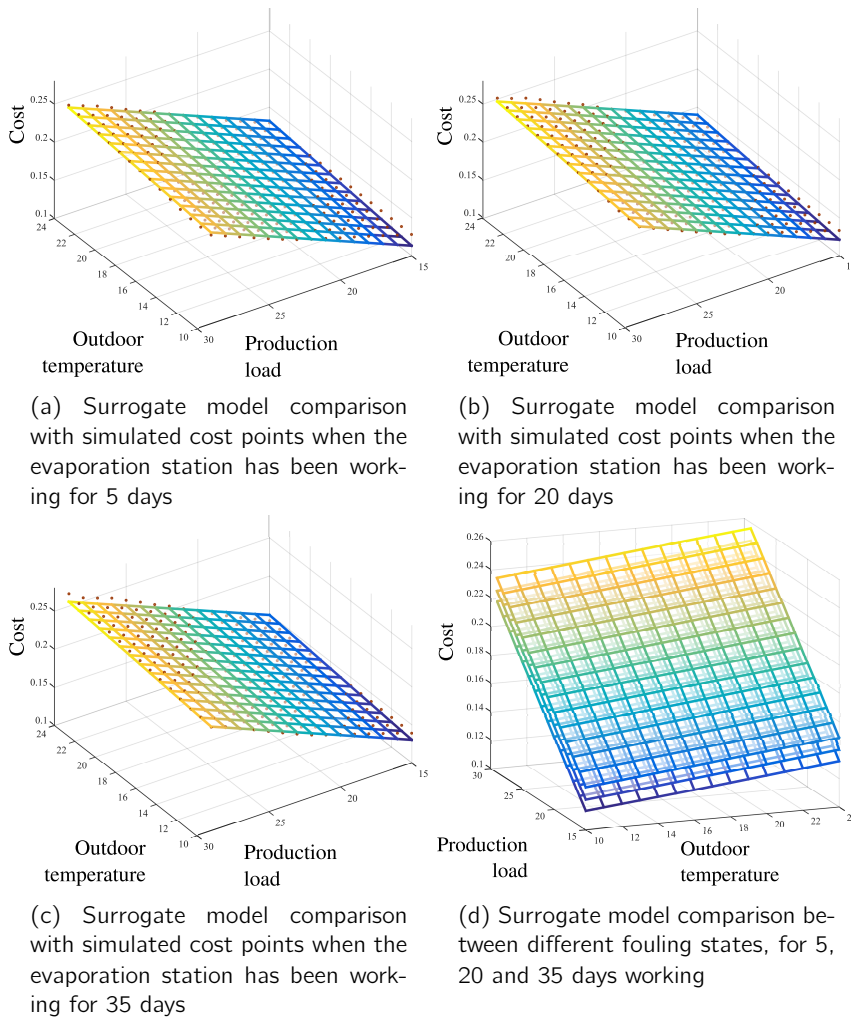


Figure 2.9: Surrogate model representation for the steam consumption cost in a single plant in three different fouling states

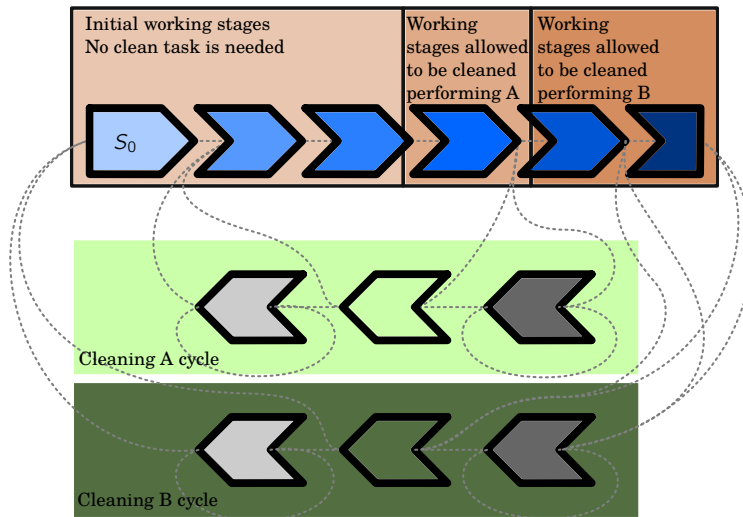


Figure 2.10: Finite state machine to represent the evolution of the current status of one evaporation plant. The states are divided in three cycles, the working cycle with a brown background and two types of cleaning cycles with green background, the darker the green the deeper the clean effect. The working cycle consist in several sequential working states (blue chevrons) that model the fouling evolution. The cleaning cycle has three different states, stand-by states before and after cleaning task (gray chevrons) that are optional, and the proper cleaning task (green chevrons) that is mandatory. The dashed gray lines represent the possible routes between the states

Then, a global scheduling problem arises, where how distributing the different spin-baths flows among the evaporation plants and programming the cleanings minimizing the overall consumption is the new objective function. The optimizer must keep track of the evolution of all evaporation plant status and decide the optimal load allocation in every instant.

The mixed integer linear programming problem approach applied establishes four sets: the evaporation plants, \mathcal{V} ; the states of the FSM, \mathcal{S} ; the different spin-baths or products, \mathcal{P} ; and the time samplings or days, \mathcal{M} with $m_F \in \mathcal{M}$ as the last day. The set of stages, \mathcal{S} , is divided in subsets: $\mathcal{S}_I \subset \mathcal{S}$, are the initial stages where

the decision of cleaning is not yet profitable; $\mathcal{S}_X \subset \mathcal{S} : X \in \{A, B, \dots\}$, are the stages where the decision of making a cleaning task has to be taken into account, representing by X the type of cleaning (type A , type B , *et cetera*); $\mathcal{S}_L \subset \mathcal{S}$, are the cleaning stages *per se*; and $\mathcal{S}_P \subset \mathcal{S}$, are the stages where the evaporation plant is stopped, this set, in turn, is divided into pre-cleaning stops, $\mathcal{S}_{PX} \subset \mathcal{S}_P : X \in \{A, B, \dots\}$, and post-cleaning stops, $\mathcal{S}_{PLX} \subset \mathcal{S}_P : X \in \{A, B, \dots\}$. All these subsets don't intersect, v.gr., $\mathcal{S}_I \cap \mathcal{S}_L = \emptyset$, $\mathcal{S}_A \cap \mathcal{S}_B = \emptyset$, and so on.

Two binaries variables are defined to relate the sets: $E_{vsm} : v \in \mathcal{V}, s \in \mathcal{S}, m \in \mathcal{M}$, denotes that the evaporation plant v is in the state s in the day m ; and $A_{vpm} : v \in \mathcal{V}, p \in \mathcal{P}, m \in \mathcal{M}$, which stands for the assignment of product p to the evaporation plant v in the day m .

First, some assignment equations are formulated, in order to represent nicely the network. Each evaporation plant can only be in one state at a time. Each plant can only be working with one product, and this is only possible when it is working. And only one cleaning task can be performed per day.

$$\sum_{s \in \mathcal{S}} E_{vsm} = 1 \quad \forall v \in \mathcal{V}, \forall m \in \mathcal{M} \quad (2.30)$$

$$\sum_{p \in \mathcal{P}} A_{vpm} + \sum_{s \in (\mathcal{S}_L \cup \mathcal{S}_P)} E_{vsm} = 1 \quad \forall v \in \mathcal{V}, \forall m \in \mathcal{M} \quad (2.31)$$

$$\sum_{s \in \mathcal{S}_L, v \in \mathcal{V}} E_{vsm} = 1 \quad \forall m \in \mathcal{M} \quad (2.32)$$

The progression through the stages is formulated using implication asserts. For the initial stages the asserts are logical bi conditionals, except for the starting ones denoted as $s_{0X} \in \mathcal{S}_I : X \in \{A, B, \dots\}$, that represent the state where an evaporation plant starts after a cleaning task of each type. These can't be bi conditionals as the previous state can be several different stages. The next equations shows the formulated paths through the FSM.

$$E_{vsm} = E_{v(s+1)(m+1)} \quad (2.33)$$

$$\forall v \in \mathcal{V}, \forall s \in \mathcal{S}_l : (s+1) \neq s_{0X}, \forall m \in \mathcal{M} \setminus \{m_F\}$$

$$E_{vsm} \leq E_{v(s+1)(m+1)} \quad (2.34)$$

$$\forall v \in \mathcal{V}, \forall s \in \mathcal{S}_l : (s+1) = s_{0X}, \forall m \in \mathcal{M} \setminus \{m_F\}$$

$$E_{vsm} \leq E_{v(s+1)(m+1)} + E_{vs'(m+1)} + E_{vs''(m+1)} \quad (2.35)$$

$$\forall v \in \mathcal{V}, \forall s \in \mathcal{S}_X, \forall s' \in \mathcal{S}_{LX}, \forall s'' \in \mathcal{S}_{PX}, \forall m \in \mathcal{M} \setminus \{m_F\}$$

$$E_{vsm} \leq E_{vs(m+1)} + E_{vs'(m+1)} \quad (2.36)$$

$$\forall v \in \mathcal{V}, \forall s \in \mathcal{S}_{PX}, \forall s' \in \mathcal{S}_{LX}, \forall m \in \mathcal{M} \setminus \{m_F\}$$

$$E_{vsm} \leq E_{vs(m+1)} + E_{vs_{0X}(m+1)} \quad (2.37)$$

$$\forall v \in \mathcal{V}, \forall s \in \mathcal{S}_{PLX}, \forall m \in \mathcal{M} \setminus \{m_F\}$$

$$E_{vsm} \leq E_{vs_{0X}(m+1)} + E_{vs'(m+1)} \quad (2.38)$$

$$\forall v \in \mathcal{V}, \forall s \in \mathcal{S}_{LX}, \forall s' \in \mathcal{S}_{PLX}, \forall m \in \mathcal{M} \setminus \{m_F\}$$

This progression through states has also to be taken into account in the products assignment, as one evaporation plant cannot change products between cleanings.

$$A_{vpm} \leq A_{vp(m+1)} + E_{vs(m+1)} \quad (2.39)$$

$$\forall v \in \mathcal{V}, \forall p \in \mathcal{P}, \forall s \in (\mathcal{S}_L \cup \mathcal{S}_{PX}), \forall m \in \mathcal{M} \setminus \{m_F\}$$

To maintain control over the global production rate, a real variable is defined that represents the load associated with each evaporation plant per day and per product, $R_{vpm} : v \in \mathcal{V}, p \in \mathcal{P}, m \in \mathcal{M}$. Each evaporation plant can have different upper and lower production bounds ($UB_v, LB_v : v \in \mathcal{V}$), and all of them have a maximum capacity every day ($UB_m : m \in \mathcal{M}$) that is related to the external temperature due to the cooling tower dependency. Also, a minimum whole production rate has to be reached every day, to fulfill plannings demands ($SP_{pm} : p \in \mathcal{P}, m \in \mathcal{M}$).

$$R_{vpm} \geq LB_v \cdot A_{vpm} \quad \forall v \in \mathcal{V}, \forall p \in \mathcal{P}, \forall m \in \mathcal{M} \quad (2.40)$$

$$R_{vpm} \leq UB_v \cdot A_{vpm} \quad \forall v \in \mathcal{V}, \forall p \in \mathcal{P}, \forall m \in \mathcal{M} \quad (2.41)$$

$$R_{vpm} \leq UB_m \quad \forall v \in \mathcal{V}, \forall p \in \mathcal{P}, \forall m \in \mathcal{M} \quad (2.42)$$

$$\sum_{v \in \mathcal{V}} R_{vpm} \geq SP_{pm} \quad \forall p \in \mathcal{P}, \forall m \in \mathcal{M} \quad (2.43)$$

2.5.1 Scheduling results

Then, the mixed-integer formulation was implemented to cover the whole evaporation network. The objective function was to minimize the system consumption over a time horizon. It has to include not only the operation costs, but also the cleaning tasks costs.

First, every state of the FSM gets an associated cost depending on the evaporation plant ($C_{vs} : v \in \mathcal{V}, s \in \mathcal{S}$). Moreover, each one has an operative cost ($K_{ov} : v \in \mathcal{V}$), and one cost related to its cooling tower efficiency ($K_{cvm} : v \in \mathcal{V}, m \in \mathcal{M}$) that depends on the external temperature. The cost function corresponds to the total cost per day, computed over the future horizon \mathcal{M} .

Therefore, the MILP optimization problem to solve is:

$$\begin{aligned} \min_{f_{w_1}, T_{ow_2}} \quad & \sum_{v \in \mathcal{V}, m \in \mathcal{M}} \frac{\sum_{s \in \mathcal{S}} ((C_{vs} + K_{cvm}) \cdot E_{vsm}) + \sum_{p \in \mathcal{P}} (K_{ov} \cdot R_{vpm})}{|\mathcal{M}|} \\ \text{s.t.} \quad & (2.30) - (2.43) \end{aligned} \quad (2.44)$$

where the objective function represent the surrogate model shown in figure 2.9. The production load requirements (SP_{pm}) predictions are fixed using demands forecasting, and the temperature constraints are included assuming average weather evolution.

The optimization problem (2.44) is modeled in GAMS, an integrated development environment (IDE) focused in mathematical programming. GAMS offers interfaces to the most common NLP, MILP and MINLP solvers, v.gr., in this case the MILP

solver CPLEX was used (*ILOG CPLEX Manuals 2020*).

With this design, we could schedule one month of operation for the whole network (23 evaporation plants) with nine different products and two different cleaning types. Only one cleaning task can be performed each day. Due to physical constraints not all evaporation plants can be connected to any product, the available combinations are shown in table 2.1.

| | v_1 | v_2 | v_3 | v_4 | v_5 | v_6 | v_7 | v_8 | v_9 | v_{10} | v_{11} | v_{12} |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|----------|----------|----------|
| p_1 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| p_2 | | ✓ | | | ✓ | ✓ | | | | ✓ | ✓ | |
| p_3 | | | | | | | | | | ✓ | | ✓ |
| p_4 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| p_5 | ✓ | ✓ | ✓ | ✓ | | ✓ | | | ✓ | | ✓ | |
| Ko | 0.6 | 0.7 | 0.8 | 0.9 | 1 | 1.1 | 1.2 | 1.3 | 1.4 | 1.1 | 1.11 | 1.12 |

Table 2.1: Possible connections of products with the evaporation plants. In addition, the operative cost for every evaporation plant

| | v_{13} | v_{14} | v_{15} | v_{16} | v_{17} | v_{18} | v_{19} | v_{20} | v_{21} | v_{22} | v_{23} |
|-------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| p_1 | | | ✓ | | ✓ | | | ✓ | | ✓ | ✓ |
| p_2 | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| p_3 | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | | ✓ |
| p_4 | ✓ | ✓ | ✓ | ✓ | | | ✓ | ✓ | ✓ | | ✓ |
| p_5 | ✓ | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Ko | 1.13 | 1.14 | 1.15 | 0.9 | 1 | 1.1 | 1.2 | 1.3 | 1.4 | 0.7 | 0.8 |

Table 2.1: (*Continuation*) Possible connections of products with the evaporation plants. In addition, the operative cost for every evaporation plant

Each product has a different production rate that have to be fulfilled. In addition, every evaporation plant has lower and upper operation bounds. The lower bound

is fixed, meanwhile the upper bound depends on the external temperature. This external factor has an estimated evolution over the time horizon.

Two types of cleaning tasks have been considered, one labeled as small and one labeled as big, each with a relative cost. Also, marginal costs have been assigned to the waiting states before the cleaning tasks.

It has been observed, using the non-linear models, that operating the plant for more than 35 days is sub optimal. Hence, the maximum number of days permitted without cleaning is 40, i.e., the number of working states is 40.

If not considered, the optimizer would stop cleaning the evaporation plants during the last days as it suppose an additional cost. This would lead the system to unfeasible situations.

To prevent a point of no return, there are two main strategies: dirt stages can be forbidden for the last days, or a residual cost can be added to not clean enough states; both options having advantage and disadvantages.

In this case, the additional constraints:

$$E_{vsm_f} = 0 \quad \forall v \in \mathcal{V}, \forall s \in (\mathcal{S}_{p\mathcal{X}} \cup \mathcal{S}'_{\mathcal{X}}) : X \in \{A, B, \dots\}, X' \notin \{A\} \quad (2.45)$$

would permit the clean states or those where a small clean can be performed ($\mathcal{S}_{\mathcal{A}} \subset \mathcal{S}$). This will let the optimizer to react in the next iterations.

Figure 2.11 shows the solution obtained. The model has a size of 35 150 binary variables, 3 452 real variables and 42 613 constraints. An optimal solution was found in around 10 minutes, with a relative gap less than 1%. The horizontal axis represents the time evolution. The evaporation plants are showed in the vertical axis. Each color represents one product, and the darken of them represent the fouling evolution. Gray are the stand-by stages and cyan cells are the cleaning tasks, light and dark to represent the two different options. The number inside each cell is the production rate associated with that evaporation plant for that day.

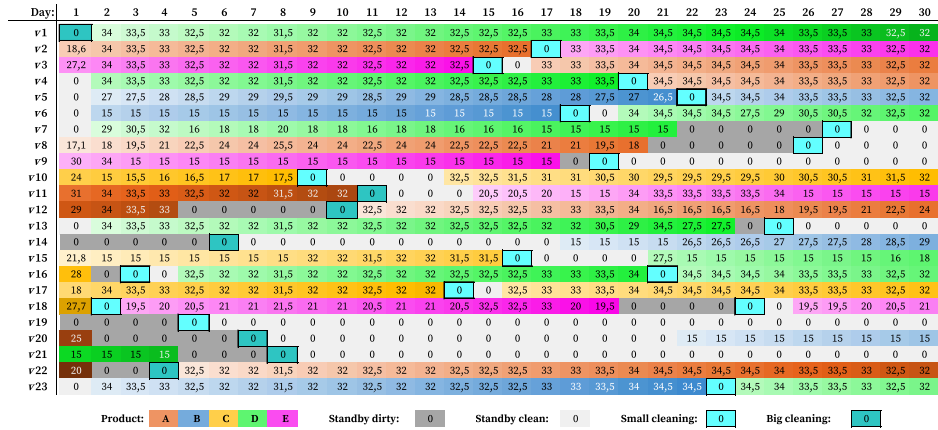


Figure 2.11: Optimal deterministic schedule example

It can be seen that the optimizer tries to maximize the production in the most efficient evaporation plants. However, when the external temperature decreases the maximum bound, the less efficient evaporation plants increase their production rate to compensate the loss in capacity in the others evaporation plants.

2.6 Stochastic approach

In order to address variations in the forecasting used to schedule an optimal planning, i.e., production rates and weather conditions uncertainties, a two-stage formulation is applied. First, one additional set is added, the scenarios \mathcal{E} , that are all the possible combinations of values that the unknown boundaries will take.

Therefore, all variables must include this new sub index, as now there are as many independent MIP problems as possible scenarios. Using the two-stage technique, variables have to be divided in *here-and-now* variables and *wait-and-see* ones (Kall et al., 1994); implementing only the first ones, and using the second ones to prepare the system to act when the moment comes, but not forcing it to adopt them. The time horizon set is divided in two subsets, $\mathcal{M}_W \subset \mathcal{M}$ denotes the robust horizon, and the rest of time samples belong to $\mathcal{M}_U \subset \mathcal{M}$, i.e., $\mathcal{M}_W \cap \mathcal{M}_U = \emptyset \wedge \mathcal{M}_W \cup \mathcal{M}_U = \mathcal{M}$. In this problem *here-and-now* variables are the choices that

have to be made on the first week (\mathcal{M}_W), meanwhile *wait-and-see* variables are the possible options that will be taken in the future (\mathcal{M}_U) in dependence of the realization of one of the possible scenarios, see figure 2.12.

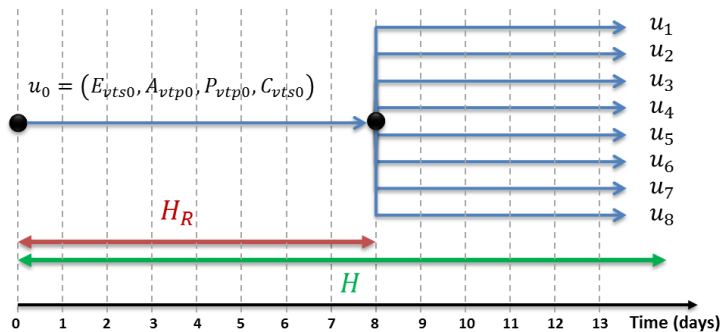


Figure 2.12: Representation of the two-stage decision variables

To implement the *here-and-now* variables, the decisions during the first week are forced to be equal irrespective of future scenarios. This applies to the decisions around what products are assigned to which evaporation plants and the evolution through the states for every evaporation plant. The rest of the days, the scenarios are independent from each other. However, as they all start from the same work point, the results will not differ too much.

$$E_{vsme} = E_{vsme'} \quad \forall v \in \mathcal{V}, \forall s \in \mathcal{S}, \forall m \in \mathcal{M}_W, \forall e, e' \in \mathcal{E} : e \neq e' \quad (2.46)$$

$$A_{vpme} = A_{vpme'} \quad \forall v \in \mathcal{V}, \forall p \in \mathcal{P}, \forall m \in \mathcal{M}_W, \forall e, e' \in \mathcal{E} : e \neq e' \quad (2.47)$$

$$R_{vpme} = R_{vpme'} \quad \forall v \in \mathcal{V}, \forall p \in \mathcal{P}, \forall m \in \mathcal{M}_W, \forall e, e' \in \mathcal{E} : e \neq e' \quad (2.48)$$

Notice that this is different to a classical robust approach as there are additional degrees of freedom in the second stage, one per scenario, that allows adapting the

solution to the realization of the uncertainty in this stage

To inspect the robustness of a solution as every scenario could present a different schedule for the days in \mathcal{M}_U , one *similarity index* (S_I) has been designed. It measures how much the different scenarios solutions look alike by comparing the similarities on the decision variables. The two-stage approach is already able to cope with disturbances if considered in the scenario tree. Nevertheless, there is always a possibility that the suggested optimal schedule cannot be fully applied when the uncertainty realization is not explicitly considered due to the discrete nature of the scheduling decisions.

A straightforward approach could consist in increasing the number of scenarios. However, this might be discarded for practical implementations. A second option that would reduce the risk of infeasibilities, is enlarging the robust horizon, to cover with only one solution the whole convex region considered. Nonetheless, this approach will vanish the benefits of the two-stage approach. The similarity index will indicate whether a suggested schedule is closer to the more risky one obtained by the standard two-stage approach, or to the risk averse single schedule.

Binary variables are hard to compare as they can be either equal or completely different. In order to ease this, the decisions are *fuzzified* based on the studies by Sakawa and Kubota (2000), where the concept of *minimum agreement index* is developed for fuzzy due date or fuzzy completion time. The similarity index is computed per day. The proximity between the same state in the different scenarios for the same evaporation plant is weighted with respect to a predefined time span, see figure 2.13a.

As it can be seen, if the solutions for all the scenarios match, the similarity index will be 100 %. Hence, the schedules coincide for all scenarios, which represents the risk-averse solution.

In our case, the time span is set to three days, the proper execution day and both days backward and forward, and the decisions are weighted by 100 % if they coincide for all the scenarios, and by 50 % if they occur with one day difference, see

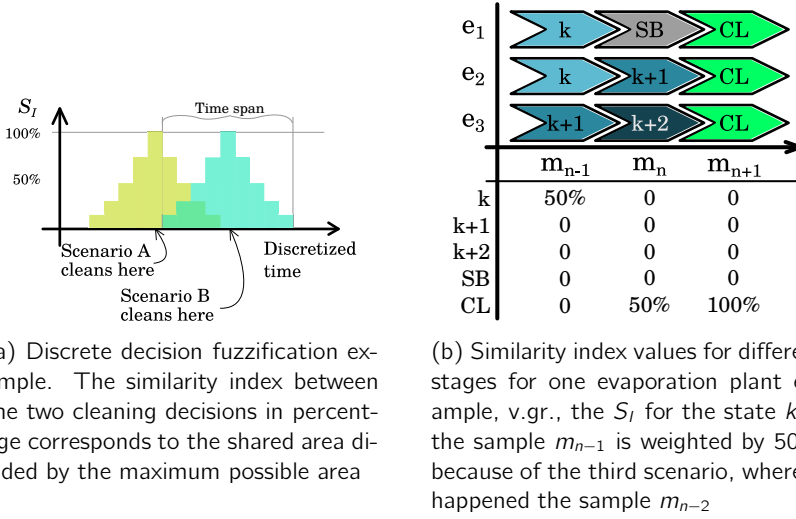


Figure 2.13: Illustrative calculations for the similarity index

figure 2.13b. The similarity index is then computed as:

$$S_I := \sum_{v \in \mathcal{V}} \sum_{s \in \mathcal{S}} \sum_{m \in \mathcal{M}_U \setminus t_F} \frac{\min_{e \in \mathcal{E}} (100 \cdot E_{vmse} + 50 \cdot E_{v(m+1)se} + 50 \cdot E_{v(m-1)se})}{n_v(200 \cdot (n_u - 1) + 150)} \quad (2.49)$$

where n_v is the number of evaporation plants, and n_u is the number of days in the uncertain area, i.e., $|\mathcal{M}_U|$.

This similarity index can be computed to measure the robustness of the solutions obtained. Nevertheless, it can be also included as a constraint to force the solutions to be similar.

2.6.1 Stochastic scheduling solution

The deterministic schedule detailed in section 2.5, uses a fixed weather prediction, and supposes that the production plannings are known over the time horizon. Practice reveals that both conjectures are not true.

To tackle this uncertainties a two-stage stochastic formulation was adopted

| | V_1 | V_2 | V_3 | V_4 | V_5 | V_6 | V_7 | V_8 | V_9 |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| p_1 | | ✓ | ✓ | ✓ | ✓ | | ✓ | ✓ | |
| p_2 | ✓ | ✓ | | | ✓ | ✓ | ✓ | ✓ | ✓ |
| p_3 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | |
| K_o | 1 | 0.88 | 1.1 | 1.01 | 0.77 | 0.95 | 1.2 | 1 | 1.05 |

Table 2.2: Possible connections of products with the evaporation plants and operative cost for every evaporation plant

considering some scenarios for each of the uncertain variables. Specifically, splitting the respective constraints into two choices instead of one known value. See figure 2.14 for a graphical explanation. It can be seen that the previous forecast is now extended to include some errors providing a sensible trustworthy region.

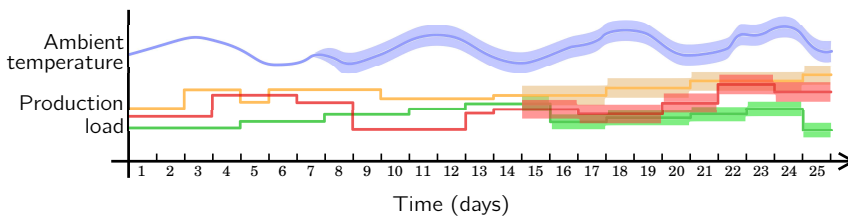


Figure 2.14: Uncertainty evolution over time example. The time is sampled in days. The weather forecast and production planning is trusted for the first week, the following week a possible region for the weather forecast is included and then sensible changes in the production load for every product is added

As the problem is linear, fulfilling both bounds will ensure the feasible solution of any middle option. Applying this to the whole system, $2^{|\mathcal{E}|}$ different combinations between the gap limits appear.

One complex model with three different products and nine evaporation plants has been implemented to test the feasibility of the approach. In table 2.2 the possible combination between products and evaporation plants are shown. Considering also the weather uncertainty, there are sixteen possible scenarios in total.

The time horizon is 25 days. The *here-and-now* variables are the decision taken for the first week, meanwhile the rest of the days are the *wait-and-see* ones. During the first week both, the temperature and the production forecast, are supposed accurate enough. Then, the next seven days, only the production rates estimation is trusted. Finally, all the values include the confidence gap. In figure 2.14 the evolution of the bounds considered is shown.

The solution obtained is plotted in figure 2.15. Again, the problem was modeled using the IDE GAMS and solved using CPLEX. The size of the problem is 130 536 binary variables, 7 994 real variables and 152 863 constraints. This optimizer took around five minutes to reach a one percent gap. Only two evaporation plants are shown instead of the whole network. The Gantt chart has a similar structure than the previous one, where the colors represent the type of product, and the darken is the fouling evolution. The horizontal axis represents the time. However, the vertical axis represents the possible scenarios. Every evaporation plant has a different Gantt chart. It can be seen that the first seven days (*here-and-now* variables) all the scenarios are forced to coincide, while in the future the optimal decisions could be adapted to the specific realization of the uncertain variables (*wait-and-see* variables).

To see the robustness of the stochastic solution, the similarity index is included in the two-stage scheduling problem, (2.49). A multi-objective problem is defined with cost functions as the economical cost, the global production rate, and the robustness measured with the similarity index.

In figure 2.16 the Pareto front is plotted. The vertical axis represents the cost, the higher the worse.

To compute the Pareto front efficiently, additional constraints with bounds in the cost functions are added to the MILP scheduling problem (Reynoso-Meza et al., 2014). The bounds to the constraints are increased for each execution, defining a grid where only one cost function is the objective for the optimization problem.

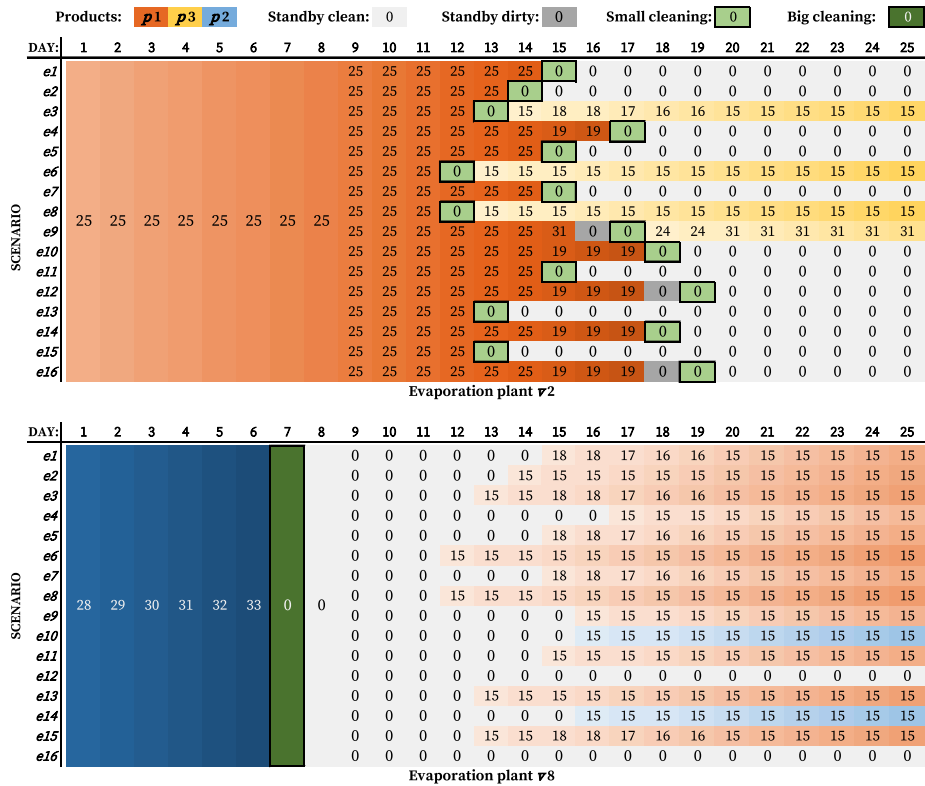


Figure 2.15: Two-stage stochastic schedule example

In this case, this Pareto front is calculated forcing a minimum similarity index and minimizing the cost, then raising it and launching again the optimization problem. This procedure is repeated until reaching the 100 % similarity.

In the solution plotted in figure 2.16 the relation between robustness and cost can be seen. When the production rate is bounded to a low rate, the optimization problem has more degrees of freedom. There are more differences between scenarios, and there is a possible gain in efficiency. On the other hand, when the production rate is fixed to a high value, there are only one possible solution to all the scenarios.

It can be also seen that, even though the cost is lower for low production rates, the efficiency is higher for high production rates, see the color legend.

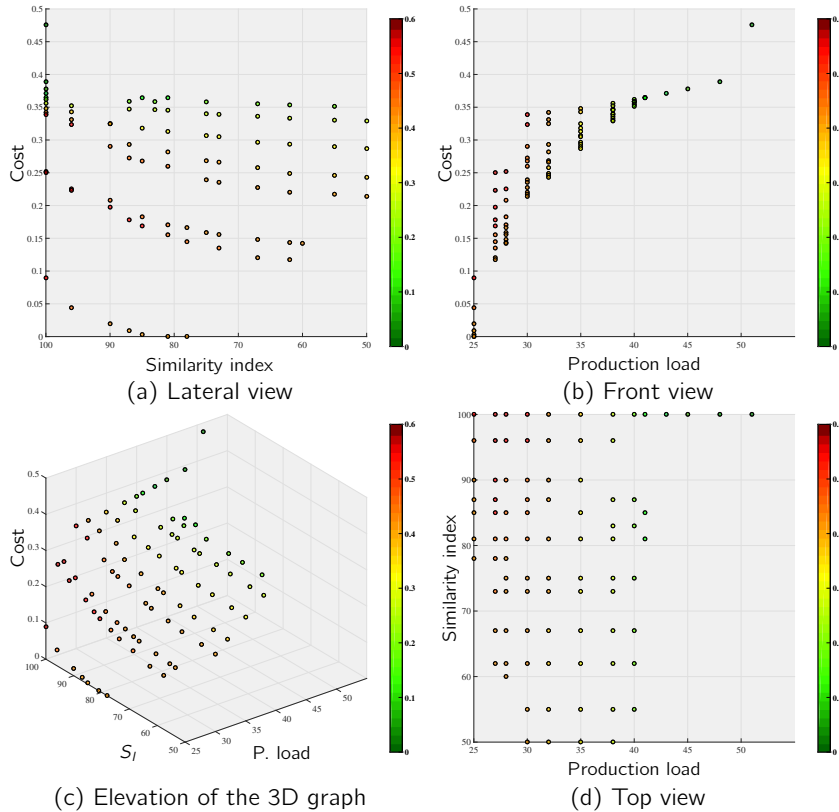


Figure 2.16: Representation of the efficiency with respect to robustness

2.7 Conclusions

Resource efficiency was achieved using a gray-box model. A methodology to obtain accurate models was described, and was proven reliable to obtain trustworthy mathematical models. The optimal working policy for the steady state was obtained. Then, the fouling dynamics were included and a new working policy that

evolves over time was obtained.

The optimization for one evaporation plant has been performed and the results implemented in the factory successfully.

The scheduling for the evaporation network is achieved. The decrease in efficiency of the plants due to fouling effect is considered. The mandatory maintenance tasks are also taken into account.

Last, two-stage stochastic scheduling formulation was implemented to tackle uncertainty in the external variables. One similarity index was developed to measure the differences between solutions to different scenarios.

Multi-objective optimization is performed to analyze the possible production situations.

Chapter 3

Real-time scheduling of batch processes in between continuous processing lines

In several industrial sectors with discrete material fluxes, such as consumer goods industries or food factories, the products require an appropriate organization of works and equipment, as the products usually have to be submitted to batch processes.

Batch processes impact significantly on intermediate storage and resources consumption of the system, the feed or the outcome of these processes are performed at specific time instants. Therefore, the synchronization with previous or following continuous lines is an important issue to consider in scheduling and planning. In general, planning manages the workflow of the plants, programs the jobs calendar and assigns equipment for orders to fulfill certain production deadline; meanwhile, scheduling manages the material movement, the start time and duration of production procedures or maintenance ones. Both are important for productivity and are in need of each other (Pegden, 2017).

Typical scheduling problem has to deal with four main questions: what tasks to perform, which equipment to assign to every task, what sequence do the tasks

have to follow, and when does every task have to start (Maravelias, 2012). The formulation used to solve the scheduling problem depends on the type of problem in the scope. For example, when tasks have to follow some kind of sequence, the formulations relies in different types of precedence, being the most extended the global precedence general (Méndez et al., 2006); meanwhile for general problems, the system usually is modeled based in reserving equipment and storage for the task (Kondili et al., 1993; Pantelides, 1994).

In this chapter, a section with an architecture of similar units in parallel is treated. The machines can work concurrently and are similar, which penalizes obtaining the optimal solution in a short time. The main problem is the one of real-time operation and adaptation to the changing conditions of the process. In figure 3.1 a graphical representation is shown, where there are several continuous producing lines feeding intermediate products to different equivalent batch units. Products have to wait in buffers, tanks or warehouses to be processed until a unit is available. There are forbidden assignments feeding line - batch unit. In some cases, products can be gathered in one unit and then be submitted to the same process.

Regarding real-time operation, we present a formulation where the scheduling is executed periodically with a short time interval, distinguishing between what is already in place and what is expected to arrive in order to improve the adaptation to the current plant conditions. This requires integration with the MES system and the results of the planning.

In addition, instead of direct task assignment to units, the schedule has to gather products and select the appropriate type of task for every group. While making groups of the same type might increase the resource consumption efficiency, mixing different products would decrease the global processing time.

Since the solution is intended to be executed in an industrial site, all the developments are designed with the objective of obtaining a real-time tool at the end. This solution will have to deal with uncertainties in the supply of raw material attached

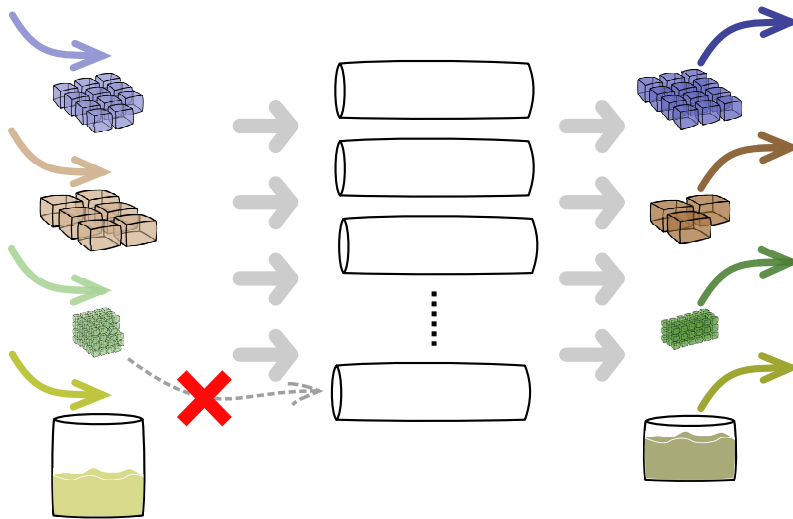


Figure 3.1: Scheme of the allocation problem, with several previous processing lines feeding different intermediate products that have to be treated

to the discrete nature of the flux. Therefore, the reschedule will be as important as the initial schedule.

The rest of the chapter is organized as follows: first the real case study where the solutions will be implemented is described; then the objectives of the thesis in this particular type of problems are listed; in the following section a simulation of the plant is presented; in Section 3.4 the methodology followed to solve the problem is described; followed by the mathematical model of the optimization problem; in the last section, some examples of the results obtained in the case study are shown.

3.1 Industrial site

The process considered in this chapter corresponds to the sterilization section of a canned sea food factory. It is located in the autonomous community of Galicia, a north-west region of Spain where there is a huge industrial fabric associated with fishing sector. In particular, the study is focused on a tuna canning plant. Final

product consists on metallic cans filled with a piece of tuna and food filling liquid, as olive oil, water or pickle sauce. There are several types of container, with different shapes and volumes, and different products with respect to the cut of the tuna that is included, v.gr., the belly is more appreciated than the loin.

Tuna is fished and deeply frozen immediately. Therefore, the initial raw product in the site is frozen fish pieces. First, in order to fulfill the production demand planned by the ERP, the required amount of fish is introduced in big thawing chambers, with a controlled humidity and temperature. Then, the fish is boiled and cleaned. The different cuts are extracted. The appropriate cuts are selected with respect to the products that are being process, and the pieces are placed into the feeders of automatic filling lines. Here, a piece of the tuna cut (pill) is introduced in every can, and these are filled up with their respective food filling liquid. The cans are then sealed.

The combination of fish cuts, filling liquids and type of containers leaves a big amount of different final products that are released by the sealing lines.

The cans must be sterilized, once they are sealed, to remove all the present microorganism that could be harmful to health and assure a long preservation time (Alonso, Arias-Méndez, et al., 2013). This is achieved by making the cans go under a thermal treatment in industrial autoclaves, or sterilization. The degree of the sterilization cycle is measured by the reduction on the number of microorganism by log cycles in minutes. This feature is called microbial lethality, or lethality to shorten (Stumbo, 1973). The lethality (\mathfrak{F}) is measured in the coldest point, the center of the tuna pill. The lethality evolution over time corresponds with a thermal death time kinetic equation (Alonso, Banga, et al., 1997), as:

$$\mathfrak{F}_0(t) = \int_0^t 10^{\frac{T(t)-T_{ref}}{z_{ref}}}$$

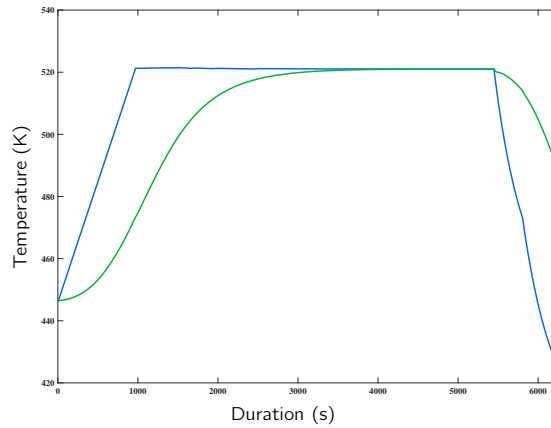
where $T(t)$ is the temperature at time instant t in the point where the lethality is being measured, and T_{ref} and z_{ref} are reference values depending on the bacterial lethality to measure. The sub-index $_0$ in lethality indicates that lethality is computed for *Clostridium Botulinum*, which is the most dangerous bacteria that is usually

found in the tuna pills, and is one of the most resistant to the thermal sterilization processes. Thereby, the reference values will be $T_{ref} = 121.1\text{ }^{\circ}\text{C}$ and $z_{ref} = 10\text{ }^{\circ}\text{C}$.

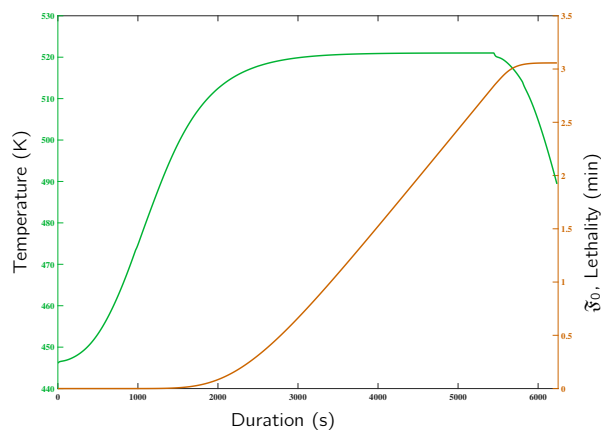
The reference temperature (T_{ref}) and the thermal resistance factor (z_{ref}) indicate the temperature change needed to reduce by ninety percent (one log-cycle) the time required to reduce the microorganism population by one log-cycle. Consequently, cans must achieve certain lethality level to fulfill sanitary constraints after been sealed, that means that they have to be heated during certain time at a high temperature, depending these values of time and temperature on the type of product or reference (type of filling liquid, cut of fish, ...). Henceforward, the combination of time and temperature set point inside the autoclaves of any sterilization cycle is called recipe. In figure 3.2b it is represented the temperature profile followed in the center of one can pill (green), and the lethality value relative to it (red). Notice that all numerical data have been scaled to guarantee complete confidentiality.

The temperature inside the cans cannot be measured online, as this would require invasive techniques that will ruin the product. Instead, the temperature evolution is simulated using accurate mathematical models. Obtaining rigorous models is a complex process, as it requires to describe the physical, chemical and biological changes in the system. Nevertheless, these models will require a subsequent refinement using real data to reduce the uncertainty present in unknown parameters (Vilas et al., 2018).

The figure 3.2a shows the temperature set point profile over time (blue) on a typical sterilization thermal process, and the temperature inside the can for this recipe (green). In this figure, it can be seen that the sterilization cycle can be divided into three stages: the heating stage, when the temperature inside the autoclave rises up to the recipe temperature set point; the maintenance or plateau stage, when the temperature is maintained during certain fixed time indicated in the recipe; and the cooling stage, that reduces the temperature as high temperatures reduce the quality of the product. The duration of the plateau stage is computed as the time needed to achieve certain lethality level for all the cans inside the autoclave, in this case around 2.8 (see figure 3.2b).



(a) Estimated temperature and temperature set point profiles



(b) Estimated temperature and its respective bacterial lethality growth over time

Figure 3.2: Graphs of the simulated temperature inside the can evolution over time

The autoclaves used in the site to perform the sterilization cycle consist of two main parts, the sterilizer *per se*, and the plate heat exchanger, see figure 3.3. The cans are introduced inside the autoclave using metallic carts, containing thousands of cans each one. These carts are automatically filled at the end of the sealing

lines. Each cart is filled by one sealing line, for this reason we can talk about type of cart referring to the type of product or reference it carries. Obviously, the total number of cans in every cart depends on the size of the can; therefore, the time that the sealing lines take to fill one cart also depends on the reference that is being produced. Then, carts arrive more or less randomly to the sterilization section and must be joined in groups of similar characteristics waiting in a storage place until a sterilizer is available. When this happens, the carts of a group are moved to a sterilizer. Once the carts have been placed inside the autoclave, its doors are closed hermetically and a control sequence is started, which begins by pumping a certain amount of water inside the autoclave. There is a closed loop circuit that recirculates the water and sprinkles it over the carts. This water is heated or cooled in a plate heat exchanger using steam or water respectively. During the heating phase, the steam consumption is almost continuous; however, during the plateau phase, the consumption is minor, as it only has to compensate the heat losses. At last, during the cooling phase, the steam circuit is opened, and cooling water is used instead. During the whole process there is an independent pressure control to prevent damages in the containers lids. Vilas Fernández and Alonso (2018) provide a more detailed explanation of the sterilization cycle.

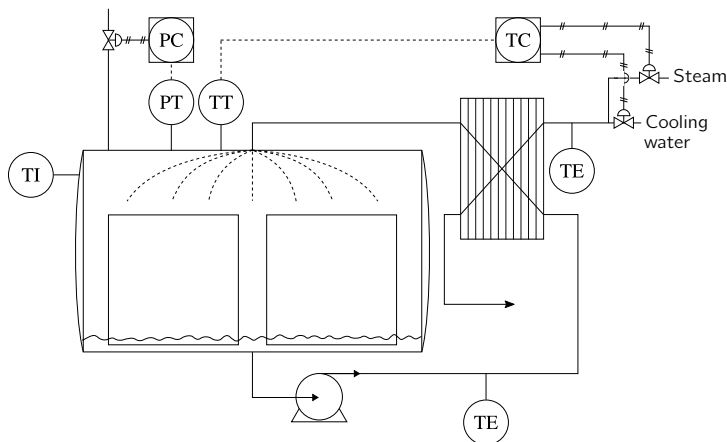


Figure 3.3: Autoclave deployment scheme

When the carts are released from the sealing lines, they may be introduced in an autoclave or wait in a kind of input buffer in front of one autoclave until one of them is available. However, the time one cart can wait in the buffer before it starts being sterilized is limited due to health reasons, which constitute one of the hard constraints of the problem. In addition, the storage space is limited; if the buffer is full, it will force to stop the previous sealing lines. Consequently, a good assignment of each cart to the right autoclave is key to maintain a high productivity in the sterilization section avoiding bottlenecks and providing safe operation.

Regarding grouping of the carts, it is important to mention that some of the different type of carts share a similar recipe, or even the same as some references only differ on labeling or packing. Those that have a cycle time that do not differ more than a few minutes, for the same temperature set point, can be gathered together and undergo the same sterilization cycle. However, this must be combined with the fact that due to space limitations, there are no feasible paths from some sealing line to some autoclaves, this constraint does not depend on the type of the carts.

3.2 Objectives

In this chapter, this thesis will provide solutions to the different scheduling problems that arise in the previously explained sterilization section, formulating an optimization problem that computes current and future decisions about the grouping and assignment of carts to the autoclaves over a prediction horizon, with the purpose of using it in real-time in near closed-loop operation mode.

The steps followed in the development of such a systems are the following ones:

First, we will implement one reliable simulation of the process with the aim of better understand the behavior of the system. Then, with the knowledge acquired with the simulation, we will compute the optimal scheduling to increase the productivity of the sterilization section.

One key point in the formulation of the scheduling problem, is the choice of the base time. We decided to select a continuous one in order to avoid the need of using

a small step required by a realistic implementation, which implies large number of variables, and correspondingly computational times, making more difficult its real-time implementation. Consequently, we followed a precedence approach for the associated assignment problem.

- To obtain a reliable model to simulate the behavior of the system
- To calculate an optimal scheduling to increase the productivity of the sterilization section

3.3 Operation simulation

The simulations of production sections constitute a valuable tool to increase the knowledge of the process within scope. They are useful to visualize different scenarios, from the default working routine to the undesirable situations. If the model developed is accurate enough, the simulation can be used to test new policies.

The ERP systems themselves model an evolution of the works to be done and the tracking of material and resources. However these systems lack on tackling devices availability and uncertainties, as them just divide the expected outcome by a theoretical production rate associated with each section of the plants. These high level programs are used to plan the overall production course, and they don't take into account the low level arrangement or scheduling.

On the other hand, even though the mathematical model of the system can be used to simulate the behavior of the system, the interface to show the solution is not simple, and does not give a global vision. This disadvantage makes the operators and plant managers to not truly trust the decision making blindly on the optimization tools.

Here arise the scheduling simulators, where the system is truly depicted. They do not only render the section by representing every device, resource and even human operator, but also implement the interactions between them in real time, i.e., they will show how an operator has to push a cart through an indicated path, or how another will have to be present in a equipment in order to fulfill the associated jobs.

These high-level simulators let the operators face the site as it will be in real world beforehand, allowing them to be prepared as much to the usual work that will occur on a daily basis as to unexpected fails that would block some equipment and have to be avoided somehow to achieve the production plan. They also provide a tool to the modeler to understand the most important constraints in the system that must be included in the optimization problem, and those that may be disregarded.

In our case study the chosen software is SIMIO (Smith et al., 2018). Among the different tools that SIMIO provides, there are probabilistic break downs of the devices, and uncertainty on the frequency of incoming material flows, making the likeness of the simulator to the real plant very high and, henceforth, the experience that the operators would get by interacting with the software.

It also offers, once a simulation has been run, tables of the results obtained. A huge amount of information can be recovered, as the sum of productive hours of every equipment or the probability of problems with the supply from previous lines or to following ones. Having a controlled model to simulate the system will allow to fairly compare between the usual methodology to handle the sterilization section and the implementation of the optimized results, that would be impossible in practice as the uncertainties in real world are too many to monitor.

The sterilization section is fully modeled inside the simulation environment SIMIO, see figure 3.5a. The model is composed by four main different parts:

- the carts and the operators. The carts are the units need to move the cans between the different processing units. The carts need an operator to push them between the units. The operators can move freely. Empty carts are picked from an empty carts buffer in the middle of the initial section. Every empty cart can be filled with any type of can
- the production section. There are eight filling units, with their respective supply lines, see figure 3.5a. Every supply line feed their respective filling unit with only one type of can for a certain period of time, but this can change according to the overall production plan. The frequency differs between lines.

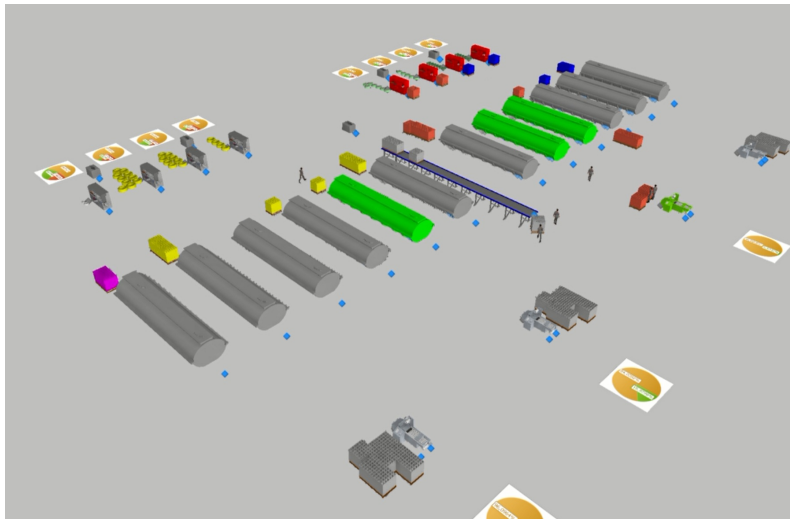


Figure 3.4: Screenshots of the SIMIO interface showing the whole sterilization section model

There are four types of cans being produced. The filling lines need an empty cart to put the cans in. Every filling unit has an stochastic duration set for every filling process, which differs depending on the type of can. Once a cart is fill with one type of can, it changes it color to ease the visualization. One empty cart is gray, meanwhile one fill cart can be: pink, yellow, blue or red

- the sterilization section. Twelve equivalent autoclaves have been included in the model, (figure 3.5b), even though the real factory operates with sixteen. Once the carts are filled, the operators push them to one of the closest sterilizers. The autoclaves wait until they reach certain number of carts introduced, then the sterilization cycle is executed. The sterilization cycle duration depends on the type of cans introduced. When it has finished the sterilizer waits until the operators remove the carts
- the packaging section. There are one packaging line for every reference (figure 3.5c). The operators push the carts after the sterilization process to the packaging lines, where the carts are emptied. Then, the carts have to be

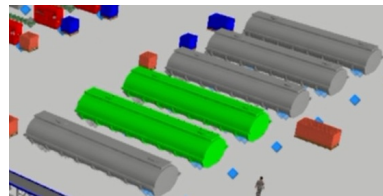
pushed again to a conveyor that takes them to the initial empty carts buffer to be used again

All the working units change their color to visualize their status. When one unit is working it is filled green, when it is idle it is filled gray, and when it is blocked it is filled red.

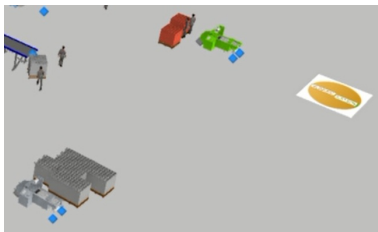
In addition to final statistics after one simulation, SIMIO also provides tools to follow the evolution of different units in real time. In the figure 3.5d the evolution of some sterilizers occupation is plotted in pie graphs.



(a) Sealing units section in detail. In red sealing units that are blocked and waiting for its already filled cart to be removed



(b) Parallel autoclaves. Some are waiting to be filled with carts (gray) and some are performing an sterilization cycle (green)



(c) Two packaging units, one is currently emptying some carts (green) and the other one is idle (gray). Some empty carts (gray) are waiting to be removed by one operator



(d) Example of the graphs tool included in the software that let the user visualize the degree of use of different units: working (green), starving (yellow) or blocked (red)

Figure 3.5: Close-up screenshots of the SIMIO interface showing different sections of the simulation

As cons of the simulator, the communication with external tools has to follow a very restrictive format. However, it presents an *application program interface*,

more commonly known as API. These routines and protocols allow the simulation program to incorporate existing complex logic. It can be used for importing or exporting data, and modify the behavior of objects in the simulation.

3.4 Methodology

First, the scheduling problem present in the sterilization section had to be classified among the different types of approaches and formulations. The problem presents a multiply parallel units architecture, with different shared resources.

Each cart that arrives to the system can be modeled as an independent task, whose deadline is the maximum waiting time. Every task has an associated recipe. The number of carts entails the number of tasks. Every autoclave is supposed as a multipurpose unit with capacity for several tasks.

The scheduling problem arises as a sequencing problem among the tasks. The precedence methodology is more suitable for this kind of models than the networks formulations (Harjunkoski et al., 2014).

The carts arrive every few minutes, however the sterilization cycles could take several hours. The differences between the time scales suggests a continuous slots formulation rather than a discrete-time approach. The tasks relative to the carts are gathered in slots or group of carts. How to form this groups is one of the key decisions of the problem. The slots are assigned to the multipurpose units or sterilizers. The maximum size of the slot depends on the capacity of the autoclaves. Every slot has an initial time instant and a duration. Both are the same for all the task gathered inside the slot.

The slots are ordered to fulfill all the tasks deadlines. Two slots cannot be performed at the same time in the same unit. The number of tasks affects the solution time. In this thesis a new precedence methodology is applied, where the slots have a predefined order and the optimizer assigns the tasks to the ordered slots. This formulation improves the sequence-position proposed by Wagner (1959), removing the binary sequence variable. Therefore, by definition, this formulation will

have less constraints than it would using a precedence formulation, while keeping a reduce number of variables.

The carts arrive from different sealing lines. Due to infrastructure limitations, some paths from sealing lines to autoclaves are forbidden. Even though the units are multipurpose, some carts cannot be assigned to some units to prevent the forbidden paths. The constraint does not depend on the type of task.

The final problem is intended to be executed in real-time. The time horizon one of the parameters that must be selected in the formulation of the problem. The choice was to use a short-time but with frequent re-scheduling adapted to the conditions of the process, operating in near closed-loop.

A similar approach has been proposed in Gupta and Maravelias (2019). The optimizer will be executed iteratively with a high frequency.

The set of tasks considered in every iteration comprises the carts that have already arrived to the sterilization section but have not been introduced in one autoclave, and the forecast of carts arrival. The inherent uncertainty in the arrival of carts makes imprudent to firmly determine future task.

The optimizer does not force the solution to set all the carts to be submitted to an sterilization process. The system can react in the future and adapt the situation to new predictions. Nevertheless, the solution has to assure that the tasks are performed before their due dates. The optimizer considers the carts up to a robust horizon, which value depends on the frequency of the optimization executions.

3.5 Mathematical Model

The system is modeled using mathematical programming. In order to speed up the computations, it is important to use linear formulation, so that the resulting problem is a MILP problem. First, all the units that belong to the sterilization section are represented as abstract sets. The gathering problem have to group the carts considering their recipes, and the sterilization cycles of the groups are scheduled to reduce the so-called *makespan* of the section. The makespan is the

time elapsed since the execution of the scheduling to the end of the last slot. It includes the possibility of combining different recipes that don't differ more than an allowed time threshold, and the physical constraints that ban the connections between every sealing line and every autoclave

First, the following sets are defined:

- \mathcal{I} , represents the carts that have to be sterilized. The set includes the carts that have already arrived, and the forecast of carts expected to leave the sealing lines
- \mathcal{J} , represents the set of groups of carts that will be gathered in the same autoclave, this means that it also corresponds with the sterilization cycles or slots. From now on, these terms will be used indistinctly
- \mathcal{K} , represents the set of autoclaves or sterilizers
- \mathcal{L} , represents the sealing lines that release the carts filled with cans
 - In addition, we use $\mathcal{L}_{k \in \mathcal{K}} \subseteq \mathcal{L}$, to represent the subset of sealing lines whose carts can be loaded inside the autoclave k , i.e., the paths to the autoclaves available for every sealing line
- \mathcal{H} , represents the set of commercial products or references, each one has a determined sterilization cycle temperature-time profile

As known parameters to the problem we have: the expected arrival time of every cart (or the actual one if the cart has already been released from the sealing lines); the type of commercial product loaded in a cart; the production line that has filled each cart; the required sterilization cycle for every reference, and the maximum number of references that can be gathered.

- $o_{i \in \mathcal{I}}$, is a real value that indicates the arriving instant of the carts; it is modeled as the time distance in minutes between the current instant and the arrival of cart i . It will be negative if the cart has already arrived, or positive if it is an estimation.

- $C_{i \in \mathcal{I}, h \in \mathcal{H}} \in \{0, 1\}$, is a binary value that defines the relation between the type of can or reference and every cart. If its value is one then the cart i is filled with cans of reference h
- $Q_{i \in \mathcal{I}, l \in \mathcal{L}} \in \{0, 1\}$, is a binary value that relates the sealing lines with the carts to indicate which line releases each cart. If its value is equal to one, then production line l has filled cart i
- $r_{d_{h \in \mathcal{H}}}$ and $r_{t_{h \in \mathcal{H}}}$, are real values that indicate the maintenance phase duration and the maintenance temperature set point required for each reference.
- $\xi_{h \in \mathcal{H}}$ and $\chi_{h \in \mathcal{H}}$, are real values that set the duration of the heating and cooling phases required for every reference respectively
- ζ , is an integer value that constraints the maximum number of different references that are allowed in one sterilization cycle
- $\Gamma_{k \in \mathcal{K}}$, is an integer value that sets the maximum number of carts that can fit in the autoclave k
- $\gamma_{k \in \mathcal{K}}$, is an integer value that determines the minimum number of carts allowed to be in one sterilization cycle in the autoclave k , this parameter helps the optimizer to discard sub optimal solutions
- θ , is a real value that indicates the maximum time that a cart might wait since it is released by the sealing lines until its sterilization cycle starts
- η , is a real value that sets the robust horizon of the solution. It will be explained in more details later
- δ , is the maximum difference in minutes between the maintenance stage duration of two carts included in the same sterilization cycle
- $\tau_{k \in \mathcal{K}}$, is a real value that indicates when one autoclave k , that is busy, will be released. Will be explained in detail later.

Carts are released by the sealing lines with different frequencies. Each producing line will have associated one reference, nevertheless more than one line can share

the same reference. Each reference has its own duration and temperature set point. However, different references might share the same recipe and differ on the container, or final package. Therefore, the first stage on the optimization of the sterilization section, is to group the carts that arrive in batches of up to Γ carts that can be submitted to the same sterilization cycle, and to choose the appropriate recipe.

To model this issue in mathematical programming language the following binary and real variables are defined:

- $V_{j \in \mathcal{J}, h \in \mathcal{H}}$, is a binary variable that indicates if group of carts j includes at least one cart of type h
- $X_{i \in \mathcal{I}, j \in \mathcal{J}}$, is a binary variable that determines if cart i belongs to group of carts j
- $Y_{j \in \mathcal{J}, k \in \mathcal{K}}$, is a binary variable which equals one if group of carts j is introduced in autoclave k
- $Z_{i \in \mathcal{I}, k \in \mathcal{K}}$, is a binary variable that indicates the relation between carts and autoclaves. This variable is the conjunction of the previous ones, i.e., $Z_{i,k} \equiv X_{i,j} \wedge Y_{j,k} \quad \forall i \in \mathcal{I}, \forall j \in \mathcal{J}, \forall k \in \mathcal{K}$, meaning that cart i is introduced in autoclave k
- $U_{j \in \mathcal{J}}$, is a binary variable that determines if a group of carts is defined, which also means if one sterilization cycle is executed or not
- $s_{j \in \mathcal{J}}$, is a real variable that specifies the start instant of the sterilization cycle of group of carts j , it is stored as the distance between the current time and the start instant
- $p_{d_{j \in \mathcal{J}}}$, is a real variable that sums up the whole sterilization cycle duration required for the group of carts j , all the stages included
- $p_{t_{j \in \mathcal{J}}}$, is a real variable that indicates the temperature set point during the maintenance stage of the group of carts j

together with the following inequalities that relate the variables between them:

$$\sum_{j \in \mathcal{J}} X_{i,j} \leq 1 \quad \forall i \in \mathcal{I} \quad (3.1)$$

$$\sum_{i \in \mathcal{I}} X_{i,j} \leq U_j \cdot \Gamma \quad \forall j \in \mathcal{J} \quad (3.2)$$

$$\sum_{i \in \mathcal{I}} X_{i,j} \geq U_j \cdot \gamma \quad \forall j \in \mathcal{J} \quad (3.3)$$

$$\sum_{k \in \mathcal{K}} Y_{j,k} = 1 \quad \forall j \in \mathcal{J} \quad (3.4)$$

Inequality 3.1 establishes that each cart must be processed only once. Note that the solution approaches the problem as a closed-loop scheduling, which means that it will be executed iteratively as time passes by; hence, the sterilization of carts is not forced considering the upcoming executions or runs of the optimizer, analogous to the concept of moving horizon in advanced control. Inequalities 3.2 & 3.3 define the amount of carts introduced in the sterilization cycles, being the upper bound (Γ) the autoclaves capacity, and the lower bound ($\gamma \geq 1$) is an optional parameter selected by the operators to prevent quasi-empty procedures. Depending on the objective function chosen such parameter may or may not be required. Equation 3.4 ensures that each sterilization cycle is assigned only to one autoclave.

$$\sum_{h \in \mathcal{H}} V_{j,h} \leq \zeta \quad \forall j \in \mathcal{J} \quad (3.5)$$

Inequality 3.5 sets the maximum number of references per sterilization cycle that is admissible (ζ). This parameter is chosen by the operators to ease the unload of the autoclaves.

$$V_{j,h} \geq C_{i,h} \cdot X_{i,j} \quad \forall i \in \mathcal{I}, \forall j \in \mathcal{J}, \forall h \in \mathcal{H} \quad (3.6)$$

$$V_{j,h} \leq \sum_{i \in \mathcal{I}} (C_{i,h} \cdot X_{i,j}) \quad \forall j \in \mathcal{J}, \forall h \in \mathcal{H} \quad (3.7)$$

The belonging of references to group of carts is constrained by a convex hull (Balas, 2018) in: (3.6), where the variable $V_{j,h}$ is set to one if one reference is included in one group of carts; and (3.7), where the variable $V_{j,h}$ is set to zero if one reference is not included in the group.

And the logical conjunction between the variables $X_{i,j}$ and $Y_{j,k}$ (i.e. the relation cart to autoclave) is included in the following inequalities:

$$Z_{i,k} \geq X_{i,j} + Y_{j,k} - 1 \quad \forall i \in \mathcal{I}, \forall j \in \mathcal{J}, \forall k \in \mathcal{K} \quad (3.8)$$

$$\sum_{k \in \mathcal{K}} Z_{i,k} \leq \sum_{j \in \mathcal{J}} X_{i,j} \quad \forall i \in \mathcal{I} \quad (3.9)$$

$$\sum_{i \in \mathcal{I}} Z_{i,k} \leq \Gamma \cdot \sum_{j \in \mathcal{J}} Y_{j,k} \quad \forall k \in \mathcal{K} \quad (3.10)$$

where the variable $Z_{i,k}$ is forced to zero for every cart that is not gathered in any group (inequality 3.9) and for every autoclave that is not related with any group of carts (inequality 3.10); and is forced to one when the group of carts which the cart belongs to, is introduced in the autoclave, $X_{i,j} = Y_{j,k} = 1$ (inequality 3.8).

Once the relation between carts and autoclaves is modeled, as each cart has its respective sealing line that has released it, the allowed path sealing lines - autoclaves can be included in the MILP problem.

$$\sum_{l \in \mathcal{L}_k} Q_{i,l} \geq Z_{i,k} \quad \forall i \in \mathcal{I}, \forall k \in \mathcal{K} \quad (3.11)$$

where the variable $Z_{i,k}$ is forced to zero if the sealing line that released the cart does not belong to the subset of sealing lines that can feed the autoclave, \mathcal{L}_k .

With the previous inequalities included in the model, the carts are gathered in groups with an acceptable size to fit in the autoclaves, and considering the allowed paths and the operator working choices (maximum different references in the same group and minimum size). However, no time constraint has been defined yet.

$$s_j \geq o_i - M_1 \cdot (1 - X_{i,j}) \quad \forall i \in \mathcal{I}, \forall j \in \mathcal{J} \quad (3.12)$$

$$s_j \leq o_i + \theta + M_1 \cdot (1 - X_{i,j}) \quad \forall i \in \mathcal{I}, \forall j \in \mathcal{J} \quad (3.13)$$

The starting time of every sterilization cycle is bounded by the carts included in its group j . It has to start before the maximum waiting-time gaps (θ) elapses for every cart included in the group, (3.13); but after all those carts have arrived to the system, (3.12).

In these inequalities one new parameter is introduced, M_1 . This parameter is needed to formulate the constraints using *big M* formulation (Winston and Goldberg, 2004), which makes the constraints idle when the term that multiply the parameter is not canceled and vice versa. The parameter has to have a value high enough to make the constraint idle, but if it meets the same order of magnitude than the rest of the values of the inequality, it can improve the computational time. Therefore, the different sets of constraints formulated using the *big M* approach will have their respective parameter, (Stein et al., 2004).

Note that the above starting time inequalities only consider the carts selected to be sterilized, not those left waiting for the next run: $i \in \mathcal{I} : \sum_{j \in \mathcal{J}} X_{i,j} = 0$. So far, no constraint has forced the carts to be included in any sterilization cycle. To prevent carts that have to start their sterilization cycles before the next run of the problem from being left out of the autoclaves, the following constraint is added:

$$o_i \geq \eta - M_1 \cdot \sum_{j \in \mathcal{J}} X_{i,j} \forall i \in \mathcal{I} \quad (3.14)$$

meaning that all carts that have arrived before the robust horizon η will be necessarily gathered in a group. The robustness horizon parameter (η) is defined relatively to the current time. It will maintain its value, but as the current time evolves, so the horizon will. As a real-time optimization tool, the optimizer should be executed at periods much lower than η to avoid infeasibilities.

On the other hand, every reference $h \in \mathcal{H}$ has a related sterilization cycle with an established recipe. As long as the temperature set point for the reference is respected (r_{th}) the procedures can be extended up to a maximum fixed span δ without a significant loss in the product quality properties, but never reduced. This feature increases the degrees of freedom in the MILP problem, and reinforces the

need for the parameter ζ , that imposes a maximum number of different references in every group of carts. Therefore, the duration of every sterilization cycle is not fixed, but bounded according to the following inequalities:

$$p_{dj} \geq \xi_h + r_{dh} + \chi_h - M_2 \cdot (1 - V_{j,h}) \quad \forall j \in \mathcal{J}, \forall k \in \mathcal{K} \quad (3.15)$$

$$p_{dj} \leq \xi_h + r_{dh} + \chi_h + \delta + M_2 \cdot (1 - V_{j,h}) \quad \forall j \in \mathcal{J}, \forall k \in \mathcal{K} \quad (3.16)$$

note from (3.16) that the allowed variation is only included in a positive sense.

Then, the temperature set point is set by:

$$p_{tj} \geq r_{th} - M_3 \cdot (1 - V_{j,h}) \quad \forall j \in \mathcal{J}, \forall h \in \mathcal{H} \quad (3.17)$$

$$p_{tj} \leq r_{th} + M_3 \cdot (1 - V_{j,h}) \quad \forall j \in \mathcal{J}, \forall h \in \mathcal{H} \quad (3.18)$$

where it is not only fixed, but it must be exactly the same for all the references included in the same sterilization cycle.

3.5.1 Ordering the slots

Once the carts have been gathered into groups considering the sealing lines, the references recipes, the arrival time and the maximum waiting time, the groups have to be ordered over the time horizon. In the literature of scheduling problems, it is usually declared additional binary variables that determine the precedence between slots (Méndez et al., 2006). With this approach becomes tedious to synchronize the whole set of slots. To overcome this, a pre-ordered slots strategy has been applied, or predefined precedence, explained in Palacín, Carlos G. et al. (2019), where the slots are neither assigned previously to any equipment nor to any sterilization recipe. Henceforth, as we can assure that every sterilization cycle will only be executed once, these procedures will occur in certain *a priori* unknown order.

First, the sterilization cycles are defined as an ordered set. These sterilization cycles coincide with the group of carts set \mathcal{J} , as only one group of carts is submitted to one sterilization cycle in one autoclave at a time. This order is extended to the slots starting time, and then used to organize the cycles that have been assigned to the same unit.

$$s_j \leq s_{j'} \quad \forall j, j' \in \mathcal{J} : j \prec j' \quad (3.19)$$

$$s_j + p_{d_j} \leq s_{j'} + M_4 \cdot (2 - (Y_{j,k} + Y_{j',k})) \quad \forall j, j' \in \mathcal{J} : j \prec j', \forall k \in \mathcal{K} \quad (3.20)$$

Inequality 3.19 extends the set order to the starting time. In (3.20) it is constrained that, if two group of carts are going to be introduced in the same autoclave, then the one that succeed cannot start before the previous one has finished its sterilization cycle, i.e., cannot start before p_{d_j} has passed since the starting instant s_j .

In addition, the first sterilization cycle in every autoclave cannot start if these are busy, v.gr. when an already started sterilization cycle has not finished yet. This is modeled in the following constraint:

$$s_j \geq \tau_k \cdot Y_{j,k} \quad \forall j \in \mathcal{J}, \forall k \in \mathcal{K} \quad (3.21)$$

where the parameter τ_k is the time in minutes that will take for autoclave k to be ready again. In this way, inequality 3.21 constraints all the sterilization cycles assigned to the autoclave k to start after τ_k .

With the previous inequalities, the gathering and scheduling sub problem is completely modeled, and could be solved by computing the MILP optimization problem:

$$\begin{aligned} & \min_{V_{j,h}, X_{i,j}, Y_{j,k}, Z_{i,k}, U_j, S_j, p_{d_j}, p_{t_j}} MK \\ & \text{s.t. (3.1) – (3.21)} \\ & MK \geq s_j + p_{d_j} \quad \forall j \in \mathcal{J} \end{aligned} \quad (3.22)$$

where the objective function is the makespan.

3.6 Examples of execution instances

In this section some results of the implementation of the previous formulation are shown.

The model guarantees that the cans will be sterilized with an adequate thermal treatment within a time window shorter than the maximum waiting time regardless of the cost function. Usually this cost function is related to economical aims, however this does not have to be the case for all scenarios.

If the workload of the sterilization section is huge, a sensible objective function to be maximized would include the production, v.gr. $\max \sum_{i \in \mathcal{I}, j \in \mathcal{J}} X_{i,j}$ will increase the number of carts per sterilization cycle. In these cases, it is recommended to include one constraint for the maximum ending time, i.e., $MK \leq 2 \cdot \eta$.

In time periods when the section has enough available working capacity, energy consumption could become the cost function. The energy consumed by every sterilization cycle depends on the maintenance stage temperature set point, see figure 4.3. As a convex non-linear function, this curve could be approximated by linear functions, in a similar way as the feasible region was approximated:

$$\begin{aligned} En_{h,j} &\geq C_{x,h} \cdot p_{t,j} + D_{x,h} - M_6 \cdot (1 - V_{j,h}) && \forall h \in \mathcal{H}, \forall j \in \mathcal{J}, \forall x \in \{1 \dots \kappa'\} \\ En_{h,j} &\geq 0 && \forall h \in \mathcal{H}, \forall j \in \mathcal{J} \end{aligned}$$

where $En_{h,j}$ is the energy consumed by the sterilization cycle j if its thermal treatment follows the reference h , the parameters $C_{x,h}$ and $D_{x,h}$ would represent the slope and the initial value of the straight lines respectively for the reference h , and κ' is the number of straight lines used to approximate the relation energy-temperature.

Then, the cost function would be determined as the aggregation of every sterilization cycle, i.e. $\min \sum_{h \in \mathcal{H}, j \in \mathcal{J}} En_{h,j}$.

3.6.1 Implementation details

Since the model has been formulated as a mathematical programming problem, it will be coded in a language based on OPL (Optimization Programming Language), (Hentenryck et al., 1999). This type of languages provides tools to ease the in-

roduction of sets, and constraints applied to the elements of those sets. They automatically repeat the desired constraints over specific subsets, which reduces the number of bugs and speed up the coding phase.

In the beginning, a specific mathematical programming software was used GAMS (Bisschop and Meeraus, 1982). This software includes interfaces to the most common solvers, and incorporates an editor to write the code inside the program HMI (human-machine interface). It is, however, a closed-source tool, and the communication to load data to or save data from the application is difficult and requires an intermediate step using a spreadsheet program.

Furthermore, it is possible to use a general purpose high level programming language, and apply frameworks or libraries that increase their potential to include an OPL translator. Such is the case of the library JuMP (Dunning et al., 2017), that let the programmer to code using mathematical programming formulations over the language julia, (Bezanson, Karpinski, et al., 2012; Bezanson, Edelman, et al., 2015). This approach increases the communication capacity of the MILP optimization, and allows to work with the solution data once the MILP solver has been executed.

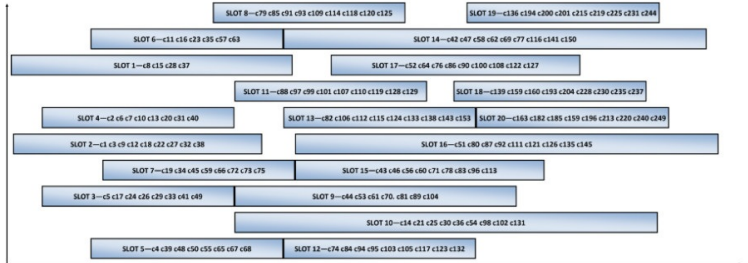
As solvers to the MILP optimization problem, two commercial solvers have been tested: CPLEX and Gurobi (*Gurobi Optimizer Reference Manual 2020*). Both have a similar performance.

3.6.2 Gathering the carts

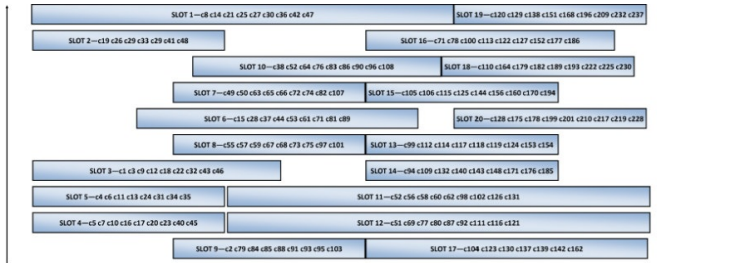
Next, we will present results of the scheduling of the operation of the sterilization section in different contexts, using both, simulated and plant data. In the later case, due to confidentially reasons, the data and results are scaled. In the first case study the consumption of shared resources is not considered. The carts are gathered in groups with a maximum difference of time in the duration of the processes of 15 minutes (δ). The maximum waiting time for all the carts is two hours ($\theta = 120$).

The sealing lines are not included in this model, therefore all the carts can be assigned to every autoclave.

The solutions obtained to this problem are similar to the ones provided by the ERP system, as it only considers the ideal working capacity of the system.



(a) Scheduling of 250 carts in twenty sterilization cycles without mixing products



(b) Scheduling of 250 carts in twenty sterilization cycles letting the optimizer to gather different products

Figure 3.6: Gantt charts of scheduling solution obtained using real plant data

In figure 3.6 two solutions to the scheduling problem are plotted. The horizontal axis represents the time evolution. The origin is the current time. The vertical axis represents the autoclaves. Two sterilization cycles that coincide in the vertical dimension will be executed in the same autoclave. The carts assigned to every slot are specified inside these.

The sterilization section considered has ten parallel autoclaves. The maximum number of carts per sterilization cycle is set to nine ($\Gamma = 9$), and the minimum is set to one, ($\gamma = 1$). The arrival of 250 carts were considered. The system has to order twenty sterilization cycles ($|J| = 20$). There are four different recipes ($|H| = 4$).

And the robust horizon is two hours.

The precedence methodology used is the global precedence, which needs a binary variable that defines if two slots precedes each other. To decrease the computing cost, the slots were forced to be executed ($U_j = 1 \forall j \in \mathcal{J}$). However, using the global precedence methodology, the time spent solving the optimization problem was around ten minutes.

In figure 3.6a all the carts submitted to the same sterilization cycle have to share the same recipe, $\delta = 0$. This makes that some sterilization cycles are executed half empty, as slot 1, since there is no carts with the same recipe expected before the maximum waiting time has passed. In figure 3.6b different recipes can be gathered together, $\delta \gg 0$. This solution fills the sterilization cycles to finish all the process sooner, and distributes better the slots among the autoclaves.

3.7 Conclusions

The study of scheduling of a sterilization section in a canning plant has been achieved successfully. Tools for the implementation in the process have been developed and tested.

One simulation was deployed to improve the knowledge of the section, and can be used in the plant as a test bed or to train new operators.

The section presented different scheduling problems that were solved, first independently and latter globally. The gathering of different tasks with similar processes to reduce the global completion time was performed, including constraints in the maximum number of different references, and the similarity between the tasks submitted to the same process.

Then, the sterilization tasks were ordered with a new precedence formulation than reduces the solving time. This formulation improves the commonly used precedence formulations reducing the number of constraints.

The solutions obtained can be used in real-time, and therefore, are suitable to be implemented in the MES system of the plant.

Chapter 4

Scheduling of batch processes with shared resources

In this chapter, the same sterilization section is treated. The distribution of shared resources is considered, which may influence the duration of the batch cycles. In figure 4.1 the same scheme than in the previous section is shown, with the inclusion of the shared resources (see figure 3.1). Here, the batch units share some resources, what may affect concurrent processes. Every product has its own consumption profile for every resource.

All units share some resources and supply lines. When some resources are shared among parallel equipment, scheduling has to consider the concurrent consumption, v.gr., if the demand surpasses the supply, the system might block, or it may happen that, in some cases the behavior of the equipment depends on the amount of resource available, in particular the duration of the cycle time, and is this feature what has to be included in the scheduling.

Imposing constraints over the total consumption of shared resources is not difficult when a discrete time formulation of the scheduling problem is chosen. But when using a continuous time base, this is much more difficult. Here we present an

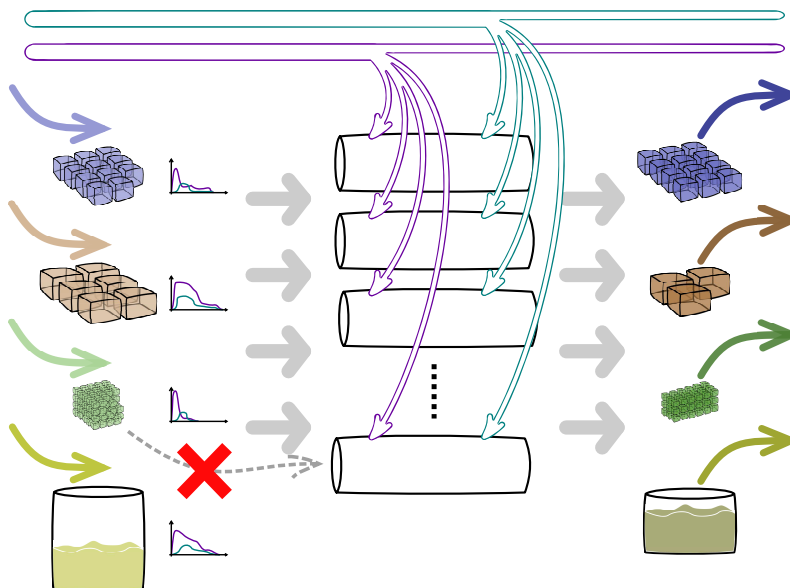


Figure 4.1: Scheme of the allocation problem, several previous processing lines feed different intermediate products to parallel batch units. The products have to be submitted to batch processes each with its own resources consumption profiles

approach that is able to deal efficiently with this difficult problem.

At the same time, we present a novel formulation for the situation in which the use of the shared resources influence the duration of the cycle time of the batch processes involved. In this chapter both approaches are studied and solutions are implemented to use in a real case study.

Since the solution is intended to be executed in an industrial site, all the developments are designed with the objective of obtaining a real-time tool at the end. This solution will have to deal with uncertainties in the supply of raw material attached to the discrete nature of the flux. Therefore, the reschedule will be as important as the initial schedule.

The rest of the chapter is organized as follows: first the description of the case study is extended; then the objectives in this particular type of problems are listed;

followed by the mathematical model of the optimization problem; in the last section, some examples of the results obtained in the case study are shown.

4.1 Case study

The process considered in this chapter corresponds to the same sterilization section.

All the autoclave's plate heat exchangers of the different autoclaves are connected to the same steam supply ring. If the consumption is too high, it may surpass the capacity of the sources. When this is the case the supply steam pressure drops, the heating phase cannot follow the optimal temperature profile and its slope decreases. This leads to an increase in this stage duration and, as the plateau phase time length is fixed, results in an overall extension on the duration of the sterilization. In figure 4.2 a comparison example is pictured: in continuous line, the normal temperature profile can be seen for a specific reference; and, in a dashed line, the temperature evolution over time if the heating resource (steam) lacks in the system. Hence, a right synchronization of the sterilization cycles to prevent exceeding a consumption bound is mandatory, or one adaptation of the scheduling to the cycle time changes created by overlapping.

Notice that the sterilization cycle has to guarantee that certain bacterial lethality has been achieved. In the current situation, the recipes have a predefined temperature set point for the plateau stage, and its duration is computed such that fulfills the lethality constraint. However, this temperature set point might be a decision variable, which would modify the duration of the sterilization cycle. In the figure 4.3 the relation temperature-duration is pictured (blue line) for one lethality bound value, it is also represented the energy consumption during the sterilization cycle (red line) with respect to the working point chosen.

This means that by proper integration of the control and scheduling levels, and taking advantage of this extra degrees of freedom that may be available, better overall operation may be achieved.

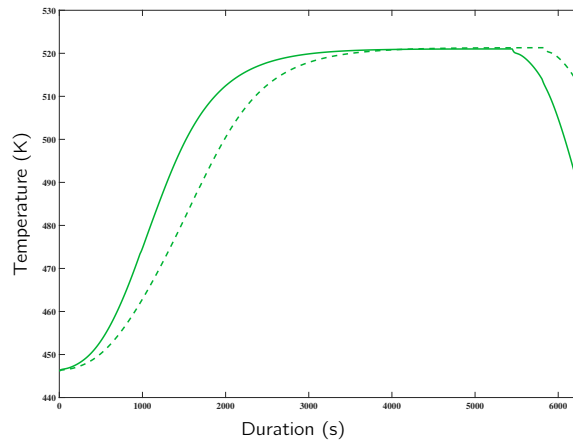


Figure 4.2: Comparison of temperature profiles over time with (continuous line) or without (dashed line) steam availability

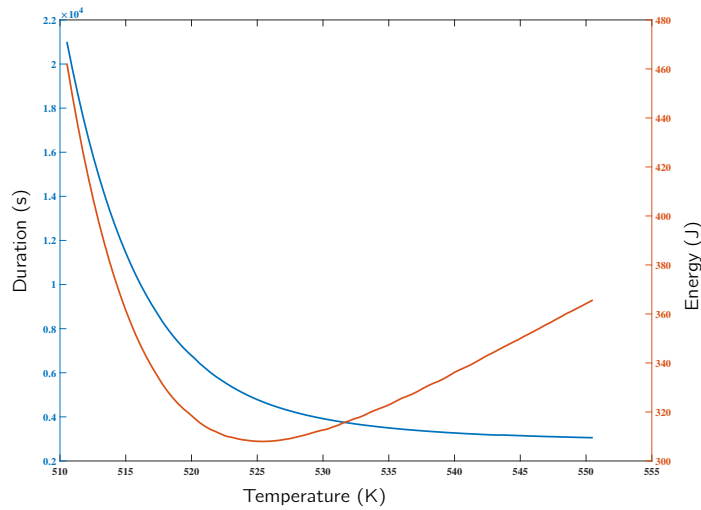


Figure 4.3: Plateau phase duration (blue) versus temperature set point for a determined lethality value and the energy consumption (red) respective to the steam required to follow the same temperature-time profile

4.2 Objectives

The shared resource consumption during the sterilization processes is an open issue in the scheduling field applied to industrial plants. Continuous consumption or production problems can be approached in several ways. In this thesis two approaches will be included: one approach tries to maintain the total consumption of shared resources at every time instant below a threshold, in order to avoid disturbances in the planned sterilizers recipe in real-time; and the other one schedule the tasks optimally considering the effects of tasks overlapping over the cycle time.

Finally, as the main matter in the thesis, the control layer actions will be integrated in the scheduling. The control of the sterilizers presents, on its own, one multi-objective problem between time, energy consumption and quality. This problem has to be included in the global schedule to obtain the real optimum of the section. This will increase the degrees of freedom provided to management.

In summary:

- To include shared resource consumption constraints
 - Approaching the system with certain maximum resource availability
 - Approaching the problem constraining the performance of the units due to a lack in the resource
- To include the control of the batch units in the global scheduling model
- To test the system with real plant data and prepare the integration in the MES system of the factory

4.3 Methodology

With the scheduling of task already implemented, in this chapter we focus in the share resources consumption. A concurrent consumption alters the duration of the slots. The optimization problem can forbid to surpass certain consumption bound. To include continuous profiles in continuous-time formulation is not trivial. The

resource level has to be considered all over the time horizon, and not only in the events needed in the precedence formulation. Adding more events than necessary can affect dramatically the computational cost. In this thesis a new approach is implemented, which combines different time scales.

This approach is useful to certain systems where this resource bound is enough to assure a high productivity and/or can be managed at will by the operators or plant managements.

Nonetheless, in the case study, the system does not provide enough steam to feed more than one sterilizer during the heating stage concurrently without alterations in the duration. To schedule the system without affecting the productivity, the alterations have to be included in the model.

In this thesis a new approach is developed, where the duration dependence on concurrent slots is included in the scheduling. This new formulation increases the range of problems manageable by mathematical programming scheduling.

Scheduling problems usually leave the control of the batch tasks aside. The batch control sets the duration of the tasks and other important features as energy consumption. To obtain a closer solution to the real optimum operation point, they should be included in the optimization model.

The duration of the tasks is not fixed anymore. The duration takes a value depending on the parameters chosen by the optimizer. The duration value becomes a feasible region constrained by the linearized relation.

The language chosen to implement the optimization problem depends on the future use of it. To perform theoretical studies a closed environment could be useful; however, to include the optimizer in a bigger system one multipurpose programming language should be preferred.

4.4 Mathematical Model

Every sterilization cycle must follow its own temperature-time profile, which implies a time varying steam consumption. As steam is a limited shared resource, if many

sterilizers coincide in their operation in time, the feeding pipes could suffer supply shortage. This implies alterations in the temperature-time profiles, which could affect the effectiveness of the thermal treatment.

All the autoclaves are connected to the same steam ring supply. The main steam consumption takes place during the heating stage, when the unit and the load itself must be heated from room temperature to the maintenance temperature. When one autoclave starts a sterilization cycle, it opens the steam supply valve that generates a drop in the pressure in the ring. If there are several pressure drops at the same time, the volumetric flow of steam to the plate exchangers is trimmed in excess. Hence, the slope of the temperature rising inside the autoclave is reduced, see figure 4.2, and the time spent heating the autoclave is increased. In order to assure fulfilling the lethality threshold (i.e., achieving a lethality \mathfrak{F}_0 higher than certain value), the operators maintain unchanged the duration of the plateau stage. This situation will alter the previously computed schedule, requiring a quick reschedule, which sometimes could become unfeasible.

As mentioned before, to tackle these alterations due to shared resources, there are two main approaches: one can constraint the resource consumption in the section to the maximum available supply, when this is the case, the temperature-time profiles in the autoclaves will remain unchanged; on the other hand, the alterations in the duration of the processes can be included in the model, in this case the MILP problem will consider the real duration of the sterilization cycles when concurrent heating phases occur.

In the next subsections both approaches are explained in details.

4.4.1 Consumption bound

New constraints will be added to the previously modeled scheduling problem, imposing that the overall consumption cannot surpass a maximum bound determined by the supply source. The scheduling MILP model must maintain a continuous constraint on the total shared resource demand. Instead of discretizing the time,

which would reduce the degrees of freedom by decreasing the maneuver range, a continuous-time formulation will obtain a closer solution to the optimum. However, the computation of continuous flow of products becomes difficult as it is only known in the predefined time events, not providing sensible estimations over time of the shared resource.

In this approach, an alternative method that simultaneously considers several time bases is implemented (César de Prada et al., 2019): one global regular discrete-time base is appended, which will help keeping track of the whole consumption; meanwhile, one non-regular discrete-time base will be assigned to every sterilization cycle, to approximate the consumption profile per slot; finally, the scheduling of sterilization cycles maintains its continuous-time formulation, to avoid losing degrees of freedom.

Thereby, in addition to the whole model explained in Section 3.5, two new sets are defined, as well as a new subset:

- $\mathcal{M} = \{0, 1, 2, \dots, \mu\}$, represents the global regular discrete-time base intervals. The number of elements will define the accuracy when computing the aggregate consumption, as it will be computed at the start of every interval $m \in \mathcal{M}$. The purpose is to use them as time instants where the availability of the total shared resources can be checked.
- $\mathcal{N} = \{-\nu, 0, 1, 2, \dots, \nu - 1, \nu\}$, represents the particular non-regular discrete-time base intervals. The size of the set will affect the precision of the approximations to the sterilization cycle consumption. The purpose is to facilitate the description over time of the shared resource consumption of every sterilization cycle to be executed
- $\mathcal{H}_h \subset \{\mathcal{H}, \emptyset\}$, represents the subset of references that have a thermal treatment more restrictive than h , but are yet allowable for the cans of type h because these cans would not suffer a high quality loss

Followed by the variables:

- $t_{\mathcal{M}m \in \mathcal{M}}$, is a real variable, represents the starting instant of every interval m . It determines the time distance between the real current time and the start instant of the interval. Hence, both time scales (the global continuous and this one discrete) coincide in their origin with the current time, i.e., $t_{\mathcal{M}0} = 0$.
- $t_{\mathcal{N}n \in \mathcal{N}}$, is a real variable, whose values correspond to the time distance between the starting time of the interval n and the starting time of its own interval 0, v.gr., $t_{\mathcal{N}2} - t_{\mathcal{N}0} = t_{\mathcal{N}2}$. This relative time base covers the global discrete-time base:

$$\begin{aligned} t_{\mathcal{N}-\nu} &\leq -t_{\mathcal{M}\mu} \\ t_{\mathcal{N}\nu} &\geq t_{\mathcal{M}\mu} \end{aligned}$$

Every sterilization cycle is connected to one non-regular base, synchronizing the starting time of the slot with the starting time of the interval 0 of that particular base. Thus, every slot has a local time scale, divided in intervals. This relation between sterilization cycles or slots and the relation between time scales is presented in figure 4.5a.

However, this time base is equal for all the different sterilization cycles, and it has to support the description of the steam profile at time instant relative to the start of operation ($t_{\mathcal{N}n}$). The thermal treatment may be different between references; hence, their steam consumption profiles will differ in the duration, amount of demand and main time instants. Therefore, the duration and the number of non-regular intervals of \mathcal{N} are chosen such that all the profiles important time points coincide with the intermediate interval start instants, $t_{\mathcal{N}n} : n \in \mathcal{N} \setminus \{-\nu, \nu\}$

- $\omega_{m \in \mathcal{M}, j \in \mathcal{J}, n \in \mathcal{N}}$ is a binary variable defined to relate the global discrete-time base with the time base relative to every slot. It identifies where any sample time of the global discrete base is located relatively to the start of every slot or sterilization cycle (s_j). It indicates if the relative time distance $t_{\mathcal{M}m} - s_j$ belongs to the interval $[t_{\mathcal{N}n}, t_{\mathcal{N}n+1})$ of the time base relative to the slot j .
- $\alpha_{m \in \mathcal{M}, j \in \mathcal{J}, n \in \mathcal{N}} \in [0, 1]$ is a real variable that weights the proximity of the

global interval start t_{M_m} to the interval start instant t_{N_n} of the time base relative to slot j , when $t_{M_m} - s_j$ belongs to the intervals $[t_{N_{n-1}}, t_{N_n})$ or $[t_{N_n}, t_{N_{n+1}})$.

- $\Upsilon_{j \in \mathcal{J}, h \in \mathcal{H}}$ is a binary variable that determines the steam flow profile needed by the thermal treatment applied to the slot j . It is a more restrictive variable than $V_{j,h}$:

$$V_{j,h} \geq \Upsilon_{j,h} \quad \forall j \in \mathcal{J}, \forall h \in \mathcal{H}$$

The precise consumption profiles are approximated to first-principle models by piece-wise linear functions. For that purpose, a new parameter $e_{n \in \mathcal{N}, h \in \mathcal{H}}$, is defined. It represents the steam flow consumption for the reference h at time instant t_{N_n} , relative to the start of operation (see figure 4.4). Note that, for the first and last intervals, the consumption is zero for all the references, since they coincide with the interval previous to the start of the sterilization cycle and the interval after sterilization is completed. Consequently, steam consumption is zero at the edges of both intervals:

$$e_{-\nu,h} = e_{0,h} = e_{\nu-1,h} = e_{\nu,h} = 0 \quad \forall h \in \mathcal{H}$$

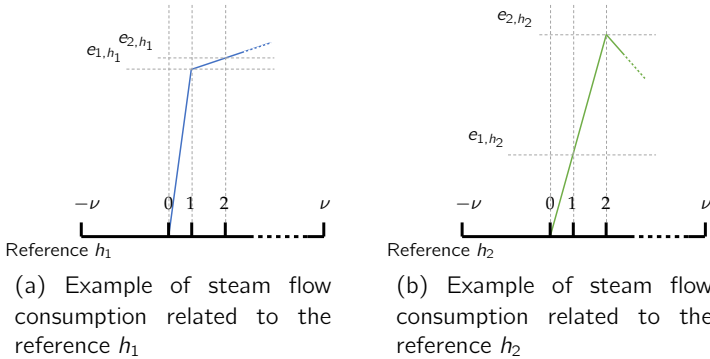


Figure 4.4: Initial intervals of the piece-wise linear functions that approximate the steam consumption profiles

The variables are related by the following inequalities:

$$\sum_{n \in \mathcal{N}} \omega_{m,j,n} = 1 \quad \forall m \in \mathcal{M}, \forall j \in \mathcal{J} \quad (4.1)$$

$$\omega_{m,j,\nu} = 0 \quad \forall m \in \mathcal{M}, \forall j \in \mathcal{J} \quad (4.2)$$

Every global time sample must be in a single interval of every slot relative time base, (4.1). Note that, as the variable $\omega_{m,j,n}$ indicates that the time sample belongs to the interval that has just started, and as the relative time base covers the global one ($t_{\mathcal{N}\nu} \geq t_{\mathcal{M}\mu}$), it is impossible for a time sample to happen after $t_{\mathcal{N}\nu}$, (4.2).

$$t_{\mathcal{M}m} - s_j = \sum_{n \in \mathcal{N}} (\alpha_{m,j,n} \cdot t_{\mathcal{N}n}) \quad \forall m \in \mathcal{M}, \forall j \in \mathcal{J} \quad (4.3)$$

$$\alpha_{m,j,n} \leq \omega_{m,j,n-1} + \omega_{m,j,n} \quad \forall m \in \mathcal{M}, \forall j \in \mathcal{J}, \forall n \in \mathcal{N} \setminus -\nu \quad (4.4)$$

$$\alpha_{m,j,-\nu} \leq \omega_{m,j,-\nu} \quad \forall m \in \mathcal{M}, \forall j \in \mathcal{J} \quad (4.5)$$

$$\sum_{n \in \mathcal{N}} \alpha_{m,j,n} = 1 \quad \forall m \in \mathcal{M}, \forall j \in \mathcal{J} \quad (4.6)$$

One time sample is defined as a linear combination of the edges of the interval it belongs to, (4.3). Therefore, all the weighting variables must be zero but the ones relative to both edges of the interval, (4.4) and (4.5). The aggregation of α must be equal to one, (4.6). For clarifications see figure 4.5, where a little example is pictured.

Since any global time sample can be computed related to the evolution of every sterilization cycle, and as the steam flow consumption has been approximated by piece-wise linear functions, the consumption in every global sample time can be calculated as a linear combination of the consumption at the edges of the interval the sample time belongs to. However, there are as many different piece-wise functions as references, $e_{n \in \mathcal{N}, h \in \mathcal{H}}$. In order to keep linearity, avoiding product of decision variables in the calculation of each slot consumption, the real variable $\alpha_{m,j,n}$ is modified to include the type of reference, so that it becomes $\alpha_{m,j,n,h \in \mathcal{H}}$. This new variable will only weight the edges of the interval for the appropriate reference. The

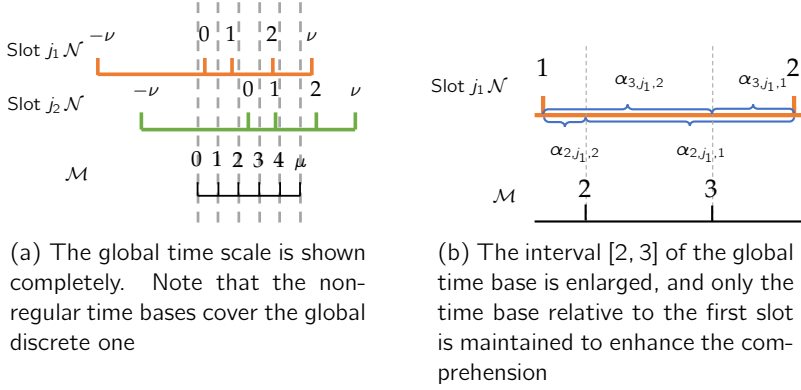


Figure 4.5: Two non-regular time bases relative to the slots j_1 and j_2 are plotted over the global time base. Both relative time bases are similar. It can be seen that the length of \mathcal{N} is bigger than twice \mathcal{M} . In the right subplot, the value of α are shown. Notice that $t_{\mathcal{M}2} - s_{j_1} = (\alpha_{2,j_1,1} \cdot t_{\mathcal{N}1}) + (\alpha_{2,j_1,2} \cdot t_{\mathcal{N}2})$ and $t_{\mathcal{M}3} - s_{j_1} = (\alpha_{3,j_1,1} \cdot t_{\mathcal{N}1}) + (\alpha_{3,j_1,2} \cdot t_{\mathcal{N}2})$

equations and inequalities (4.3) - (4.6) become:

$$t_{\mathcal{M}m} - s_j = \sum_{n \in \mathcal{N}, h \in \mathcal{H}} (\alpha_{m,j,n,h} \cdot t_{\mathcal{N}n}) \quad \forall m \in \mathcal{M}, \forall j \in \mathcal{J} \quad (4.7)$$

$$\sum_{h \in \mathcal{H}} \alpha_{m,j,n,h} \leq \omega_{m,j,n-1} + \omega_{m,j,n} \quad (4.8)$$

$$\forall m \in \mathcal{M}, \forall j \in \mathcal{J}, \forall n \in \mathcal{N} \setminus -\nu$$

$$\sum_{h \in \mathcal{H}} \alpha_{m,j,-\nu,h} \leq \omega_{m,j,-\nu} \quad \forall m \in \mathcal{M}, \forall j \in \mathcal{J} \quad (4.9)$$

$$\sum_{n \in \mathcal{N}, h \in \mathcal{H}} \alpha_{m,j,n,h} = 1 \quad \forall m \in \mathcal{M}, \forall j \in \mathcal{J} \quad (4.10)$$

where $\alpha_{m,j,n}$ has been replaced by $\sum_{h \in \mathcal{H}} \alpha_{m,j,n,h}$.

When gathering the carts, the references were mixed in the same group of carts, with a maximum number of references per group. A specific reference related to every group of carts were not determined. The time and temperature for the

sterilization cycle were chosen such that fulfill all the requirements for the load. Considering the subset \mathcal{H}_h , the most restrictive temperature-time profile can be selected as:

$$\sum_{h \in \mathcal{H}} \Upsilon_{j,h} = 1 \quad \forall j \in \mathcal{J} \quad (4.11)$$

$$V_{j,h} - \Upsilon_{j,h} \leq \sum_{h' \in \mathcal{H}_k} \Upsilon_{j,h'} \quad \forall j \in \mathcal{J}, \forall h \in \mathcal{H} \quad (4.12)$$

$$\alpha_{m,j,n,h} \leq \Upsilon_{j,h} \quad \forall m \in \mathcal{M}, \forall j \in \mathcal{J}, \forall n \in \mathcal{N}, \forall h \in \mathcal{H} \quad (4.13)$$

Every sterilization cycle must have only one prevailing reference, (4.11). If one reference is included in one group of carts, the prevailing reference will be the more restrictive one, (4.12). Once the reference for every sterilization cycle is determined, only the variables relative to the selected type are allowed to be positive, (4.13).

Thus, the resource consumption of any sterilization cycle j_1 in every time sample can be computed as:

$$\sum_{n \in \mathcal{N}, h \in \mathcal{H}} (\alpha_{m,j_1,n,h} \cdot e_{n,h}) \quad \forall m \in \mathcal{M}$$

See figure 4.6, where, as an example, the consumption relative to the slot j_1 , whose reference is h_1 , is computed at the global time sample 3 as the linear combination of the consumption at the time samples 1 and 2 relative to the slot j_1 .

Finally, denoting as ϱ the maximum amount of shared resource available at every time sample, the overall demand is constrained as:

$$\sum_{n \in \mathcal{N}, j \in \mathcal{J}, h \in \mathcal{H}} (\alpha_{m,j,n,h} \cdot e_{n,h}) \leq \varrho \quad \forall m \in \mathcal{M} \quad (4.14)$$

The whole optimization problem that would replace (3.5.1), can be formulated

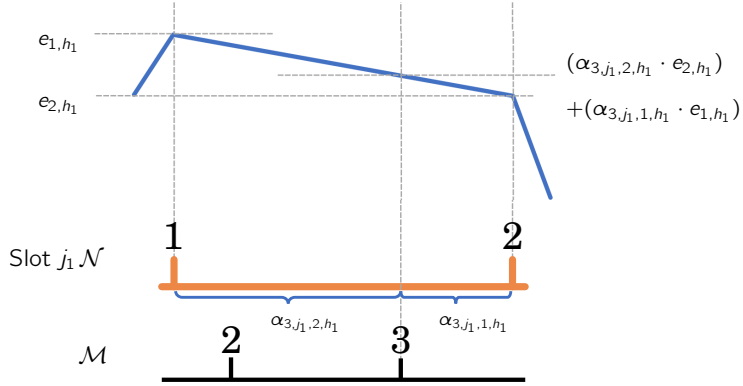


Figure 4.6: Example of resource consumption modeling via piece-wise linear function and variable α labeling as h_1 the reference corresponding to the slot j_1

as:

$$\begin{aligned}
 & \min_{V_{j,h}, X_{i,j}, Y_{j,k}, Z_{i,k}, U_j, S_j, P_{d_j}, P_{t_j}, t_{M_m}, t_{N_n}, \omega_{m,j,n}, \alpha_{m,j,n,h}, \Upsilon_{j,h}} MK \\
 & \text{s.t. (3.1) - (4.2)} \\
 & \quad (4.7) - (4.14) \\
 & \quad MK \geq s_j + p_{d_j} \quad \forall j \in \mathcal{J}
 \end{aligned} \tag{4.15}$$

Furthermore, once there is only one reference related to every sterilization cycle, some of the previous *big M* formulated constraints can be replaced by the following ones, that use the convex hull approach:

$$p_{d_j} = \sum_{h \in \mathcal{H}} ((\xi_h + r_{d_h} + \chi_h) \cdot \Upsilon_{j,h}) \quad \forall j \in \mathcal{J} \tag{4.16}$$

$$p_{t_j} = \sum_{h \in \mathcal{H}} (r_{t_h} \cdot \Upsilon_{j,h}) \quad \forall j \in \mathcal{J} \tag{4.17}$$

This formulation will replace inequalities (3.15) - (3.18).

Note that, even though by (3.2) & (3.3) the sterilization cycles are not forced to be executed, in (4.11) one consumption profile is chosen. This issue can cause

a consumption aggregation related to an empty slot. In order to solve this matter, one new virtual reference (h_0) might be included with a related consumption of zero, i.e., $e_{n,h_0} = 0 \forall n \in \mathcal{N}$. This virtual reference can be forced when the sterilization cycle is not used.

$$\Upsilon_{j,h_0} = 1 - U_j \quad \forall j \in \mathcal{J}$$

4.4.2 Variable processing time

In some cases, preventing the system to surpass the availability bound is unavoidable. This situation may cause alterations in the predefined profiles, by enlarging the duration of the batch processes. When this happens, the schedule has to consider the interference.

In the tuna canning plant, if several sterilizers are in the heating stage at the same time, the overall demand may surpass the capacity of the steam supply line. This will reduce the steam pressure, and increase, for all the slots affected, the time required to achieve the temperature set point of the recipes, see figure 4.2. However, as the consumption during the maintenance stage is low, its control will not be affected, nor the control of the cooling phase will, due to enough available cooling water.

These drops of pressure in the steam ring supply will occur always that a sterilization cycle starts, as the steam consumption has a peak at the beginning of the cycle. All the heating phases running at that instant will be affected, even those which were supposed to be finished, but had been extended due to other concurrent heating stages. Hence, the interference between slots *cannot be computed offline* by a difference in starting times.

As the drop in the steam supply ring pressure depends on the number of sterilizers connected, the alterations in the duration are modeled as a linear function of the concurrent heating phases. The duration of the heating stage (ξ_h) will be divided into:

- ξ_{f_h} a minimum fixed value, equal to the value of the heating stage when the sterilization cycle is executed alone (there is enough steam supply to follow

the temperature set point)

- ξ_{ρ_h} a value that represents the effect of two overlapping heating stages

Figure 4.7 shows different possibilities of overlapping among operations of four similar sterilization cycles.

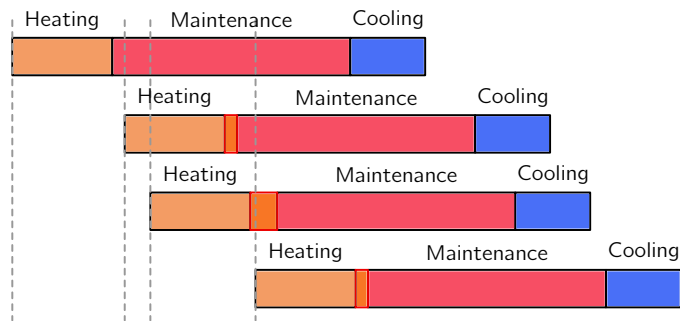


Figure 4.7: Graphical representation of the influence among four slots. Each slot is represented by a bar with colors orange, red and blue corresponding to the heating, maintenance and cooling phases respectively. Heating stage in the first slot does not coincide with the heating stage of any other slot, therefore its heating time remains unchanged. Meanwhile, the heating phases of the second and fourth slots are executed simultaneously to the third one, ergo their heating phases are increased by ξ_p each (dark-orange with red outline rectangle); finally, the third slot coincides with both of the previously commented slots, so its duration is increased twice

Accordingly, instead of using the formulation presented in the previous subsection constraining the consumption, the gathering of carts (Section 3.5) is modified to include the alteration on the duration of the batch processes when there are concurrent consumption.

Two new variables are added:

- $\xi_{h \in \mathcal{H}, j \in \mathcal{J}}$ is a real variable that equals the duration in minutes of the heating stage that will suffer the group of carts j , it replaces the parameter ξ_h
- $W_{j \in \mathcal{J}, j' \in \mathcal{J}: j \prec j'}$ is a binary variable that equals one when two slots coincide in their heating stage. As the slots are affected independently of which one was initiated first, the variable is only defined for successive slots. In addition, it is

only defined for the next $|\mathcal{K}| - 1$ slots (the number of sterilizers minus one), which is the maximum number of sterilization cycles that can coincide

This variables are related by:

$$s_j + \xi_j \geq s'_j - M_5 \cdot (1 - W_{jj'}) \quad \forall j, j' \in \mathcal{J} : j \prec j' \wedge dist(j, j') < |\mathcal{K}| \quad (4.18)$$

$$s_j + \xi_j \leq s'_j + M_5 \cdot W_{jj'} \quad \forall j, j' \in \mathcal{J} : j \prec j' \wedge dist(j, j') < |\mathcal{K}| \quad (4.19)$$

where $dist(j, j')$ represents the distance between the slots j and j' , meaning as distance the number of elements between them in the ordered set \mathcal{J} . In addition to the order of the sterilization cycles (constrained in (3.19)), these inequalities force the heating stage of the first slot to finish after the second slot has started if $W_{jj'} = 1$, (4.18), or before if the binary variable equals zero, (4.19).

$$W_{jj'} \geq W_{jj''} \quad \forall j, j', j'' \in \mathcal{J} : j \prec j' \prec j'' \quad (4.20)$$

$$W_{j'j''} \geq W_{jj''} \quad \forall j, j', j'' \in \mathcal{J} : j \prec j' \prec j'' \quad (4.21)$$

To reduce the computational cost, some constraints can be added in order to remove unfeasible solutions, (4.20) and (4.21), which consider that, if two slots coincide in the heating stage, all the intermediate slots will coincide too.

$$\xi_{hj} = \xi_{fh} + \xi_{ph} \cdot \left(\sum_{j' \in \mathcal{J} : j' \prec j} W_{j'j} + \sum_{j'' \in \mathcal{J} : j \prec j''} W_{jj''} \right) \quad \forall j \in \mathcal{J}, \forall h \in \mathcal{H} \quad (4.22)$$

$$\rho_{dj} \geq \xi_{hj} + r_{dh} + \chi_h - M_2 \cdot (1 - V_{j,h}) \quad \forall j \in \mathcal{J}, \forall h \in \mathcal{H} \quad (4.23)$$

$$\rho_{dj} \leq \xi_{hj} + r_{dh} + \chi_h + M_2 \cdot (1 - V_{j,h}) \quad \forall j \in \mathcal{J}, \forall h \in \mathcal{H} \quad (4.24)$$

In (4.22) the duration of the heating phase is computed, as the fixed value plus the proportional part times the number of concurrent heating stages. This time is performed for all the possible references, as each one has a different heating profile and it would suffer different alterations. Then, the inequalities that constrained the duration of the process, (3.15) and (3.16), will be modified to include the new

variable $\xi_{h,j}$, (4.23) & (4.24).

The optimal schedule for the section, considering the extensions in the heating stages due to concurrent steam consumption, could be solved as:

$$\begin{aligned}
 & \min_{V_{j,h}, X_{i,j}, Y_{j,k}, Z_{i,k}, U_j, S_j, P_{d_j}, P_{t_j}, W_{j,j'}, \xi_{h,j}} MK \\
 & \text{s.t. (3.1) – (3.14)} \\
 & \quad (3.17) – (3.21) \\
 & \quad (4.18) – (4.24) \\
 & \quad MK \geq s_j + p_{d_j} \quad \forall j \in \mathcal{J}
 \end{aligned} \tag{4.25}$$

Nevertheless, this approximation is an overestimation of the duration of the heating stages, as two slots will be equally affected by ξ_{p_h} whether they coincide in the start of the heating stages or only in the last few minutes of the first one and the initial ones of the second slot.

To reduce the conservatism of the formulation, the affected stages can be split into different divisions, each with a different proportional increase to the duration. And include as many binary variables as divisions are, to mark where the slots coincide.

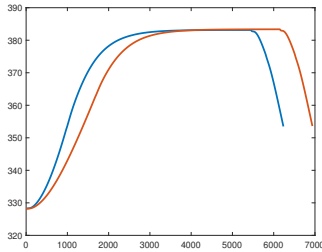
For example, if the heating stage were split into two regions, there will be needed two binary variables: $W_{1j,j'}$, that will determine if j' starts before the first region of j has passed; and $W_{2j,j'}$, that will value one if j' starts before j has finished its heating phase.

$$\begin{aligned}
 \xi_{h,j} &= \xi_{f_h} + \xi_{p1_h} \cdot \left(\sum_{j' \in \mathcal{J}: j' \prec j} W_{1j',j} + \sum_{j'' \in \mathcal{J}: j \prec j''} W_{1j,j''} \right) \\
 &+ \xi_{p2_h} \cdot \left(\sum_{j' \in \mathcal{J}: j' \prec j} W_{2j',j} + \sum_{j'' \in \mathcal{J}: j \prec j''} W_{2j,j''} \right) \\
 &\forall j \in \mathcal{J}, \forall h \in \mathcal{H}
 \end{aligned}$$

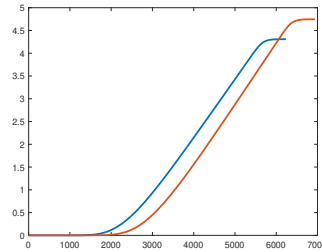
If both slots coincide during most part of the heating phase, and supposing there

are no more slots, the final heating duration would be $\xi_{fh} + \xi_{p1h} + \xi_{p2h}$; meanwhile, if they coincide only during the second defined region, the heating duration would be $\xi_{fh} + \xi_{p2h}$.

4.4.3 Reducing quality loss due to altered profiles



(a) Temperature set point profile over time for the default sterilization cycle (blue), and for a case where the heating phase has taken more time but the duration of the maintenance stage is not modified (red)



(b) Lethality growth over time for cans submitted to the temperature-time profiles shown in the left subplot

Figure 4.8: Graphs comparing the optimal case and the worst case of sterilization cycles in case of concurrent heating phases when the duration of the maintenance is the constraint

The thermal treatment required to fulfill the lethality threshold, besides killing the microorganism, reduces the quality of the food inside the cans, hence the duration of the sterilization cycle should be minimized as much as possible. The maintenance stage is performed for the time required to achieve certain lethality level, see figure 3.2b.

Nevertheless, when the heating time duration is increased due to concurrent sterilization cycles, the duration of the maintenance stage is kept untouched to guarantee the lethality is achieved. As the product inside the cans is already being heated during the first stage, the microbial lethality level has already started to grow

before the maintenance stage begins. Hence, when the sterilization cycle ends, the lethality level has surpassed the threshold value, see figure 4.8.

Consequently, in order to reduce energy consumption and to increase final product quality, the time of the whole process could be decreased to obtain a profile similar to the one shown in figure 4.9, where the lethality is achieved, but the total duration is minimized.

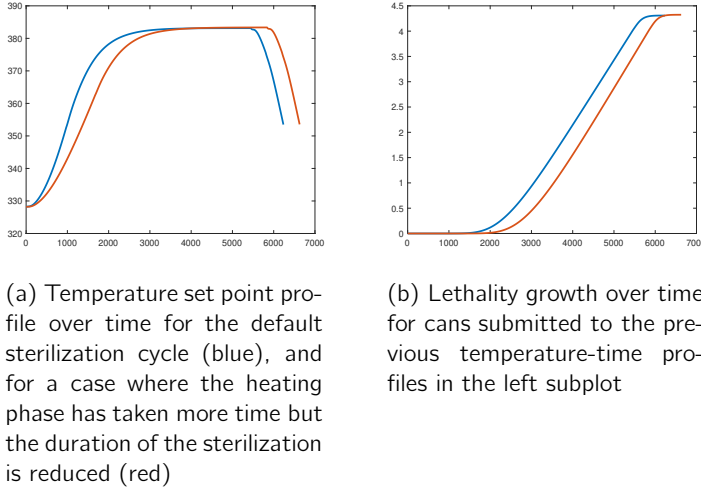


Figure 4.9: Graphs comparing the optimal case and the worst case of sterilization cycles in case of concurrent heating phases when the lethality is the constraint

To perform the reduction in the maintenance stage duration, one new variable is defined:

- $r_{dh,j}$ is a real variable that represents the duration of the maintenance stage of the slot j if the reference of the slot is h

The value of the new variable is computed by:

$$r_{dh,j} = r_{dh} - \rho_h \cdot \left(\sum_{j' \in \mathcal{J}: j' \prec j} W_{j',j} + \sum_{j'' \in \mathcal{J}: j \prec j''} W_{j,j''} \right) \quad \forall j \in \mathcal{J}, \forall h \in \mathcal{H} \quad (4.26)$$

where r_{dh} is the default duration of the maintenance stage for the reference h , and ρ_h is a new parameter equal to the number of minutes that the maintenance has to be reduced, if the heating stage is increased by $\xi_{\rho h}$.

This change in the duration of the maintenance will modify the duration of the sterilization cycle as:

$$p_{dj} \geq \xi_{hj} + r_{dhj} + \chi_h - M_2 \cdot (1 - V_{j,h}) \quad \forall j \in \mathcal{J}, \forall h \in \mathcal{H} \quad (4.27)$$

$$p_{dj} \leq \xi_{hj} + r_{dhj} + \chi_h + M_2 \cdot (1 - V_{j,h}) \quad \forall j \in \mathcal{J}, \forall h \in \mathcal{H} \quad (4.28)$$

replacing (4.23) & (4.24).

4.4.4 Integrating control in the scheduling

Usually schedule is highly constrained by the job orders that comes from the ERP system, and it considers the control of the units as fixed values. These artificial constraints prevent the system from obtaining a better plant optimum for the sections or departments.

In the previous subsections, the job orders are altered inside the optimization problem by mixing the different references inside the autoclaves. This improves the final objective cost at the expense of increasing computational time. In addition, the presented algorithms are able to tackle this without a significant rise in solving time. In this subsection, the lower control layer is integrated in the scheduling problem.

The temperature-time profile required for every reference is chosen such that the lethality constraint is achieved. When the temperature set point during the maintenance stage is constant, obtaining this lethality can be performed by submitting the cans either to a very high temperature for a relatively short time or to a lower temperature for a longer time, see figure 4.3. This relation, between temperature set point and duration of the maintenance stage, draws a curve that represents the minimum operation point that fulfills the lethality constraint. However, all the points over the line will submit the cans to a more restrictive thermal treatment

and, consequently, they are feasible operation points, see figure 4.10.

Using an accurate enough model for the sterilization section, which considers the heat transmission inside the food cans, the relation temperature-time can be obtained for all the references (Vilas Fernández and Alonso, 2018). The feasible region drawn is a convex non-linear space. In order to include it in the MILP model, it is inner approximated by linear functions that will outline the bound of the region, see figure 4.10a. The number of linear functions used will determine the accuracy of the model, but they will increase the complexity of the MILP problem.

In addition to the lower bounds, one additional upper bound is included to fully define the feasible region. This upper bound will decrease the computational cost by rejecting *a priori* known sub-optimal solutions. Even though operation points far away from the curve are feasible thermal treatments, these will increase the energy consumption and decrease the quality of the final product. Therefore, a sensible region will be near the curve.

When more than one reference are gathered into the same group of carts, the operation point has to be inside both feasible regions, see figure 4.10b.

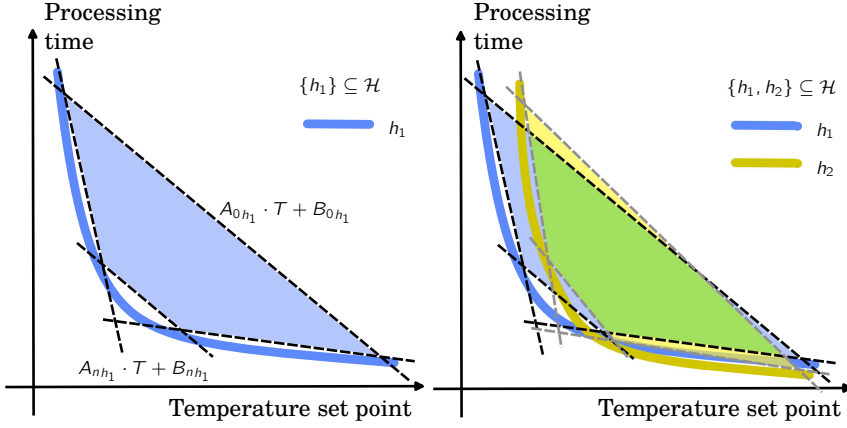
The following formulation extends the one from the section 3.5.

A new pair of parameters are defined to define the feasible areas for every reference:

- $(A_{x,h \in \mathcal{H}}, B_{x,h \in \mathcal{H}}) : x \in \{0 \dots \kappa\}$ are real parameters that describe the linear functions that outline the feasible region for the reference h . A is the slope of the function, and B is the intersection with the vertical axis. The parameter κ is the number of linear functions that will approximate the non-linear curve. The upper bound is determined for the first combination of parameters, i.e., $x = 0$. In this model, temperature has been chosen as the independent variable, so the bounds that the linear functions describe are:

$$\text{duration} = A_{x,h} \cdot \text{temp. set point} + B_{x,h} \quad \forall x \in \{0 \dots \kappa\}, \forall h \in \mathcal{H}$$

And the duration of the process, that was first defined as a known parameter, is now treated as a new variable:



(a) Feasible area to select the operation point for reference h_1 . The combination temperature-time has to be over the blue line. The area is approximated by several linear functions

(b) Combination of two different references feasible areas. The feasible region approximation for h_1 is colored in light blue, meanwhile the approximation for h_2 is colored in light-yellow. The feasible area for both can be seen as the green area, and covers the operation points over both lines

Figure 4.10: Feasible area approximation for different references

- $r_{dh,j}$ is a real variable that represents the duration of the maintenance stage of sterilization cycle j if it performs the thermal treatment relative to reference h . It replaces r_{dh}

The relation between the duration of the maintenance stage and the temperature set point is modeled as follows:

$$r_{dh,j} \geq A_{x,h} \cdot p_{tj} + B_{x,h} \quad \forall h \in \mathcal{H}, \forall j \in \mathcal{J}, \forall x \in \{1 \dots \kappa\} \quad (4.29)$$

$$r_{dh,j} \leq A_{0,h} \cdot p_{tj} + B_{0,h} \quad \forall h \in \mathcal{H}, \forall j \in \mathcal{J} \quad (4.30)$$

The inequalities that constrained the temperature for every sterilization cycle are removed from the model ((3.17) & (3.18)). And the constraints (3.15) and (3.16), which set the bounds to the whole duration of the sterilization cycle, have

to be replaced with:

$$\rho_{dj} \geq \xi_h + r_{dhj} + \chi_h - M_2 \cdot (1 - V_{j,h}) \quad \forall j \in \mathcal{J}, \forall h \in \mathcal{H} \quad (4.31)$$

$$\rho_{dj} \leq \xi_h + r_{dhj} + \chi_h + M_2 \cdot (1 - V_{j,h}) \quad \forall j \in \mathcal{J}, \forall h \in \mathcal{H} \quad (4.32)$$

where the variable for the duration has been included instead of the parameter.

The scheduling MILP problem, considering the gathering of the carts and integrating the control parameters of the batch units, can be solved by:

$$\begin{aligned} & \min_{V_{j,h}, X_{i,j}, Y_{j,k}, Z_{i,k}, U_j, S_j, \rho_{dj}, \rho_{tj}, r_{dhj}} MK \\ & \text{s.t. (3.1) – (3.14)} \\ & \quad (3.19) – (3.21) \\ & \quad (4.29) – (4.32) \\ & \quad MK \geq s_j + \rho_{dj} \quad \forall j \in \mathcal{J} \end{aligned} \quad (4.33)$$

On the other hand, if the shared resource constraints are considered within this formulation, using the extension in the duration of the stage approach, the MILP problem would look like:

$$\begin{aligned} & \min_u MK \\ & \text{s.t. (3.1) – (3.14)} \\ & \quad (3.19) – (3.21) \\ & \quad (4.18) – (4.22) \\ & \quad (4.29) – (4.30) \\ & \quad \rho_{dj} \geq \xi_{h,j} + r_{dhj} + \chi_h - M_2 \cdot (1 - V_{j,h}) \\ & \quad \quad \forall j \in \mathcal{J}, \forall h \in \mathcal{H} \\ & \quad \rho_{dj} \leq \xi_{h,j} + r_{dhj} + \chi_h + M_2 \cdot (1 - V_{j,h}) \\ & \quad \quad \forall j \in \mathcal{J}, \forall h \in \mathcal{H} \\ & \quad MK \geq s_j + \rho_{dj} \quad \forall j \in \mathcal{J} \end{aligned} \quad (4.34)$$

where the decision variables are represented as the vector $\mathbf{u} \equiv (V_{j,h}, X_{i,j}, Y_{j,k}, Z_{i,k}, U_j, S_j, \rho_{d_j}, p_{t_j}, r_{d_{h,j}}, W_{j,j'}, \xi_{h,j})$, and constraints (4.31) and (4.32) have been modified to include the variable $\xi_{h,j}$.

4.4.5 Considering quality loss

This formulation can be improved by incorporating the developments explained in the subsection 4.4.2. With the same models used to compute the default relation temperature-time, several surfaces that cover the different possible extensions in the heating stage can be obtained for every reference.

The new feasible surfaces, related to every heating stage extension, are not approximated by straight lower bounds. Instead, they are approximated by moving the original feasible region through the vertical axis ("Processing time"). In this way, the duration of the maintenance stage related could be computed by:

$$r_{d_{h,j}} \geq A_{x,h} \cdot p_{t_j} + B_{x,h} - \rho_h \cdot \left(\sum_{j' \in \mathcal{J}: j' \prec j} W_{j',j} + \sum_{j'' \in \mathcal{J}: j \prec j''} W_{j,j''} \right) \quad (4.35)$$

$$\forall h \in \mathcal{H}, \forall j \in \mathcal{J}, \forall x \in \{1 \dots \kappa\}$$

$$r_{d_{h,j}} \leq A_{0,h} \cdot p_{t_j} + B_{0,h} - \rho_h \cdot \left(\sum_{j' \in \mathcal{J}: j' \prec j} W_{j',j} + \sum_{j'' \in \mathcal{J}: j \prec j''} W_{j,j''} \right) \quad (4.36)$$

$$\forall h \in \mathcal{H}, \forall j \in \mathcal{J}$$

Notice that, when the sterilization cycle is executed without overlapping with other units, the feasible operation points considered is the same that in the previous version, as the binary variables relative to the concurrent processes are canceled, i.e., $\sum_{j' \in \mathcal{J}: j' \prec j} W_{j',j} + \sum_{j'' \in \mathcal{J}: j \prec j''} W_{j,j''} = 0 \forall j \in \mathcal{J}$. Meanwhile, when there are concurrent heating phases, the feasible operation points are moved down through the vertical axis by ρ_h times the number of concurrent sterilization cycles, v.gr., when two heating phases coincide, the duration of all the points belonging to their respective regions will be reduced by ρ_h .

4.5 Results

In this section, results from implementing the previous formulations are shown. The cost function that has been defined for the different examples is minimizing the makespan, i.e., minimizing the maximum ending time.

The shared resources consumption is included in the MILP model. The system considers four different thermal treatments, with their respective steam consumption profiles, see figure 4.11. The references are ordered by total steam demand, the higher the reference number, the higher the steam flow demand. It is considered that all the references can be submitted also to the superior thermal treatments. The shorter reference has a duration of 85 minutes and the longest one of two hours.

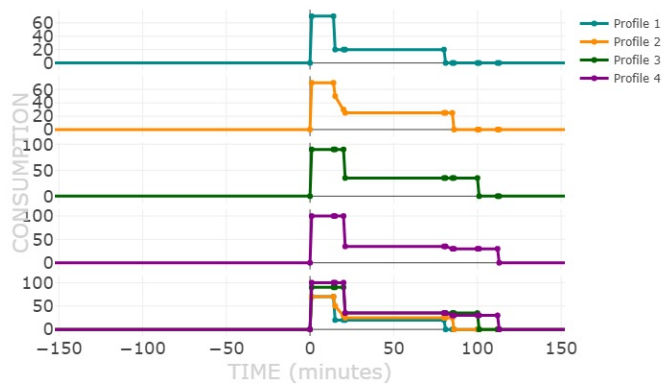


Figure 4.11: Piece-wise linear approximations to the steam consumption required by four different recipes. The horizontal axis represents the time evolution. The figure is divided in five subplots, one per reference and one with all the profile superimposed. The vertical axis represents the steam flow demand in tn/h

The steam consumption profiles are approximated by piece-wise linear functions, which values are computed in the edges of the non regular time base interval. Note that all the profiles share the same time base (\mathcal{N}). The markers in the figure show the edges of the intervals. Every important change in any steam demand profile is well defined with the selected intervals, in total there are fifteen intervals, $|\mathcal{N}| = 16$.

Not all the intervals are required for every reference. The first and last time points of the time base $(-\nu, \nu)$ are far to the left and to the right respectively.

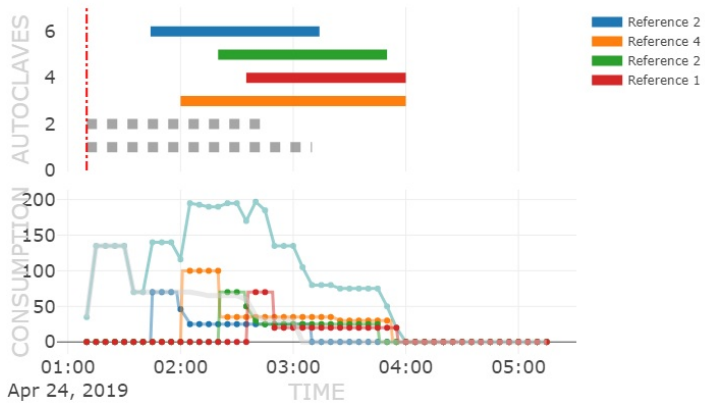
The optimization problem solved is (4.15), which limits the total steam consumption. To implement this solution in a real industrial environment, a moving horizon approach has been applied within a framework similar to the one of MPC. This methodology let the system to adapt the scheduling to the changing process conditions in real time. Some constraints only affect a short horizon (η), releasing more degrees of freedom to the future in order to being able to react. In this case the solver is executed every twenty minutes, and the solving time is five minutes.

The sterilization section considered in this example consists in six parallel autoclaves to process the orders. The carts arrive with a mean frequency about ten minutes. The carts included in the scheduling are the ones expected to arrive in less than two hours fro the current time. The robust horizon is one hour. The carts must be gathered in groups with a size between $\gamma = 2$ and $\Gamma = 8$ carts. The maximum number of slots is computed automatically in every execution of the problem, it values the geometric mean between the maximum and the minimum number of groups needed to sterilize all the carts considered. The maximum waiting time for every cart is one hundred minutes.

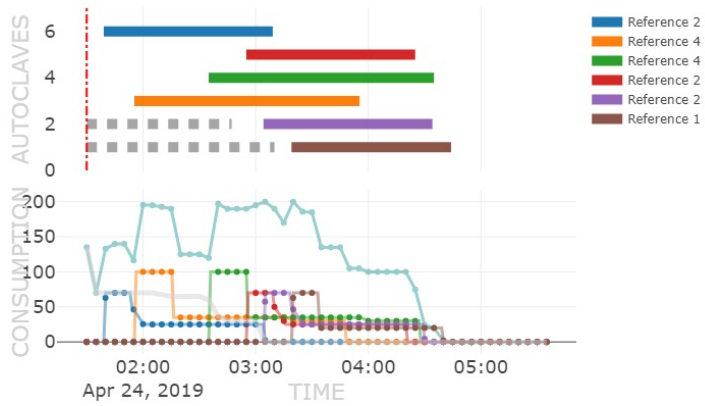
The discrete-time base has a sample step of five minutes, up to two hundred and fifty, so that $|\mathcal{M}| = \frac{250}{5} = 50$. The maximum available steam in every time sample is $\varrho = 200$ tn/h.

As the moving horizon approach has been used, four successive solutions are shown in the figure 4.12. Every sub figure is divided in two subplots: one Gantt chart and one linear chart. Both subplots share the same horizontal axis, which represents the time. A vertical red line shows current time. The Gantt chart describes the scheduling *per se*. The autoclaves are represented in the vertical axis. When one autoclave is busy, v.gr. due to a previous sterilization cycle, gray squares are plotted until it is released again. In the legend, the type of recipe selected for every sterilization cycle is shown.

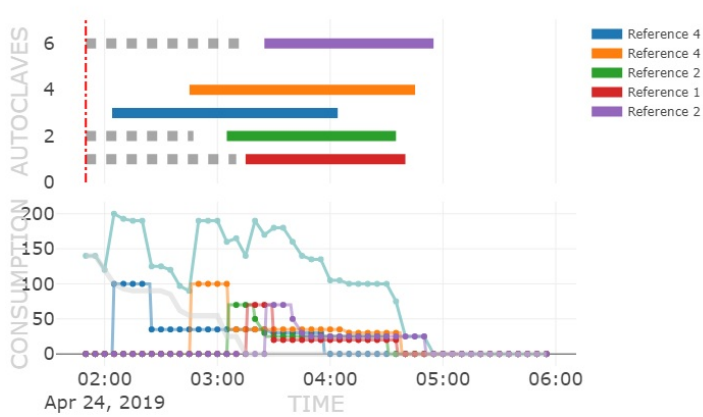
The linear chart plots the steam consumption. The light-blue line represents the whole steam demand. Notice that the upper limit of 200t/h of steam is never exceeded and the time constraints are fulfilled. The demand related to every slot



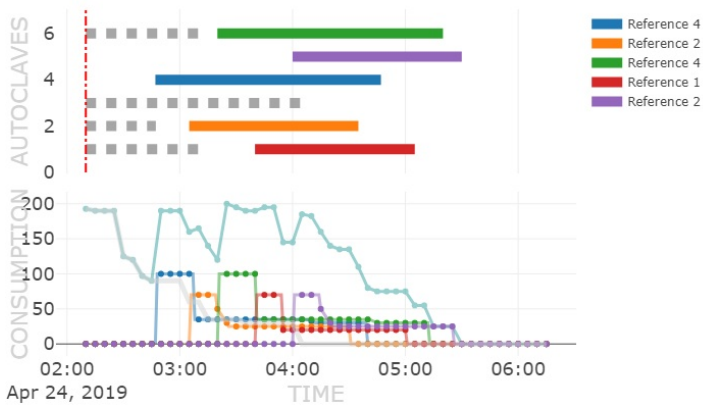
(a) Solution obtained at 01:10 am to the carts expected to arrive until 03:10 am. The carts that are expected to arrive before 02:10 are introduced in a sterilization cycle



(b) Solution obtained twenty minutes after the one in 4.12a. The carts that are expected to arrive before 02:30 are introduced in a sterilization cycle. As there are new carts considered, the previous schedule is modified



(c) Solution obtained at 1:50 am. The carts that are expected to arrive before 02:50 are introduced in a sterilization cycle. The previous first slot assigned to the sixth autoclave has already started at current time, and is plotted as gray squares



(d) Solution obtained twenty minutes after the solution in 4.12c. The carts that are expected to arrive before 03:10 are introduced in a sterilization cycle

Figure 4.12: Four successive solutions to the scheduling problem considering a bound in the steam availability. It can be seen that the whole demand never surpasses the bound of 200 tn/h.

is plotted in the same color used in the Gantt chart. The consumption related to already started sterilization cycles is plotted in a light-gray line. The steam consumption is only computed in the markers, that represent the global discrete-time base samples, to ease the visual comprehension straight line that join the markers are drawn.

4.5.1 Variable duration

In this example, the problem (4.25) is solved, which includes the effects of sterilizers overlapping instead of constraining the total steam consumption. First, the solution was tested independently, and then a moving horizon was implemented.

In this case we considered the real plant layout where the sterilization section has sixteen sterilizers. The heating stage has a minimum default duration of fifteen minutes and it is increased by five minutes for every concurrent heating stage. The carts considered are the ones expected to arrive in less than two hours since the current time, and are gathered in groups between $\gamma = 1$ and $\Gamma = 9$. Each group cannot include more than two different references, even though they share the same recipe ($\zeta = 2$).

The recipes in the same sterilization cycles cannot differ more than five minutes ($\delta = 5$).

This solution is shown in figure 4.13, where fifteen slots are schedule over the sixteen autoclaves. The horizontal axis represents the time, and the autoclaves are represented in the vertical one. Every sterilization cycle appears divided in two parts, the first one represents the heating stage, and the second one is the maintenance and the cooling stage aggregated. The color of the rectangles represent the type of sterilization cycle. In this case there are four different recipes.

The optimizer tries to distribute the heating stage as much as possible to prevent the interference. This optimization problem is solved in less than a minute.

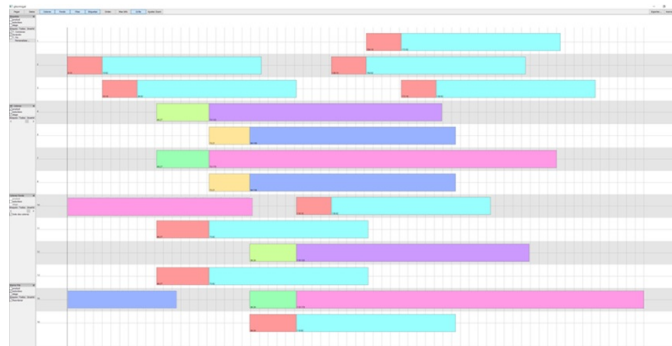
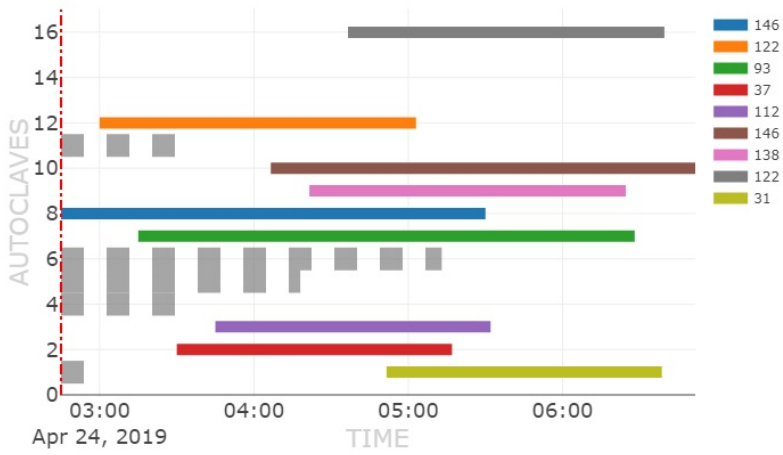


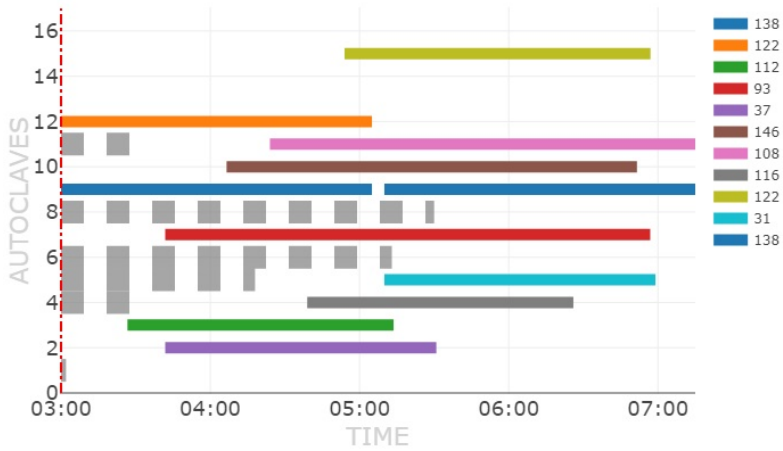
Figure 4.13: Gantt chart showing the schedule of several sterilization cycles considering the extension in the heating stages duration due to concurrent steam consumption

When the moving horizon approach is implemented, the sealing lines are added to the system and therefore the assignment constraints between carts that have left one sealing line and some forbidden autoclaves. There are twelve different sealing lines, every autoclave can receive carts from the closest five sealing lines. Seventeen different recipes are included in the problem, even though the number of different references is much bigger.

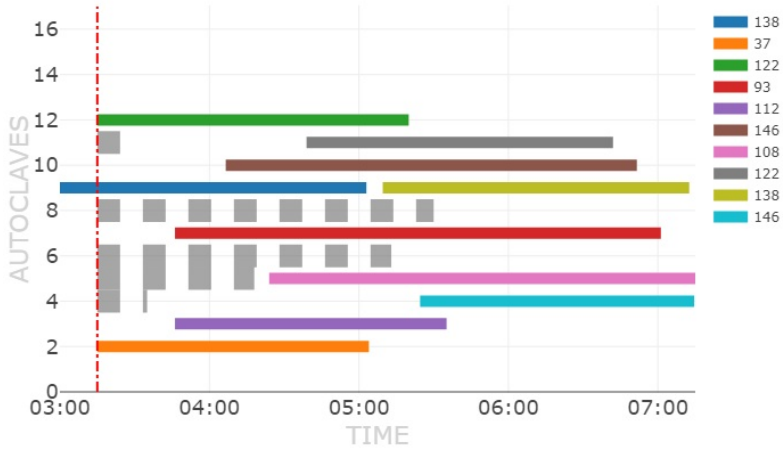
The problem is solved every fifteen minutes, see figure 4.14. The Gantt chart represents as gray squares when one autoclave is busy. The current time is shown with a vertical red line. It is considered that the forecast of carts arrivals made by the ERP system will have fails, and some error have been introduced on purpose. It can be seen that the sterilization cycle assigned to the autoclave twelve is postponed several times, probably because it has assigned one cart that haven't arrive yet. This sterilization cycle will not wait longer than the maximum waiting time for every cart already introduced in the sterilizer.



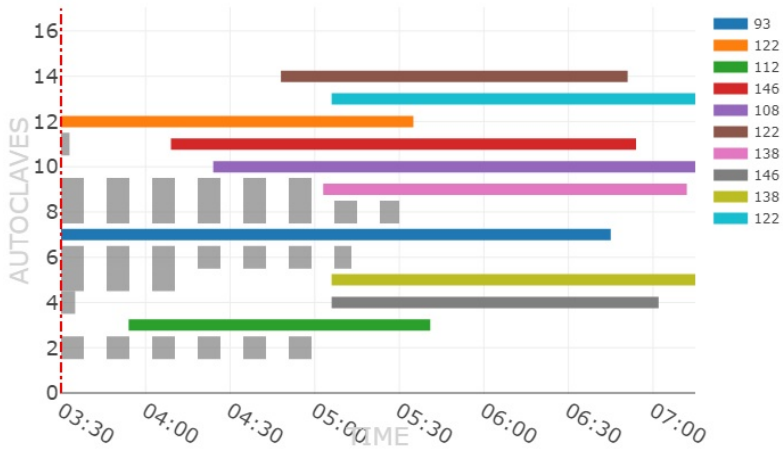
(a) Solution obtained at 02:45 am. The optimizer gather the carts in nine different sterilization cycles



(b) Solution obtained at 03:00 am, fifteen minutes after the solution in 4.14a



(c) Solution to the schedule at 03:15 am

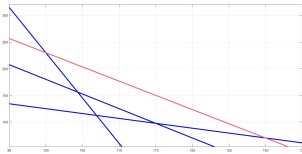


(d) Solution at 03:30 am

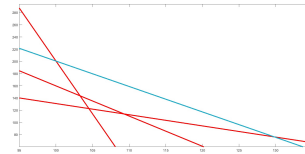
Figure 4.14: Four successive solutions to the scheduling problem considering an increase in the duration of the heating stages when concurrent steam consumption occurs

4.5.2 Integrating control parameters

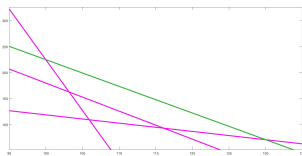
In this subsection some solutions to problem (4.33), where the operation recipe of the sterilizers is modified, are shown. The optimizer gathers two hundred carts and schedule the groups over sixteen autoclaves. Five different recipes are considered.



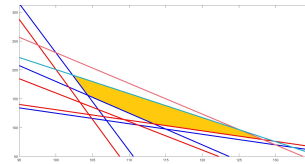
(a) Blue lines represent the lower bounds to approximate the relation temperature set point - maintenance stage duration for the first reference. The magenta line is the upper bound



(b) Red lines represent the lower bounds to approximate the relation temperature-duration. Cyan line represent the upper bound



(c) Third example for the profile approximation with pink lines for the lower bounds, and a green line for the upper bound



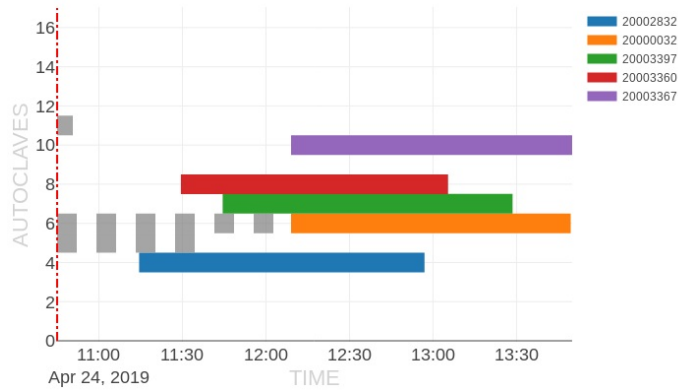
(d) Representation of the feasible region (yellow) of operation points for a sterilization cycle that combines the references shown in 4.15a and 4.15b. The feasible region is the area over all blue and red lines and under magenta and cyan lines

Figure 4.15: Straight lines examples to approximate the feasible operation point region. In the horizontal axis the temperature set point, and in the vertical axis the duration required for the maintenance stage

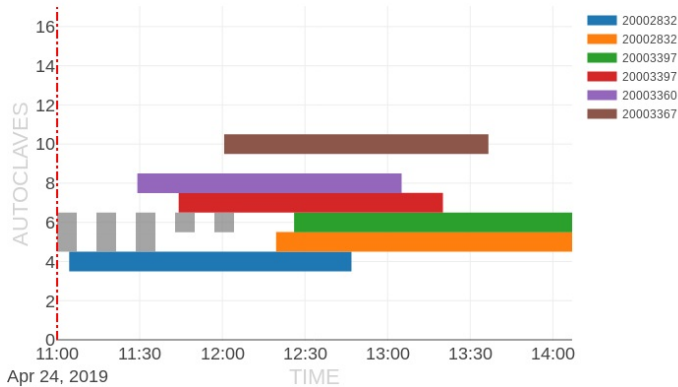
The feasible region for every reference is approximated by three straight lines, in addition to the upper bound. The figure 4.15 shows the straight lines used to approximate three of the five references, and the feasible region for one sterilization cycle that combines carts from the first and the second, 4.15d. The lower bounds

are computed to approximate the non-linear relation temperature-time. The upper bound is computed to prevent the operation point chosen to select temperatures under 100 °C or over 130 °C.

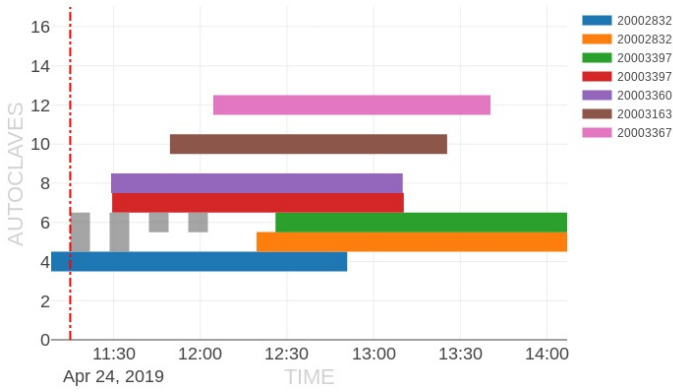
The optimizer finds a solution in less than a minute. Some solutions applying the moving horizon approach can be seen in figure 4.16. As all the carts can



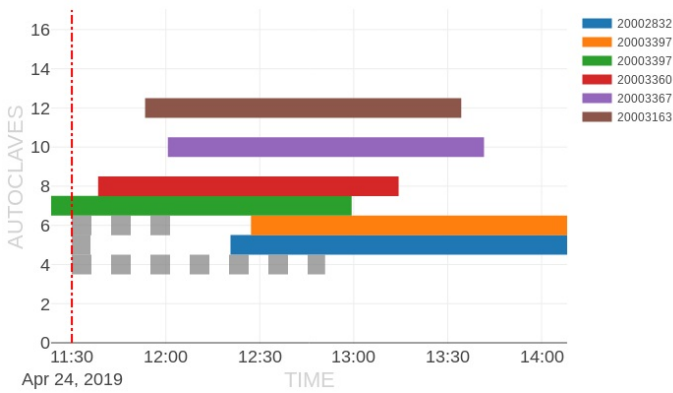
(a) Solution obtained at 02:45 am. The optimizer gather the carts in nine different sterilization cycles



(b) Solution obtained at 03:00 am, fifteen minutes after the solution in 4.16a



(c) Solution to the schedule at 03:15 am



(d) Solution at 03:30 am

Figure 4.16: Four successive solutions to the scheduling problem considering an increase in the duration of the heating stages when concurrent steam consumption occurs

be gathered in the same sterilization cycle, the optimizer reduces the number of sterilizers needed.

4.5.3 Uncertainty in the cart supply

In continuous industrial production exists a vast number of possible causes for uncertainties in supply lines behavior. In the canning industry case study, the different properties of the filling liquids and the tuna parts chosen modify the sealing lines speed. In addition, the release of filled carts and the arrangement of the empty carts are performed by human operators, which increases the time uncertainty on the supply to the sterilization section.

On the other hand, the work orders set by the ERP system can be modified by the operators. This alterations might be motivated by external factors, such as last minute orders, or by internal ones, v.gr., a mistake when select in the products to thaw.

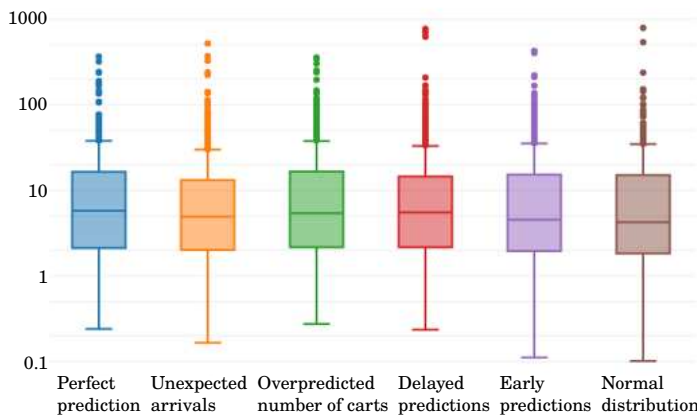


Figure 4.17: Solution times for close-loop schedule of two days with different uncertainty distributions in the arrival of carts. In the vertical axis the time spent in seconds. In the horizontal axis the different uncertainty distributions used

In summary, there is a huge uncertainty in the arrival of carts to the sterilization section, in the arrival time instants and in the reference included in every cart. This unknown possibilities have been taken into account in the real-time scheduling tool by performing online scheduling (Gupta, Maravelias, and Wassick, 2016); nevertheless, how to tackle this uncertainties in discrete fluxes is out of scope of this thesis

and is not treated here. For more information about how to treat uncertainty, readers can find details in Li and Ierapetritou (2008) and Gupta and Maravelias (2019).

In figure 4.17, the time spent by the solver to obtain a solution is compared with several uncertainty distributions in the arrival of carts.

4.6 Conclusions

Novel approaches to deal with shared resources were presented. Both approaches are proved suitable to be run in real-time applications.

The control of the batch processes was integrated in the scheduling and a near closed-loop formulation was developed.

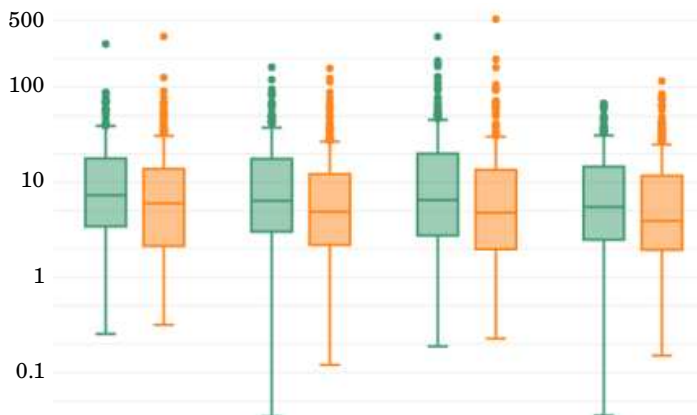


Figure 4.18: Solution times for four different runs of two days close-loop schedule. In the vertical axis the time spent is represented in seconds. Gurobi 8.1 (green) is compared with CPLEX 12.9 (orange)

One formulation to tackle a maximum allowable concurrent consumption, with a variable profile over time using continuous-slot. And one formulation to tackle variable duration of the tasks to schedule.

The final scheduling optimization tool has been prepared to be integrated in the MES system. An interface to read data from the plant database has been implemented. Therefore, the optimization problem will be updated in real-time to the actual situation of the sterilization section. See figure 4.18, it can be seen that the average time spent is less than 10 s.

Main contributions and conclusions

There is a wide variety of problems in the industrial sector where continuous and batch process are connected. The main issue in these cases consists in the assignment of tasks to units, and the sequencing of those tasks.

However, even though these problems share the base problem, the broad range of specific issues inside every industrial environment, makes nearly impossible finding a common formulation. Every situation has to be studied in advanced to choose the optimal methodology for every circumstance.

Nevertheless, we can distinguish certain types of problems with common traits that can be approached in a similar way. In this thesis we have been dealing with some of these problems. Some have not been treated extensively in the literature and the solutions offered room for improvement. With this thesis we tried to provide novel and efficient formulations for them contributing to the development of the field of Process Systems Engineering and the implementation of principles and aims of Industry4.0.

In particular, we focused the attention to the integration of maintenance of equipment in RTO problems and in the real-time scheduling of combined continuous-batch processes, with reference to two relevant industrial case studies: one corresponding to a viscose-fiber plant and other to a tuna canning factory. Even though we have implemented the solutions to particular industrial cases, the developments

here described can be easily extended to similar scenarios. The second chapter deals with scheduling considering degradation in the efficiency of some units. The third chapter deals with the scheduling of similar tasks that can be processes sharing the batch units. Finally, the fourth chapter deals with shared resources between concurrent processes and present two approaches that might be applied depending on the type of particular problem.

With regard to the aims of the thesis mentioned in the introduction, the main contributions have been:

- We have studied the problems of incorporating maintenance due to fouling in continuous operating plants in RTO, modeling an evaporation plant and developing a procedure for deciding the optimal operation. A self-optimizing version of the solution has been implemented in the factory providing good and consistent results
- We have studied the problems of optimally scheduling the operation of a set of continuous plants, incorporating maintenance operations, providing efficient scheduling formulations able to deal with the complexity of an industrial case
- We have studied the problem of incorporating an explicit consideration of the effect of uncertain variables into the scheduling formulations, generating two-stage optimization ones
- We have studied the problems of operation of plants with continuous lines followed by parallel batch units, proposing the use of real-time scheduling, developing formulations and codes able to deal with an industrial scale problem
- We have developed efficient scheduling formulations in continuous time dealing with limited shared resources
- We have elaborated a novel formulation to tackle variable batch times that depends on shared resource consumption synchronization
- We have proposed solutions of that problems that integrates the operation of the scheduling and control of batch units, improving the overall performance of the system

- We have validated the methods and algorithms with industrial data and we have studied its industrial implementation.

There are other points that may be worth mentioning at this stage:

Current simulation and modeling tools offers a high potential to improve the knowledge of the systems. In these simulators, undesirable scenarios can be performed and study how the system reacts. They can be used to train the operators. Different working policies can be compared without altering the production schedule.

In the current global situation, industries try to reduce environmental footprint while keeping a competitive position in market. Management departments are searching for efficiency in the use of resources. Mathematical programming techniques, which assure obtaining an optimal operation, rise as the best options for the scheduling problems.

To obtain the optimal schedule, the optimization problem has to include, not only the supervision layer details, but some from the control or planning layers. Depending on the system, it will be suitable to implement the global optimization problem as a MINLP problem. However, in most cases, some details have to be linearized to be included in a MILP problem that will be solved faster.

In the last years, the algorithms efficiency have been improved either for NLP problem as to MILP ones. The solvers are able to be used for real-time tools. However, the solving time seek to solve one real-time optimization problem depends on the application that it will have, but human factors have to be taken into account which may exclude some solutions, v.gr. for the optimization of a week work, one optimization that take hours could seem admissible, but not for someone waiting for the answer.

A global scheduler will provide the operators with a powerful tool to control the production quality or load.

When the final product is intended to be implemented in a real industrial site, the tool used to model the system has a significant weight. Industrial applications have to be robust and easy to use. Industrial sector tends to move their management systems to the cloud. High level multipurpose languages offers a powerful tool to interface with these on-line environments.

As a result of the work performed in the thesis, several papers were presented in international conferences and published in indexed peer reviewed journals:

- ◇ Palacín, Carlos G., Méndez, Carlos A., and de Prada, Cesar (May 2019). "Slots Startup Synchronization with Shared Resources Dependency." In: *Chemical Engineering Transactions* 74, pp. 1321–1326.
- ◇ Palacín, Carlos G., José Luis Pitarch, Christian Jasch, Carlos A. Méndez, and Cesar de Prada (Feb. 2018). "Robust integrated production-maintenance scheduling for an evaporation network." In: *Computers & Chemical Engineering* 110, pp. 140–151. issn: 0098-1354.
- ◇ Pitarch, José Luis, Carlos G. Palacín, Cesar de Prada, Bernhard Voglauer, and Gerhard Seyfriedsberger (2017). "Optimisation of the resource efficiency in an industrial evaporation system." In: *Journal of Process Control* 56, pp. 1–12. issn: 0959-1524.
- ◇ Palacín, Carlos G., Carlos Vilas, Antonio A. Alonso, José Luis Pitarch, and Cesar de Prada (2020). "Integration of sterilization dynamics with shared resources in the production scheduling." In: . (*under review*)
- Palacín, Carlos G. and Cesar de Prada (Jan. 2019). "Optimal Coordination Of Batch Processes with Shared Resources." In: *IFAC-PapersOnLine*. 12th IFAC Symposium on Dynamics and Control of Process Systems, including Biosystems DYCOPS 2019 52(1), pp. 826–831. issn: 2405-8963.
- Palacín, Carlos G., Pablo Riquelme, and Cesar de Prada (2019). "Scheduling óptimo de procesos Batch de duración interdependiente." spa. In: Universidade da Coruña, Servizo de Publicacións, pp. 560–567. isbn: 978-84-9749-716-9.

-
- Palacín, Carlos G., Cesar de Prada, and José Luis Pitarch (2018). “Ayuda al operario en la distribución óptima de carga entre equipos equivalentes.” In: Universidad de Extremadura.
 - Palacin, Carlos G., José Luis Pitarch, Cesar de Prada, and Carlos A. Méndez (2017). “Robust multi-objective scheduling in an evaporation network.” In: *2017 25th Mediterranean Conference on Control and Automation (MED)*. IEEE, pp. 666–671.
 - Palacín, Carlos G., José Luis Pitarch, Cesar de Prada, and Carlos A. Méndez (2017). “An Efficient Way to Tackle Uncertainty in the Scheduling of a Continuous Evaporation System.” In: *27th European Symposium on Computer Aided Process Engineering*. Ed. by Antonio Espuña, Moisès Graells, and Luis Puigjaner. Vol. 40. Computer Aided Chemical Engineering. Elsevier, pp. 1411–1416.
 - Palacín, Carlos G., José Luis Pitarch, Gloria Gutiérrez, and Cesar de Prada (2017). “Aproximación de modelos algebraicos mediante ALAMO y ECOSIM-PRO.” in: Servicio de Publicaciones de la Universidad de Oviedo.
 - Pitarch, José Luis, Carlos G. Palacín, Alejandro Merino, and Cesar de Prada (2017). “Optimal Operation of an Evaporation Process.” In: *Modeling, Simulation and Optimization of Complex Processes HPSC 2015*. Ed. by Hans Georg Bock, Hoang Xuan Phu, Rolf Rannacher, and Johannes P. Schlöder. Springer International Publishing: Cham, pp. 189–203. isbn: 978-3-319-67168-0.
 - Pitarch, José Luis, Carlos G. Palacín, Cesar de Prada, Marc Kalliski, and Christian Jasch (2017). “Online Decision Support for an Evaporation Network.” In: Servicio de Publicaciones, pp. 575–581.
 - Palacín, Carlos G., José Luis Pitarch, and Cesar de Prada (2016). “Planificación y distribución óptima de cargas en un sistema de evaporadores industriales.” In: *XXXVII Jornadas de Automática*. Comité Español de Automática de la IFAC(CEA-IFAC), pp. 801–806. isbn: 978-84-617-4298-1.

- Palacín, Carlos G., José Luis Pitarch, and Cesar de Prada (2015b). "Optimización en tiempo real de sistemas estacionarios en EcosimPro." In: *XXXVI Jornadas de Automática*. Comité Español de Automática de la IFAC(CEA-IFAC), pp. 502–508. isbn: 978-84-15914-12-9.
- Palacín, Carlos G., José Luis Pitarch, and Cesar de Prada (2015a). "Efficient modelling and real-time optimisation of stationary systems: Application to an evaporation process." In: *Proceedings of SIMUL*, pp. 6–11.

In addition, the developments of this thesis have been applied in different case studies belonging to both European projects: *Real-time Monitoring and Optimization of Resource Efficiency in Integrated Processing Plants (MORE)* and *Improved energy and resource efficiency by better coordination of production in the process industries (CoPro)*. Both intended for a reduction in the resource consumption by the European process industry, reducing the climate footprint, and led by Dr. Sebastian Engell, full professor.

Future work In this thesis, the problem of integrating some control parameters in the schedule has been solved linearizing the non-linear optimal control problem. However, this is neither always possible, nor the real optimal solution. In order to incorporate non-linearities to the schedule, while keeping a reasonable solving time, advance techniques have to be applied.

For example, decomposition methods, where the duality problem is exploited to reduce the complexity of some constraints by dividing the initial problem into two optimization problems. This path can be followed to, for example, incorporate the temperature set point optimization problem in the scheduling.

On the other hand, including uncertainty in discrete decisions is a current topic, with a high importance to the upcoming Industry4.0. Robustness studies of the solutions obtained against changes in the production demands or the market, is highly valuable. There are plenty studies currently to include statistics techniques in the scheduling optimization problems.

Bibliography

- Albers, Donald J. and Constance Reid (1986). "An Interview with George B. Dantzig: The Father of Linear Programming." In: *The College Mathematics Journal* 17(4), pp. 292–314 (cited on page 5).
- Albuquerque, João S. and Lorenz T. Biegler (1996). "Data reconciliation and gross-error detection for dynamic systems." In: *AIChE Journal* 42(10), pp. 2841–2856 (cited on page 1).
- Alonso, Antonio A., Ana Arias-Méndez, Eva Balsa-Canto, Míriam R. García, Juan I. Molina, Carlos Vilas, and Marcos Villafín (Aug. 2013). "Real time optimization for quality control of batch thermal sterilization of prepackaged foods." en. In: *Food Control* 32(2), pp. 392–403. issn: 0956-7135 (cited on page 58).
- Alonso, Antonio A., Julio R. Banga, and Ricardo Perez-Martin (Apr. 1997). "A complete dynamic model for the thermal processing of bioproducts in batch units and its application to controller design." en. In: *Chemical Engineering Science* 52(8), pp. 1307–1322. issn: 0009-2509 (cited on page 58).
- Åström, Karl Johan, Tore Hägglund, and Karl J Astrom (2006). *Advanced PID control*. Vol. 461. ISA-The Instrumentation, Systems, and Automation Society Research Triangle . . . (cited on page 2).
- Baker, Kenneth R. and Dan Trietsch (2009). *Principles of Sequencing and Scheduling*. Wiley Publishing. isbn: 0470391650 (cited on page 5).

- Balas, Egon (1975). "Disjunctive programming: cutting planes from logical conditions." In: *Nonlinear Programming 2*. Elsevier, pp. 279–312 (cited on page 12).
- Balas, Egon (2018). *Disjunctive Programming*. en. Springer International Publishing: Cham. isbn: 978-3-030-00147-6 (cited on page 73).
- Baldea, Michael and Iiro Harjunkoski (2014). "Integrated production scheduling and process control: A systematic review." In: *Computers & Chemical Engineering* 71, pp. 377–390. issn: 0098-1354 (cited on page 13).
- Basán, Natalia P., Ignacio E. Grossmann, Ajit Gopalakrishnan, Irene Lotero, and Carlos A. Méndez (2017). "Optimal scheduling for power-intensive processes under time-sensitive electricity prices." In: *27th European Symposium on Computer Aided Process Engineering*. Ed. by Antonio Espuña, Moisès Graells, and Luis Puigjaner. Vol. 40. Computer Aided Chemical Engineering. Elsevier, pp. 1423–1428 (cited on page 38).
- Behrmann, Gerd, Kim G Larsen, and Jacob I Rasmussen (2005). "Optimal scheduling using priced timed automata." In: *ACM SIGMETRICS Performance Evaluation Review* 32(4), pp. 34–40 (cited on page 4).
- Bell, Bradley M. and James V. Burke (2008). "Algorithmic Differentiation of Implicit Functions and Optimal Values." In: *Advances in Automatic Differentiation*. Ed. by Christian H. Bischof, H. Martin Bücker, Paul Hovland, Uwe Naumann, and Jean Utke. Springer Berlin Heidelberg: Berlin, Heidelberg, pp. 67–77. isbn: 978-3-540-68942-3 (cited on page 26).
- Bezanson, Jeff, Alan Edelman, Stefan Karpinski, and Viral Shah (July 2015). *Julia: A Fresh Approach to Numerical Computing* (cited on page 78).
- Bezanson, Jeff, Stefan Karpinski, Viral B. Shah, and Alan Edelman (2012). *Julia: A Fast Dynamic Language for Technical Computing* (cited on page 78).

- Biegler, Lorenz T. (June 2018). "Advanced optimization strategies for integrated dynamic process operations." In: *Computers & Chemical Engineering*. FOCAPO/CPC 2017 114, pp. 3–13. issn: 0098-1354 (cited on page 26).
- Birand, B., M. Chudnovsky, B. Ries, P. Seymour, G. Zussman, and Y. Zwols (2012). "Analyzing the Performance of Greedy Maximal Scheduling via Local Pooling and Graph Theory." In: *IEEE/ACM Transactions on Networking* 20(1), pp. 163–176. issn: 1558-2566 (cited on page 4).
- Bisschop, Johannes and Alexander Meeraus (1982). "On the development of a general algebraic modeling system in a strategic planning environment." In: *Applications*. Ed. by Jean-Louis Goffin and Jean-Marc Rousseau. Springer Berlin Heidelberg: Berlin, Heidelberg, pp. 1–29. isbn: 978-3-642-00852-8 (cited on page 78).
- Boyd, Stephen and Lieven Vandenbergh (2004). *Convex optimization*. Cambridge university press (cited on page 2).
- Brucker, Peter (2007). *Scheduling algorithms*. Vol. 3. Springer (cited on page 4).
- Burns, Roland (2001). *Advanced control engineering*. Elsevier (cited on page 2).
- Camacho, Eduardo F and Carlos Bordons Alba (2013). *Model predictive control*. Springer Science & Business Media (cited on page 16).
- Castro, Pedro M., Ignacio E. Grossmann, and Qi Zhang (2018). "Expanding scope and computational challenges in process scheduling." In: *Computers & Chemical Engineering* 114. FOCAPO/CPC 2017, pp. 14–42. issn: 0098-1354 (cited on page 13).
- Chen, JJJ (1987). "Comments on improvements on a replacement for the logarithmic mean." In: *Chemical Engineering Science* 42(10), pp. 2488–2489 (cited on page 29).

- Constantinescu, E. M., V. M. Zavala, M. Rocklin, S. Lee, and M. Anitescu (Feb. 2011). "A Computational Framework for Uncertainty Quantification and Stochastic Optimization in Unit Commitment With Wind Power Generation." In: *IEEE Transactions on Power Systems* 26(1), pp. 431–441. issn: 1558-0679 (cited on page 27).
- Dakin, Robert J (1965). "A tree-search algorithm for mixed integer programming problems." In: *The computer journal* 8(3), pp. 250–255 (cited on page 12).
- Dantzig, George B (1951). "Maximization of a linear function of variables subject to linear inequalities." In: *Activity analysis of production and allocation* 13, pp. 339–347 (cited on page 5).
- Dantzig, George B. and Philip Wolfe (1961). "The decomposition algorithm for linear programs." In: *Econometrica: Journal of the Econometric Society*, pp. 767–778 (cited on page 12).
- Dantzig, George Bernard (1983). "Reminiscences about the origins of linear programming." In: *Mathematical Programming The State of the Art*. Springer, pp. 78–86 (cited on page 5).
- Dunning, Iain, Joey Huchette, and Miles Lubin (2017). "JuMP: A Modeling Language for Mathematical Optimization." In: *SIAM Review* 59(2), pp. 295–320 (cited on page 78).
- Floudas, Christodoulos A (1995). *Nonlinear and mixed-integer optimization: fundamentals and applications*. Oxford University Press (cited on page 11).
- Gilchrist, Alasdair (2016). *Industry 4.0: the industrial internet of things*. Springer (cited on page 1).
- Gomory, Ralph (1960). *An algorithm for the mixed integer problem*. Tech. rep. RAND CORP SANTA MONICA CA (cited on page 12).

Green, Meghan, Jean-Denis Caron, and Van Fong (2016). "On the use of simulated annealing to optimize maintenance schedules for naval platforms." In: (cited on page 4).

Griewank, A. (Jan. 1988). "On automatic differentiation." In: (cited on page 26).

Grossmann, Ignacio E. (2012). "Advances in mathematical programming models for enterprise-wide optimization." In: *Computers & Chemical Engineering* 47. FOCAPO 2012, pp. 2–18. issn: 0098-1354 (cited on page 3).

Grossmann, Ignacio E., Robert M. Apap, Bruno A. Calfa, Pablo García-Herreros, and Qi Zhang (2016a). "Recent advances in mathematical programming techniques for the optimization of process systems under uncertainty." In: *Computers & Chemical Engineering* 91. 12th International Symposium on Process Systems Engineering & 25th European Symposium of Computer Aided Process Engineering (PSE-2015/ESCAPE-25), 31 May - 4 June 2015, Copenhagen, Denmark, pp. 3–14. issn: 0098-1354 (cited on page 13).

Grossmann, Ignacio E., Robert M. Apap, Bruno A. Calfa, Pablo García-Herreros, and Qi Zhang (Aug. 2016b). "Recent advances in mathematical programming techniques for the optimization of process systems under uncertainty." en. In: *Computers & Chemical Engineering*. 12th International Symposium on Process Systems Engineering & 25th European Symposium of Computer Aided Process Engineering (PSE-2015/ESCAPE-25), 31 May - 4 June 2015, Copenhagen, Denmark 91, pp. 3–14. issn: 0098-1354 (cited on page 27).

Gupta, Dhruv and Christos T. Maravelias (Oct. 2019). "On the design of online production scheduling algorithms." en. In: *Computers & Chemical Engineering* 129, p. 106517. issn: 0098-1354 (cited on pages 68, 119).

Gupta, Dhruv, Christos T. Maravelias, and John M. Wassick (2016). "From rescheduling to online scheduling." In: *Chemical Engineering Research and De-*

sign 116. Process Systems Engineering - A Celebration in Professor Roger Sargent's 90th Year, pp. 83–97. issn: 0263-8762 (cited on page 118).

Gurobi Optimizer Reference Manual (2020). url: <https://www.gurobi.com/documentation/9.0/refman/index.html> (visited on 01/21/2020) (cited on page 78).

Harjunkoski, Iiro, Christos T. Maravelias, Peter Bongers, Pedro M. Castro, Sebastian Engell, Ignacio E. Grossmann, John Hooker, Carlos Méndez, Guido Sand, and John Wassick (Mar. 2014). "Scope for industrial applications of production scheduling models and solution methods." In: *Computers & Chemical Engineering* 62, pp. 161–193. issn: 0098-1354 (cited on page 67).

Held, Michael and Richard M Karp (1970). "The traveling-salesman problem and minimum spanning trees." In: *Operations Research* 18(6), pp. 1138–1162 (cited on page 12).

Hentenryck, P. van, L. Michel, L. Perron, and J. -C. Régin (1999). "Constraint Programming in OPL." In: *Principles and Practice of Declarative Programming*. Ed. by Gopalan Nadathur. Springer Berlin Heidelberg: Berlin, Heidelberg, pp. 98–116. isbn: 978-3-540-48164-5 (cited on page 77).

Heyman, Daniel P and Matthew J Sobel (2004). *Stochastic models in operations research: stochastic optimization*. Vol. 2. Courier Corporation (cited on page 27).

Huber, P.J. (2014). "Robust Statistics." In: *International Encyclopedia of Statistical Science*. Ed. by Miodrag Lovric. Springer Berlin Heidelberg, pp. 1248–1251. isbn: 978-3-642-04897-5 (cited on page 25).

ILOG CPLEX Manuals (2020). url: <https://www.ibm.com/support/pages/ilog-cplex-manuals> (visited on 01/21/2020) (cited on page 44).

-
- Industria Conectada 4.0* (2019). url: <https://www.industriaconectada40.gob.es/Paginas/Index.aspx> (visited on 09/18/2019) (cited on page 1).
- Kall, Peter, Stein W Wallace, and Peter Kall (1994). *Stochastic programming*. Springer (cited on page 46).
- Kondili, E., C. C. Pantelides, and R. W. H. Sargent (Feb. 1993). "A general algorithm for short-term scheduling of batch operations—I. MILP formulation." In: *Computers & Chemical Engineering*. An International Journal of Computer Applications in Chemical Engineering 17(2), pp. 211–227. issn: 0098-1354 (cited on pages 11, 56).
- Krämer, Stefan and Sebastian Engell (2018). *Resource efficiency of processing plants: monitoring and improvement*. John Wiley & Sons (cited on page 25).
- Leibman, M.J., T.F. Edgar, and L.S. Lasdon (1992). "Efficient data reconciliation and estimation for dynamic processes using nonlinear programming techniques." In: *Computers & Chemical Engineering* 16(10–11), pp. 963–986. issn: 0098-1354 (cited on page 24).
- Lenzing AG (2018). *Fiber production*. url: <https://www.lenzing.com/en/sustainability/production/fiber-production/> (visited on 11/26/2018) (cited on page 18).
- Li, Zukui and Marianthi Ierapetritou (2008). "Process scheduling under uncertainty: Review and challenges." In: *Computers & Chemical Engineering* 32(4). Festschrift devoted to Rex Reklaitis on his 65th Birthday, pp. 715–727. issn: 0098-1354 (cited on page 119).
- Maravelias, Christos T. (2012). "General framework and modeling approach classification for chemical production scheduling." In: *AIChE Journal* 58(6), pp. 1812–1828 (cited on page 56).

- McAfee, Andrew, Erik Brynjolfsson, Thomas H Davenport, DJ Patil, and Dominic Barton (2012). "Big data: the management revolution." In: *Harvard business review* 90(10), pp. 60–68 (cited on page 1).
- Méndez, Carlos A., Jaime Cerdá, Ignacio E. Grossmann, Iiro Harjunkoski, and Marco Fahl (May 2006). "State-of-the-art review of optimization methods for short-term scheduling of batch processes." In: *Computers & Chemical Engineering* 30(6), pp. 913–946. issn: 0098-1354 (cited on pages 5, 56, 75).
- Muth, John F and Gerald Luther Thompson (1963). *Industrial scheduling*. Prentice-Hall (cited on page 5).
- Narasimhan, Shankar and Cornelius Jordache (1999). *Data reconciliation and gross error detection: An intelligent use of process data*. Elsevier (cited on page 1).
- Palacín, Carlos G., Méndez, Carlos A., and de Prada, Cesar (May 2019). "Slots Startup Synchronization with Shared Resources Dependency." In: *Chemical Engineering Transactions* 74, pp. 1321–1326 (cited on pages 75, 124).
- Palacín, Carlos G., José Luis Pitarch, Gloria Gutiérrez, and Cesar de Prada (2017). "Aproximación de modelos algebraicos mediante ALAMO y ECOSIMPRO." In: Servicio de Publicaciones de la Universidad de Oviedo (cited on page 125).
- Palacín, Carlos G., José Luis Pitarch, Christian Jasch, Carlos A. Méndez, and Cesar de Prada (Feb. 2018). "Robust integrated production-maintenance scheduling for an evaporation network." In: *Computers & Chemical Engineering* 110, pp. 140–151. issn: 0098-1354 (cited on pages 27, 124).
- Palacín, Carlos G., José Luis Pitarch, and Cesar de Prada (2015a). "Efficient modelling and real-time optimisation of stationary systems: Application to an evaporation process." In: *Proceedings of SIMUL*, pp. 6–11 (cited on page 126).

-
- Palacín, Carlos G., José Luis Pitarch, and Cesar de Prada (2015b). "Optimización en tiempo real de sistemas estacionarios en EcosimPro." In: *XXXVI Jornadas de Automática*. Comité Español de Automática de la IFAC(CEA-IFAC), pp. 502–508. isbn: 978-84-15914-12-9 (cited on page 126).
- Palacín, Carlos G., José Luis Pitarch, and Cesar de Prada (2016). "Planificación y distribución óptima de cargas en un sistema de evaporadores industriales." In: *XXXVII Jornadas de Automática*. Comité Español de Automática de la IFAC(CEA-IFAC), pp. 801–806. isbn: 978-84-617-4298-1 (cited on page 125).
- Palacin, Carlos G., José Luis Pitarch, Cesar de Prada, and Carlos A. Méndez (2017). "Robust multi-objective scheduling in an evaporation network." In: *2017 25th Mediterranean Conference on Control and Automation (MED)*. IEEE, pp. 666–671 (cited on page 125).
- Palacín, Carlos G., José Luis Pitarch, Cesar de Prada, and Carlos A. Méndez (2017). "An Efficient Way to Tackle Uncertainty in the Scheduling of a Continuous Evaporation System." In: *27th European Symposium on Computer Aided Process Engineering*. Ed. by Antonio Espuña, Moisès Graells, and Luis Puigjaner. Vol. 40. Computer Aided Chemical Engineering. Elsevier, pp. 1411–1416 (cited on page 125).
- Palacín, Carlos G., Cesar de Prada, and José Luis Pitarch (2018). "Ayuda al operario en la distribución óptima de carga entre equipos equivalentes." In: Universidad de Extremadura (cited on page 125).
- Palacín, Carlos G. and Cesar de Prada (Jan. 2019). "Optimal Coordination Of Batch Processes with Shared Resources." In: *IFAC-PapersOnLine*. 12th IFAC Symposium on Dynamics and Control of Process Systems, including Biosystems DYCOPS 2019 52(1), pp. 826–831. issn: 2405-8963 (cited on page 124).
- Palacín, Carlos G., Pablo Riquelme, and Cesar de Prada (2019). "Scheduling óptimo de procesos Batch de duración interdependiente." spa. In: Universidade da

- Coruña, Servizo de Publicacións, pp. 560–567. isbn: 978-84-9749-716-9 (cited on page 124).
- Palacín, Carlos G., Carlos Vilas, Antonio A. Alonso, José Luis Pitarch, and Cesar de Prada (2020). “Integration of sterilization dynamics with shared resources in the production scheduling.” In: (cited on page 124).
- Pantelides, Constantinos C (1994). “Unified frameworks for optimal process planning and scheduling.” In: *Proceedings of the Second conference on FOCAPO*. Cache Publications New York, pp. 253–274 (cited on pages 11, 56).
- Pegden, Claude Dennis (2017). *Deliver on your promise: How simulation-based scheduling will change your business*. Simio LLC (cited on page 55).
- Pinedo, Michael L. (2008). *Scheduling: Theory, Algorithms, and Systems*. 3rd. Springer (cited on page 4).
- Pitarch, José Luis, Carlos G. Palacín, Alejandro Merino, and Cesar de Prada (2017). “Optimal Operation of an Evaporation Process.” In: *Modeling, Simulation and Optimization of Complex Processes HPSC 2015*. Ed. by Hans Georg Bock, Hoang Xuan Phu, Rolf Rannacher, and Johannes P. Schlöder. Springer International Publishing: Cham, pp. 189–203. isbn: 978-3-319-67168-0 (cited on pages 25, 125).
- Pitarch, José Luis, Carlos G. Palacín, Cesar de Prada, Marc Kalliski, and Christian Jasch (2017). “Online Decision Support for an Evaporation Network.” In: *Servicio de Publicaciones*, pp. 575–581 (cited on page 125).
- Pitarch, José Luis, Carlos G. Palacín, Cesar de Prada, Bernhard Voglauer, and Gerhard Seyfriedsberger (2017). “Optimisation of the resource efficiency in an industrial evaporation system.” In: *Journal of Process Control* 56, pp. 1–12. issn: 0959-1524 (cited on page 124).

-
- Pitarch, José Luis, Antonio Sala, and Cesar de Prada (2019). "A Systematic Grey-Box Modeling Methodology via Data Reconciliation and SOS Constrained Regression." In: *Processes* 7(3). issn: 2227-9717 (cited on page 24).
- Prada, C. de, E. Villar, and V. Alexandre (1987). "Dynamic simulation of a multiple-effect evaporation station." In: *Simulation of Control Systems*. Ed. by I. Troch, P. Kopacek, and F. Breitenecker. IFAC Symposia Series. Pergamon: Oxford, pp. 129–132. isbn: 978-0-08-034349-5 (cited on page 18).
- Prada, César de, Rogelio Mazaeda, and Smaranda P. Cristea (June 2019). "Receding horizon scheduling of processes with shared resources." In: *Computers & Chemical Engineering* 125, pp. 1–12. issn: 0098-1354 (cited on page 89).
- Prada, Cesar de and José Luis Pitarch (2018). "Real-time optimization (RTO) systems." In: *Resource Efficiency of Processing Plants: Monitoring and Improvement*, pp. 265–292 (cited on page 2).
- Prada, Cesar de, Daniel Sarabia, Gloria Gutierrez, Elena Gomez, Sergio Marmol, Mikel Sola, Carlos Pascual, and Rafael Gonzalez (Mar. 2017). "Integration of RTO and MPC in the Hydrogen Network of a Petrol Refinery." en. In: *Processes* 5(1), p. 3 (cited on page 2).
- Reynoso-Meza, Gilberto, J. Sanchis, Xavier Blasco, and Sergio Garcia-Nieto Rodriguez (Nov. 2014). "Physical programming for preference driven evolutionary multi-objective optimization." In: *Applied Soft Computing* 24, pp. 341–362 (cited on page 51).
- Rossi, Francesca, Peter van Beek, and Toby Walsh (2006). *Handbook of constraint programming*. Elsevier (cited on page 4).
- Ruiz-Femenia, R., G. Guillén-Gosálbez, L. Jiménez, and J. A. Caballero (2013). "Multi-objective optimization of environmentally conscious chemical supply

- chains under demand uncertainty." In: *Chemical Engineering Science* 95, pp. 1–11. issn: 0009-2509 (cited on page 27).
- Sakawa, Masatoshi and Ryo Kubota (2000). "Fuzzy programming for multiobjective job shop scheduling with fuzzy processing time and fuzzy due date through genetic algorithms." In: *European Journal of Operational Research* 120(2), pp. 393–407. issn: 0377-2217 (cited on page 48).
- Santos, Pedro, José Luis Pitarch, Alberto Vicente, Cesar de Prada, and Ángel García (2020). "Improving operation in an industrial MDF flash dryer through physics-based NMPC." In: *Control Engineering Practice* 94, p. 104213. issn: 0967-0661 (cited on page 2).
- Sarabia, D., C. de Prada, E. Gómez, G. Gutierrez, S. Cristea, J.M. Sola, and R. Gonzalez (2012). "Data reconciliation and optimal management of hydrogen networks in a petrol refinery." In: *Control Engineering Practice* 20(4). Special Section: IFAC Symposium on Advanced Control of Chemical Processes - AD-CHEM 2009, pp. 343–354. issn: 0967-0661 (cited on page 3).
- Skogestad, Sigurd (2007). *Multivariable feedback control: analysis and design*. Vol. 2 (cited on page 2).
- Smith, Jeffrey S, David T Sturrock, and W David Kelton (2018). *Simio and simulation: modeling, analysis, applications*. Simio LLC (cited on page 64).
- Stein, Oliver, Jan Oldenburg, and Wolfgang Marquardt (2004). "Continuous reformulations of discrete–continuous optimization problems." In: *Computers & Chemical Engineering* 28(10). Special Issue for Professor Arthur W. Westerberg, pp. 1951–1966. issn: 0098-1354 (cited on page 74).
- Stumbo, CR (1973). "Thermobacteriology in food processing. Academic." In: *Inc., New York, NY* (cited on page 58).

-
- Umetani, Shunji, Yuta Fukushima, and Hiroshi Morita (2017). "A linear programming based heuristic algorithm for charge and discharge scheduling of electric vehicles in a building energy management system." In: *Omega* 67, pp. 115–122 (cited on page 4).
- Vilas Fernández, Carlos and Antonio A. Alonso (2018). "Real time optimization of the sterilization process in a canning industry." eng. In: *Actas de las XXXIX Jornadas de Automática, Badajoz, 5-7 de Septiembre de 2018* (cited on pages 61, 103).
- Vilas, Carlos, Ana Arias-Méndez, Míriam R. García, Antonio A. Alonso, and E. Balsa-Canto (Feb. 2018). "Toward predictive food process models: A protocol for parameter estimation." In: *Critical Reviews in Food Science and Nutrition* 58(3), pp. 436–449. issn: 1040-8398 (cited on page 59).
- Wächter, Andreas and Lorenz T Biegler (2006). "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming." In: *Mathematical programming* 106(1), pp. 25–57. issn: 1436-4646 (cited on page 26).
- Wagner, Harvey M. (1959). "An integer linear-programming model for machine scheduling." In: *Naval Research Logistics Quarterly* 6(2), pp. 131–140 (cited on page 67).
- Wallace, John M and Peter V Hobbs (2006). *Atmospheric science: an introductory survey*. Vol. 92. Elsevier (cited on page 30).
- Winston, Wayne L. and Jeffrey B. Goldberg (2004). *Operations research: applications and algorithms*. 4th ed. Thomson/Brooks/Cole: Belmont, CA. isbn: 978-0-534-38058-8 (cited on pages 5, 74).
- Zhang, Qi and Ignacio E. Grossmann (2016). "Enterprise-wide optimization for industrial demand side management: Fundamentals, advances, and perspectives."

In: *Chemical Engineering Research and Design* 116. Process Systems Engineering - A Celebration in Professor Roger Sargent's 90th Year, pp. 114–131. issn: 0263-8762 (cited on page 16).

Zhou, Kemin and John Comstock Doyle (1998). *Essentials of robust control*. Vol. 104. Prentice hall Upper Saddle River, NJ (cited on page 13).