ENTERPRISE-WIDE OPTIMIZATION: INTEGRATING PLANNING, SCHEDULING AND CONTROL PROBLEMS USING FEASIBILITY ANALYSIS AND SURROGATE MODELS

By

LISIA MARINA SCHOLZ DIAS

A dissertation submitted to the

School of Graduate Studies

Rutgers, The State University of New Jersey

In partial fulfillment of the requirements

For the degree of

Doctor of Philosophy

Graduate Program in Chemical and Biochemical Engineering

Written under the direction of

Marianthi Ierapetritou

And approved by

_____

_____

_____

_____

New Brunswick, New Jersey

October 2019

ABSTRACT OF THE DISSERTATION

Enterprise-Wide Optimization: Integrating Planning, Scheduling and Control Problems Using Feasibility Analysis and Surrogate Models

By LISIA MARINA SCHOLZ DIAS

Dissertation Director:

Prof. Marianthi Ierapetritou

The US Industrial manufacturers face numerous challenges such as increasing complexity of production processes, fluctuating customer demands and expansion of supply chains. Output is expected to increase only 3.5% in 2019, according to the International Monetary Fund. The global expansion has weakened, foreign trade is at historically low levels, and nationalist governments around the world are threatening to further undermine the free flow of goods, creating more uncertainty and constraints upon manufacturing growth. In such a slow-growth environment, productivity gains are essential and there is an opportunity to profit from innovative strategies. In the field of operations research and process systems engineering, the main strategy to combat the emerging challenges and improve the efficiency of process industry is the pursue of optimal operating conditions through an enterprise-wide optimization (EWO).

EWO involves optimizing the operations of supply, manufacturing and distributions activities of a company. A major focus in EWO is the optimal operation of manufacturing facilities, which involves the decision-making processes of planning, scheduling and real-time operational control. Traditionally, these decision-making

problems are addressed individually and in a hierarchical manner, solved in a sequential way. An upper level problem is often solved with few or none information from lower levels. Its result is then transmitted to the lower levels, which must be optimized given the conditions already set by upper level problems. Consequently, sequential approaches may result in sub-optimal and infeasible solutions that can be avoided by an appropriate integration of different decision layers.

The objective of this work is to provide tools and the technology to establish optimal operating conditions by modernizing and integrating the decision-making process within a company. The integration is achieved by using simulation-based optimization techniques, surrogate modelling and feasibility analysis to transmit the information from lower levels to upper levels of the decision-making hierarchy. The problem of integrating scheduling and control is first addressed, followed by the problem of integrating planning and scheduling problems. By coupling the fundamentals of the developed integration strategies, enterprise wide optimization is achieved. The problem of planning, scheduling and control of a complex industrial-sized problem is then solved to demonstrate the adaptability, viability and performance of the proposed framework.

Acknowledgements

Looking back to the past four years of my journey as a PhD candidate, I realize I have been extremely lucky to have received constant and abundant help and support from my mentors, family and friends. I couldn't have done this without all of you.

First, I would like to express my deepest gratitude to my advisor, professor Marianthi Ierapetritou. Prof. Marianthi has been an incredible source of inspiration, motivation, and support. I thank her for trusting my abilities even when I doubted them the most. I thank her for the passionate discussions that have inspired every piece of this dissertation. I appreciate her guidance through difficult times, and her constant dedication to her students. Above all, I thank her for always pushing me to achieve the unachievable.

I would also like to sincerely thank my committee members, Dr. Rohit Ramachandran, Dr. Ioannis Androulakis, and Dr. Zukui Li for their valuable feedback and suggestions.

My time at Rutgers has been enriched by amazing fellow graduate students, who provided insights and friendship at every step of this journey. Thanks to you: Isabel, Clara, Alaina, Nirupa, Parham, Zilong, Atharv, Pooja, Ou, Abhay, Shu, Yue, Nihar, Sebastian, Yingjie. I would also like to thank my dearest Brazilian friends: Karin, Luiz, Valdir, Vito, Silvio, Alison, Pedro, Erik. I would never finish my journey without the joy and the fun that this group of people have provided.

Finally and most importantly, I would like to thank my parents Paulo and Ana, my sister Sarah and my brother Victor. They were my source of strength, and I thank them for their encouragement and constant support. Without them, I would not have chosen to pursue a PhD, and I would not be the person I am today. I would also like to thank my love,

my boyfriend, and my partner, Pupim. Thank you, for your endless support, for surrounding me with love when I needed the most, and for being there, in the hardest of times.

This work was financed by CNPq – National Council for Scientific and Technological Development – Brazil.

Table of Contents

List of tables

List of illustrations

Chapter 1

Introduction

1.1 Enterprise wide optimization

The process manufacturing industry has a tremendous significance to the US economy. It accounted for 11.6% of gross domestic product income (GDP) in the economy in 2017, and it is directly responsible for 12.75 million jobs in the US, according to the Bureau of Economic Analysis. However, this segment faces numerous challenges such as the constantly growing world-wide competition, increasing complexity of production process, fluctuating customer demand and expansion of supply chains, as well as huge structural cost disadvantages when compared to its major competitors, according to the Manufacturing Institute [1]. In the field of operations research and process systems engineering (PSE), the main strategy to combat these emerging challenges and improve the efficiency of process industry is the pursue of optimal operation conditions through an enterprise-wide optimization.

Enterprise-wide optimization proposes to optimize decision-making processes related to supply, manufacturing and distribution within a company. The major operational decisions include planning, scheduling and process control [2, 3], usually represented in a hierarchical way (Figure 1.1). On one end, tactical decisions determine allocation of resources on a time scale of months and years; on the other end, operational decisions address disturbances on a time scale of seconds.

In this hierarchy, planning problems are at the higher level of the decision-making. Based on demand forecasts and production orders, planning problems attempt to allocate resources and define production and inventory targets for manufacturing plants and storage facilities. The planning problem oversees the enterprise operation, coordinating it with market requirements and business considerations. Planning problems are usually modeled as linear or mixed integer linear problems, and are solved for a time horizon of months or years.

Below the planning problem in the decision-making hierarchy, scheduling of production ensures the profitable and effective operation of an industrial process plant. A suitable and optimal schedule will guarantee that equipment, material, utilities, personnel and other resources are available whenever a production task must be performed. Therefore, the purpose of scheduling is to optimally allocate limited resources to processing tasks over time, while ensuring that demands are met and guarantying profitable operations [4]. In practice, scheduling has been done manually by expert individuals using spreadsheets in an ad-hoc basis [5]. However, increasing production volumes, larger product portfolios and volatile customer demands raised the complexity of the decision-making process, and spurred the development of rigorous and mathematical formulations for scheduling problems. Many different classes and scheduling models can be found in the literature [6, 7], yet they all share the purpose of ensuring the feasible and optimal operation of a processing plant. Scheduling problems are usually modeled as mixed integer linear problems, and are solved for a time horizon of days or weeks.

Process control is the lower-level decision-making process which supports the implementation of scheduling solutions at the production floor, while also attempting to

ensure the stability of the system and the control of qualitative properties of streams. A very powerful and widely adopted control technique is Model Predictive Control (MPC) [8]. MPC utilizes an explicit process model to predict the future state of a plant. At each control interval, control actions are obtained by solving online a finite horizon open-loop optimal control problem, which takes into consideration the current state of the plant and the future predictions. The optimization yields an optimal control sequence; the first control in this sequence is sent to the plant, and the entire calculation is repeated at subsequent control intervals [9]. MPC has the capability of tracking setpoints established by the scheduling layer, while considering constraints on control and state variables and contributing to throughput maximization and cost-efficient production [10]. MPC problems are usually formulated as quadratic or nonlinear optimization problems, and are implemented on a time scale of minutes or seconds.

Traditionally, planning, scheduling and control problems are considered individually and solved in a sequential way. Planning problems usually set production targets for different manufacturing facilities which, in their turn, try to achieve the targets by defining an appropriate schedule. Scheduling decisions can be translated to setpoints and production sequences, which are transmitted to the control layer. The control attempts to implement the scheduling decisions, while handing disturbances over time. Such sequential strategies result in upper-level decisions made with little knowledge of the production constraints at the lower levels of the decision-making hierarchy. The decisions are usually assumed to be inflexible, and lower-levels decision makers either achieve the targets at higher costs, or fail to achieve those targets [11].

Driven by the possibility of determining an overall optimal and feasible solution, many researchers have explored the problems of integrating two or more decision making process, and techniques to solve the complex resulting problems have been developed. One of the major challenges in this integration is dealing with the different time scales related to each individual level. Usually, the simplest alternative for solving integrated problems is to adopt the time scale of the lower-level problem and formulate a single simultaneous model by incorporating the detailed lower-level problem as constraints in the upper-level model. However, this approach becomes computationally intractable when applied to large time horizons and high-dimensional problems [2]. To address these challenges, several approaches have been proposed in the literature and are discussed next.

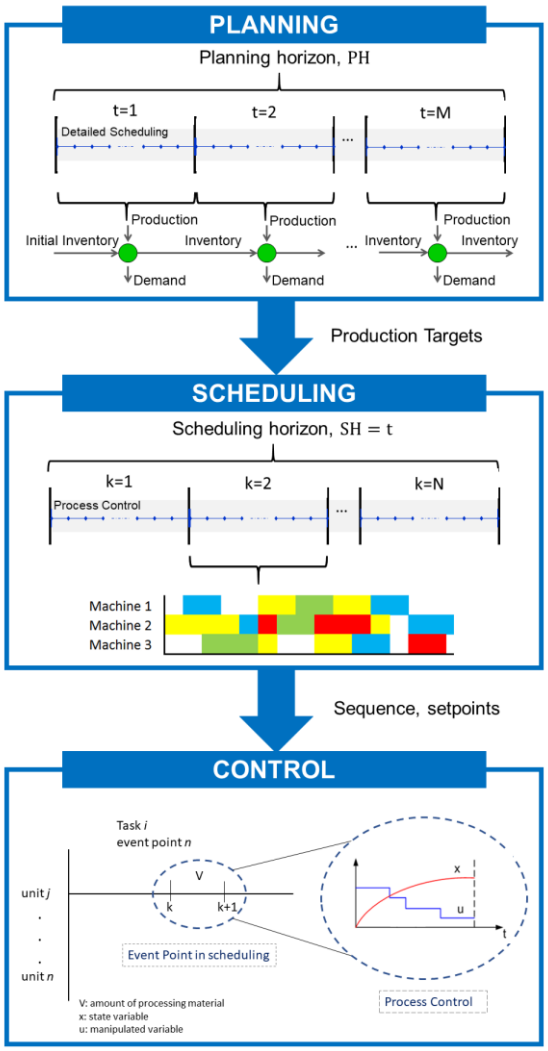

Figure 1.1 - Decision-making hierarchy

1.2 Background

1.2.1 Integration of planning and scheduling

Although associated with different time horizons, production planning and scheduling are closely related problems. While production planning determines the optimal allocation of resources and production targets over a time horizon of weeks or months, scheduling

determines the assignments of tasks to units and the sequencing of tasks over a time horizon of days or weeks. Owning to the interdependencies between planning and scheduling, the simultaneous consideration of these problems can guarantee solutions closer to optimality.

As identified by Maravelias and Sung [12], a general formulation for the planning problem is given by resource constraints, production and holding costs constraints and material balances. The integration of planning and scheduling translates to incorporating scheduling information in order to determine the resource and production cost constraints. The formulations proposed in the literature can be classified into four categories: (a) detailed scheduling models, (b) relaxations/aggregations of scheduling models, and (c) surrogate model or approximations of the feasible scheduling region derived through off-line analysis of manufacturing facilities. The first group includes intuitive ideas such as replacing the resource and production cost constraints by a monolithic scheduling model over the entire planning horizon. Clearly, such methods result in complex mathematical formulations that can become intractable for large time horizons and high-dimensional problems. The second group tries to handle the complexity problem by removing some of the scheduling constraints [13-16], or by aggregating some of the decisions of the original scheduling formulation [17-20]. The third group proposes to use surrogate models or multi-parametric programming techniques to generate constraints that define the feasible region of the scheduling model and the production cost as function of production targets [21-24]. Such methods can generate more accurate and computationally tractable descriptions of resource and production costs constraints.

Regarding the solution strategies, three methodologies have been proposed in the literature: hierarchical, iterative and full-space methods. In hierarchical methods, the

problem is decomposed into a master subproblem, which includes the planning model and approximate scheduling formulation, and a slave subproblem with detailed scheduling. The master problem is solved first, and its solution is used as input for the slave problem. Hierarchical methods can also be used within a rolling horizon framework, where detailed scheduling models are used in the master problem in a few early periods, and aggregate models are used in later periods. Since the production targets for the early periods are exact, they are implemented and the targets for later periods are updated as the horizon rolls[25]. Alternatively, iterative procedures have been proposed to close the information loop between the master and subproblem, and determine feasible production targets [18, 22, 26]. This can be achieved by the addition of integer cuts that exclude previously found solutions. Finally, the intuitive idea of integrating a detailed scheduling model to the planning formulation has been proposed in the literature (full-space models), usually associated to decomposition techniques to speed up computations of an otherwise intractable problem [27].

1.2.2 Integration of scheduling and control

While process control and scheduling of production share the common objective of identifying (economically) optimal and feasible operational decisions [28], these problems have been traditionally approached in a hierarchical and irrespective manner. Decisions at the scheduling layer are made independently of the dynamic behavior and control of the system; the decisions are then transmitted to the control layer, which effectively implements the scheduling solutions. However, significant efforts have been made by the process systems engineering community to coordinate and integrate the scheduling and

process control problems in view of the possibilities of improving performance and profitability when decision-making processes are addressed simultaneously [2].

Perhaps the most intuitive way of achieving the scheduling and control integration is to adopt the time scale of the control problem and formulate a single simultaneous model that incorporates the dynamic model of the process as constraints in the scheduling model. The result is a mixed integer dynamic optimization problem (MIDO), that is typically discretized into a Mixed Integer Nonlinear Program (MINLP) using, for example, collocation or implicit Runge Kutta methods. This approach was initially proposed by Flores-Tlacuahuac and Grossmann [29], and later extended by Terrazas-Moreno, Flores-Tlacuahuac [30] and Zhuge and Ierapetritou [31]. Alternatively, decoupled modelling approaches were proposed by Kroll and coworkers [32, 33], who attempted to solve the integrated scheduling and control problem by iterating between a master problem formulated as a MILP (scheduling problem) and a primal problem formulated as a dynamic optimization problem (process control). However, the use of high-fidelity representations of the process dynamics as constraints in the scheduling problem usually introduces complexities, nonlinearities and discontinuities to the optimization model. These challenges have been addressed by You and coworkers, who proposed a series of decomposition strategies to improve the computational efficiency in the solution of the integrated problem [34, 35].

Although the contributions discussed so far brought significant theoretical advances to the integrated scheduling and control problem, they usually do not consider questions related to the stability and disturbance rejection at the dynamic level (problems extensively studied in the control literature). Most of the frameworks proposed to solve the

integrated problem offline, defining control actions for the entire scheduling horizon with no assessment of the online performance of the system. Motivated by these issues, Zhuge and Ierapetritou [36] proposed to integrate scheduling and model predictive control by deriving explicit control laws for the MPC problem using multi-parametric programming techniques, and further incorporating such laws in the scheduling problem. The strategy enables the definition of control actions that implicitly take into account the MPC objectives (stability, safety, robustness and fast-tracking). Zhuge and Ierapetritou [37] proposed an integrated scheduling and MPC framework consisting of the use of two control loops. An integrated problem at the outer loop generated the production schedule and the state references for the inner loop. The inner loop tracked state references using fast model predictive control, and the exact control solution was computed online. More recently, Simkoff and Baldea [38] proposed to integrate scheduling and linear MPC by deriving the KKT conditions of the control problem, and incorporating such conditions at the scheduling model. The main drawback of these approaches, however, is the exponential increase of computational expenses with the dimensionality of the control problem, restraining their use in industrial-size applications.

A third strategy to handle the integrated scheduling and control problem is the use of surrogates derived from historical data and simulations. Surrogate models can capture the closed-loop behavior of the dynamic system in a time scale that is relevant to the scheduling problem. Such strategies have been applied to small scale problems [39, 40] as well as Air Separation Units [41, 42]. Although efficient, the proposed frameworks rely on manually identifying a set of "scheduling-relevant" variables and active constraints in the control problem to be approximated and modeled at the scheduling level.

1.3 Challenges and opportunities

Even though significant advances have been made by the process systems engineering community on the integration of planning-scheduling and scheduling-control problems, there is no unifying framework that systematically addresses high-dimensional problems. Few works attempted an overall integration of planning, scheduling and control, including a full-space formulation for short-period planning problems [43]; a decomposed framework with iterations between a MILP and DO problem [44]; and a nonlinear multi-parametric programming approach for the derivation of explicit controllers and further integration with planning and scheduling [45]. There is little research work on the effects of uncertainties in the integrated problems [4]. Moreover, in an era of digitization and extensive gathering of information, relatively few works focused on the possibilities that Big-data technologies can bring to the process systems engineering community. Early attempts in this direction include the use of neural networks to approximate dynamic models of systems and the use of such approximations in model predictive controllers [46, 47]; and the use of neural-networks to approximate explicit control laws [48, 49]. More recently, data-driven methods have been applied to construct polyhedron uncertainty sets, leveraging the power of machine learning and big data analytics for decision-making under uncertainty [50]. There is a huge potential in expanding the scope of data-driven methodologies within PSE and, in particular, within the problem of integrating decision making strategies.

1.4 Objectives

The objective of this work is to develop a systematic approach for the integration of decision-making processes. This work recognizes that planning, scheduling and control problems are modeled using different mathematical optimization concepts, and they address problems with different time scales. However, the underlying problems of integrating planning-scheduling and scheduling-control have several common aspects that can be addressed with a standard framework. The framework is inspired and composed by the concepts of simulation-optimization [51, 52], feasibility analysis [53], and surrogate models [54]. Therefore, this work aims to investigate and address each of these components individually, and combine them in a unified framework for the integration of planning, scheduling and control problems. Furthermore, each of the components are developed to take advantage of big-data analytics, and to further integrate classical optimization and data-based methodologies.

Specifically, the following research objectives will be studied:

Specific aim 1: Integration of scheduling and model predictive control using a hybrid simulation-optimization method

Model predictive control (MPC) has been widely and successfully adopted in manufacturing plants around the world. Therefore, the ideal framework for the integration of scheduling and control should take this factor into account. Beyond integrating scheduling and dynamic systems, the framework should aim in integrating scheduling and the closed-loop behavior of a dynamic system operating under MPC. To address this issue, a hybrid simulation-optimization approach for the integration of scheduling and control

was proposed. The framework consists of (i) solving a simulation-optimization problem to define the optimal integrated schedule and, (ii) tracking the schedule in closed-loop using the MPC controller.

The proposed framework has limited applications which will be discussed in Chapter 2. Nevertheless, the theoretical developments of this dissertation were greatly inspired by the concept of simulation-optimization. This concept had not been implicitly applied to the integration of decision-making layers in previous works, and it served as the theoretical foundation for the development of the systematic framework for decision-making integration, discussed in the next research aims.

Specific aim 2: Integration of scheduling and robust model predictive control using surrogate models

To achieve a systematic integration of decision-making processes, closed-loop information from the lower levels of the decision-making hierarchy should be transmitted to the higher levels. This work addressed this problem by employing surrogate models to approximate the closed-loop behavior of a system operating under different control strategies. Furthermore, it explores the issue of uncertainties at the control level, and it investigates how different control strategies can affect the integrated scheduling and control problem.

Specific aim 3: Data-driven feasibility analysis for the integration of planning and scheduling

One of the critical information that should flow among the different layers of the decision-making hierarchy is the feasible space of each problem. In the integration of planning and

scheduling, for example, it is critical to model the feasible space of the scheduling problem, and integrate it to the planning model. However, defining the feasible space of the scheduling problem is challenging: these problems are usually modeled as Mixed Integer Linear Problems (MILP), and there are no mathematical approaches to rigorously determine the feasible space of a MILP problem in its entirety. To address this issue, a data-driven feasibility analysis methodology is proposed. The method is based on the use of different classification methodologies, and it is applied to the problem of integrating planning and scheduling. Chapter 4 will cover this research objective.

Specific aim 4: Overall integration of planning, scheduling and control

The ultimate goal of enterprise-wide optimization is the integration of planning, scheduling and control problems. Therefore, a systematic framework for the integration of these decision-making processes is proposed. The framework is an extension of the previous research aims, and it employs the feasibility analysis methods developed in specific aim 3, and the surrogate models developed in specific aim 2. The resulting integrated problem will take the form of a Mixed Integer Nonlinear Problem (MINLP). Since this problem can be challenging to solve, we propose to reduce its dimensionality by employing the concept of feature selection in the derivation of the surrogate models. This work is presented in chapter 5.

Chapter 2

Simulation-optimization approaches for the integration of scheduling and control

Abstract: In this chapter, a hybrid simulation-optimization framework is proposed to address the challenge of integrating scheduling and model predictive control (MPC). The framework consists on identifying scheduling-relevant process variables, building low-order dynamic models to capture their evolution, and integrating scheduling and MPC by, (i) solving a simulation-optimization problem to define the optimal schedule and, (ii) tracking the schedule in closed-loop using the MPC controller. The efficacy of the framework is demonstrated via a case study that considers an air separation unit operating under real-time electricity pricing. The study shows that significant cost reductions can be achieved with reasonable computational times.

2.1 Introduction

Globalization and extensive information exchange supported by new technologies have given rise to an environment with fast changing market conditions, which must be taken into account in order to achieve optimal process operation. In the process systems engineering community, the search for the optimal operation has been translated to optimizing planning, scheduling and control decisions across the entire enterprise. In this chapter, we focus on strategies for the integration of scheduling and control problems. Specifically, we propose a nested decision-making structure, comprising two scheduling/control loops. In the *outer* loop, the scheduling problem is formulated as a simulation optimization problem, where the simulation involves solving a model predictive control problem for the entire scheduling horizon. The problem is solved to determine the optimal sequence of production rate setpoints. The solution is then communicated to the

inner loop, which tracks the setpoints over time while ensuring feasibility, stability and fast tracking. The inner loop is solved online and handles any disturbance that may affect the control layer.

The concept of a hybrid simulation-optimization has been employed by several works addressing decision making-processes [55-57]. However, to the best out knowledge, such concept has not been applied in the integration of scheduling and control. Hybrid simulation-optimization frameworks are usually employed when detailed mathematical models describing the system cannot be derived or cannot capture the complexities of the system under investigation. Therefore, such frameworks are suitable for the representation of complex dynamic systems operating under a model predictive control, for which explicit control laws and closed-form expressions for the closed-loop behavior may be difficult or impossible to derive.

The hybrid simulation-optimization framework proposed in this chapter will be applied to the problem of scheduling in an Air Separation Unit (ASU), extending the work proposed by Pattison, Touretzky [41]. Air separation units have a very high energy consumption, and the industrial gas sector utilized 19.97TWh of electricity in 2014, or about 2.55% of the amount consumed by the entire manufacturing sector in the U.S. [41]. Therefore, this industry may take advantage of the variations in electricity prices, which can change hourly (or faster) due to the change of grid-level electricity demand. Costs may be reduced by adjusting production rate according to electricity prices. Such frequent changes, however, may result in the process operating in a transient regime, possibly without ever reaching steady state conditions. This fact is contrary to the premise of most

current scheduling methodologies and emphasizes the necessity to account for the dynamic behavior of the system while making decisions at the scheduling level.

This chapter is organized as follows: In section 2.2, the problem statement is presented, followed by a description of the ASU process. In section 2.3 we describe the problem formulation, discussing the process, control strategies, and scheduling model, followed by the integrated framework description. We demonstrate the application of the framework to the ASU process in section 2.4 and provide concluding remarks and future directions in section 2.5.

2.2 Problem Definition and Motivating Example

2.2.1 Problem Definition

Consider the case of a continuous process that consume significant amounts of electricity. The process consists of several interconnected pieces of equipment/unit operations. Each unit may contribute to the power consumption of the overall plant. A storage system is available for the finished product; the product demand is given. Production rates can fluctuate (thereby changing the electricity demand of the process) and the product demand can be met either directly (i.e., by the plant itself) or by a combination of plant production and depleting the inventory of stored product.

We assume electricity is purchased in a day-ahead market, for which accurate price forecasts are available. The goal is to define the production rates (and associated rate of inventory accumulation/depletion), that lead to the optimal operating cost for a given time horizon. The following summarizes the problem statement:

Given

- Time horizon of the scheduling problem
- Demand rates
- Price forecasts
- Dynamic process model
- Dynamic model of storage system
- Inventory constraints
- Product quality constraints
- Process operating constraints

Determine

- Optimal schedule: production rate targets and rate of inventory accumulation/depletion in each time
- Optimal operating cost and/or profit for given time horizon
- (Real-time) optimal control moves that implement the optimal schedule

We assume deterministic conditions across all levels of the decision-making process. We also assume the only independent scheduling decisions are the definition of the production rate setpoint/target at each time interval of the scheduling horizon. No discrete decisions or multiple units have been considered. Moreover, we focus on continuous chemical processes that produce the same product(s) at variable production rates (rather than switching between a finite product palette). For simplicity, we assume that a single product is made, although the discussion below can be readily extended to the case where multiple products are made simultaneously.

We formulate and solve the above problem for an air separation unit, as a prototype electricity intensive process. The plant is described next.

2.2.2 Air Separation Unit Model

We base our work on the model of a cryogenic air separation process as described by Pattison, Touretzky [41]. Air separation units (ASUs) are an important industrial utility provider, supplying oxygen, nitrogen, and argon to chemical, manufacturing, and energy generation industries. ASUs also have very high electricity consumption, and, in order to improve operating economics, researchers have investigated the possibility of taking advantage of electricity price fluctuations. By switching operating modes [58] or storing energy in the form of cryogenic liquids [59], significant cost reductions can be achieved. Several studies confirm these findings by integrating linear and nonlinear steady state models to represent process performance, and optimizing the operation of ASUs (in particular, by varying production rates) subject to time-varying electricity prices [60-62]. In general, if excess production capacity and storage capabilities are available, over-production is sought in periods of low electricity prices (storing excess products), and low production rates are set (with the use of stored products to satisfy the demand) as electricity prices increase. Operating patterns characterized by the aforementioned paired events (production increases/decreases in response to fluctuations in electricity prices) belong to the broader operating paradigm referred to as "demand-side management" or "demand-response".

One important characteristic of ASUs that must be considered when exploring demand response and frequent transitions in production rates is the fact that their settling time (i.e., the time to reach steady state after a change in process inputs or controller set points) is typically in the order of hours [63]. When utility prices (and, consequently production rate targets) change at a high (e.g., hourly) frequency, accurate dynamic models

of relevant process variables should be utilized to ensure that a sequence of scheduled production rate transitions is feasible and optimal.

The process flowsheet is presented in Figure 2.1. A single cryogenic distillation column is used for producing high-purity nitrogen. The inlet air feed is first compressed to 6.8 bar and then cooled to the saturation point in a multi-stream primary heat exchanger (PHX). A fraction of the air is removed from the PHX and sent through a turbine to generate electricity, and the balance is liquefied in the PHX. Both the liquid and gas air streams are fed to the bottom of the cryogenic distillation column. The bottoms of the column, which is an oxygen-enriched mixture, is expanded adiabatically and sent to the reboiler where it provides cooling to the condenser in an integrated reboiler/condenser unit. The distillate of the column is purified nitrogen, and is split into a product stream and a reflux stream, where the reflux is sent to the condenser to be liquefied and returned to the column. The product stream is expanded in turbine 2, and both the waste from the reboiler and the nitrogen product stream exit the plant through the PHX to provide cooling for the inlet air. A separate nitrogen liquefier is included in the flowsheet along with a nitrogen product storage tank and an evaporator. This nitrogen storage allows the process to meet gas nitrogen demand with (evaporated) stored liquid nitrogen when electricity prices are high and production rate is decreased. During periods of low electricity price, production can be increased to build up liquid nitrogen inventory.

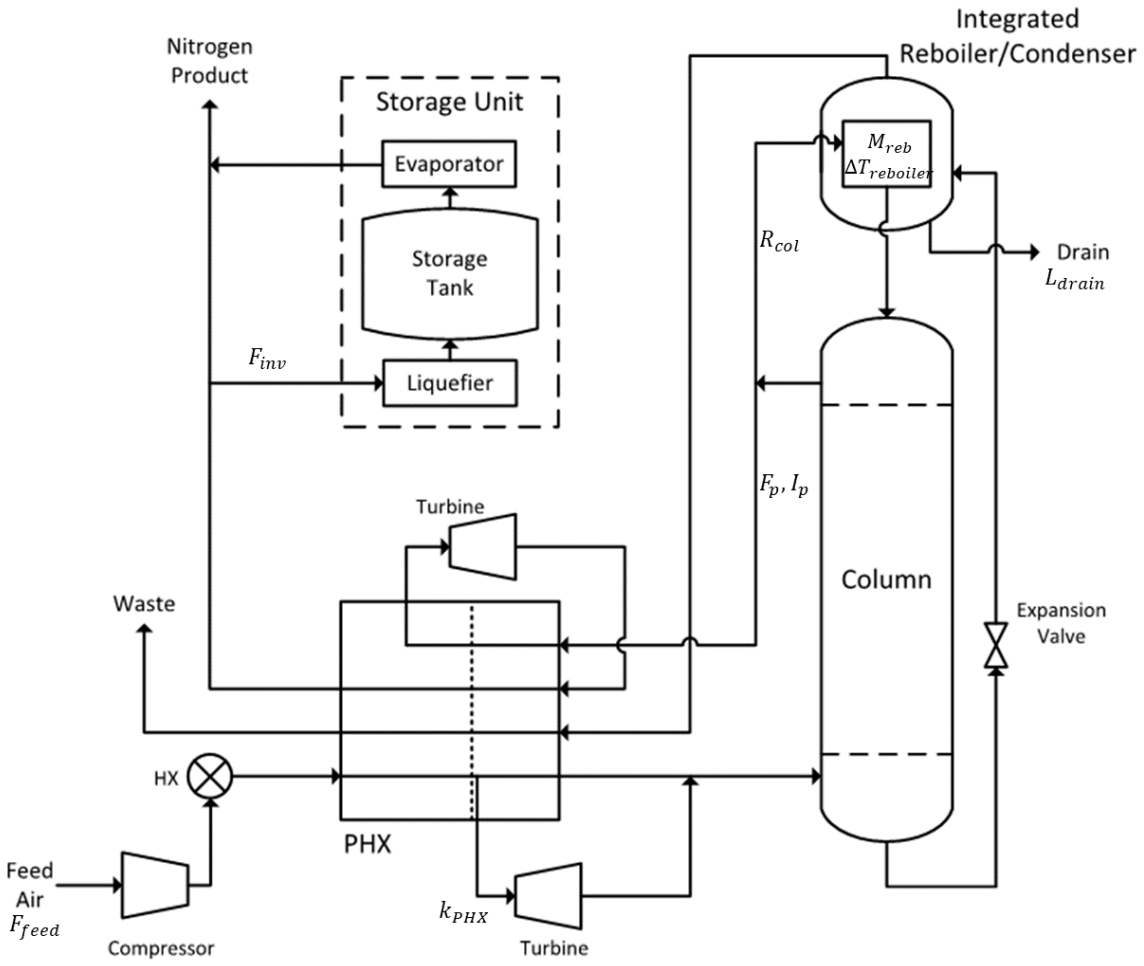

Figure 2.1 - Flowsheet for nitrogen production using a cryogenic air separation unit

The process model was presented in detail in Johansson [64] and Pattison, Touretzky [41], which in turn is based on the models developed by Cao [65], Cao, Swartz [63] and Huang, Zavala [66]. We include a synopsis of the mathematical models describing the dynamic behavior of the unit operations of the ASU process shown in Figure 2.2Figure 2.2, noting that the detailed process model comprises 6094 equations and has 430 state variables.

Distillation Column Model. The cryogenic distillation column model is based on the work by Huang, Zavala [66]. The column consists of 30 equilibrium stages and the

condenser operating pressure is 6.4 bar with a 0.2 bar linear pressure drop along the column. The phase equilibrium is modeled using a Margules activity model for the nonideal liquid phase $y_{ij}P_i = \gamma_{ij}P_{ij}^{sat}x_{ij}$, where index i and j represent the stage number and component, respectively. The vapor pressure, P_{ij}^{sat} is determined using Antoine's equation and the activity coefficients, γ_{ij} are determined using the Margules equations.

Integrated Reboiler/Condenser Model. The oxygen rich stream at the bottom of the distillation column is expanded through a valve to 2.5 bar to provide cooling (via Joule-Thomson effect) to the condenser in the heat-integrated reboiler/ condenser. The steady state model for the integrated reboiler/ condenser is adapted from the work of Cao [65]. The reboiler is modeled as an equilibrium stage, with an additional heat input equivalent to the condenser heat duty.

PHX Model. The PHX is a brazed plate-fin multistream heat exchanger and the dynamic model is adapted from the work of Cao [65]. The model consists of two zones which are delimited by the location where a fraction of the inlet air gas stream is withdrawn from the PHX in the gas phase. The fraction of cooled air removed at this point is a manipulated variable for the control system. Zone 1 corresponds to sensible heat removal from the inlet air stream, while zone 2 corresponds to latent heat removal. The first zone is further discretized into 50 segments, while the second zone is modeled by a single lumped energy balance equation to simplify the phase transformation calculations. The geometry of the channels within each segment is accounted for when calculating energy accumulation of each stream in each finite volume.

Compressor and Turbine Models. Generators are coupled to the turbine expanders used in the process to reduce the compressor power demand. We assume that the turbine,

compressor and generator dynamics are fast and the units can be approximated with steady state conservation equations. We assume that the compression and expansion are polytropic processes with corresponding head and efficiencies calculated using the approach presented in Chapter 10 of Green and Perry [67].

Liquefier and Liquid Storage Tank Model. We assume that the liquefier dimensions are much smaller than those of the ASU, and, unlike the ASU, the liquefier does not contain any significant material holdup (e.g., sumps) resulting in a fast dynamic response. As a consequence, we model the liquefier using steady state conservation equations. Further, it is assumed that the liquefier operation can be represented as an ideal refrigeration cycle with 80% efficiency. We assume that the evaporator is at ambient conditions and does not require any additional energy input to operate.

2.3 Framework for the integration of scheduling and MPC

2.3.1 Process system representation

The first-principles model of a chemical process is typically represented as a high-dimensional system of nonlinear differential equations (DAEs). Nevertheless, in practice, the number of process variables that are of interest to scheduling is likely much lower than the number of process states [40]. Pattison et al. (2016) postulated that, given a set of operating and quality constraints of a system, only those constraint that become active during static or transient operation are relevant to the optimal scheduling calculation because they can hinder the agility of the process, and thus only the variable trajectories related to these constraints are of interest to the scheduling problem.

The relevant set of variables, W, can be identified by analyzing historical operating data or using a set of dynamic simulations (Pattison et al., 2016). Subsequently, data-driven models can be identified to approximate the dynamic behavior of all schedule relevant variables. Note that, as is the case in any data-driven modeling approach, the models used here are valid for the operating regions/conditions for which data are available. In case the operating conditions of the plant change, model reidentification may be needed.

Focusing now on the air separation unit described above, we first note that the plant under consideration is capable of flexible operation, in the sense that the production rate (defined in terms of the flow rate of the gas nitrogen product) can assume any value in the range of 16mol/s to 24 mol/s. The nominal production rate is 20mol/s. The production rate set point/target is the independent variable in the integrated scheduling and control problem, and variations in this set point will define the performance and operating costs of the system. Therefore, the production rate is the first schedule relevant variable of our problem. The second schedule relevant variable is related to the purity of the nitrogen product stream. The maximum allowed impurity level (i.e., oxygen concentration) is 2000ppm at all times, as given by Pattison et al. (2016). Given the critical nature of this constraint, a back-off constraint is used, and we set $I^{max} = 1800ppm$. The back off constraint compensates for possible inaccuracies in the state-space models that will be derived to describe the dynamics of impurity levels in the product stream. The third relevant variable is defined as the temperature driving force across the reboiler/condenser, since this temperature gradient must be maintained above 1.8 K at all times, and frequent changes in the production rate set points may result in violation of this constraint. We define the fourth scheduling-relevant variable as the reboiler holdup, which must be kept

around its steady state value (100kmol) in order to guarantee operational stability and prevent the system from draining its required cooling holdup to generate a (false) improvement in operational economics.

The goal of the scheduling problem is to minimize the operational cost related to the energy consumption. Note that the energy consumed in the compressor, as well as energy produced in the turbines can be related to the feed air flow rate F_{feed} and the production rate. Therefore, F_{feed} also belongs to the set W of schedule relevant variables. Some additional variables are defined as manipulated variables for our control problem; we define the control inputs $u_p \in \mathbb{R}^m$ and outputs $y_p \in \mathbb{R}^q$ of our problem as:

- Production flow rate, $y_p(1) = F_p$
- Product purity (modeled in terms of impurity concentration), $y_p(2) = I_p$
- Temperature difference across the reboiler/condenser, $y_p(3) = \Delta T_{reboiler}$
- Liquid level in the reboiler, $y_p(4) = M_{reb}$
- Inlet air flow rate, $u_p(1) = F_{feed}$
- The split fraction in the PHX, $u_p(2) = k_{PHX}$
- The fraction of vapor product sent to the condenser, $u_p(3) = R_{col}$
- The liquid drain in the reboiler, $u_p(4) = L_{drain}$

Notice that all the scheduling relevant variables are classified either as manipulated variables or control outputs. The dynamics of the schedule relevant variables are captured using continuous state-space models of the form:

$$\dot{x}_p = Ax_p + Bu_p \tag{1a}$$

$$y_p = Cx_p \tag{1b}$$

where $x_p \in \mathbb{R}^n$ is the state, $A \in \mathbb{R}^{n \times n}$ is the state transition matrix, $B \in \mathbb{R}^{n \times m}$ is the input matrix, and $C \in \mathbb{R}^{q \times n}$ is the output matrix. The state-space models were identified using a highly detailed dynamic model of the process described in section 2. To identify the state space models, we first performed a series of open-loop step tests on the detailed model, where the manipulated variables were subject to step changes (applied one variable at a time), and transfer function models were then identified to approximate the dynamic step response of each output variable. The identification of the transfer functions was carried out using the System Identification Toolbox on MATLAB. These transfer function models were then combined to form a state space model. The state space model for the Air Separation Unit is defined by 38 state variables, 4 control inputs and 4 control outputs.

The aforementioned quality and process constraints related to the schedule relevant variables can be summarized as:

$$y_{min} \leq y_p \leq y_{max} \tag{2a}$$

$$u_{min} \leq u_p \leq u_{max} \tag{2b}$$

We note here that a linear state-space model and linear process quality and constraints have been identified for the Air Separation Unit problem. However, this is not a requirement of this framework. The state-space model will be employed in the process control problem and the closed-loop simulations of the system while the scheduling problem is being optimized. Therefore, we seek to achieve a tradeoff between model accuracy and computational performance while simulating the closed-loop behavior of the system. In this work, a linear model had a satisfactory performance in predicting the behavior of the system for the given operating range and the closed-loop simulation was performed in an order of seconds, making this model suitable for the simulation-optimization algorithm.

2.3.2 Process Control

Once the dynamic model of the system has been identified, a model predictive control (MPC) is designed for the process. This control layer is built to ensure the stability of the plant over its entire operation region, and it is assumed to be capable of imposing all changes in product type and production rate requested by the scheduling layer. The MPC controller is of the following form:

$$\min_{\boldsymbol{u_p}} J_{control}(x_p^0, \boldsymbol{u_p}) = \frac{1}{2}\sum_{k=0}^{N-1}\left(\left|y_p^k - y_p^{sp,k}\right|_Q^2 + \left|u_p^k - u_p^{sp,k}\right|_R^2\right) + \left|y_p^N - y_p^{sp,N}\right|_{P_f}^2 \quad (3)$$

$$s.t.\begin{cases} x_p^0 = x_{initial} \\ x_p^{k+1} = Ax_p^k + Bu_p^k \\ y_p^k = Cx_p^k \\ y_{min} \le y_p^k \le y_{max} \\ u_{min} \le u_p^k \le u_{max} \end{cases}$$

where the system described in (1) has been discretized and included as constraints in the MPC problem. k is a nonnegative integer denoting the sample number, which is connected to time by $t = kT_s$ in which T_s is the sample time. N is the prediction horizon, $y_p^{sp} \in \mathbb{R}^q$ is the setpoint, and Q and R are tuning parameters penalizing deviations from the setpoint and control moves, respectively. We allow the final state penalty to have a different weighting matrix P_f, for generality. Finally, $\boldsymbol{u_p}$ is the input sequence for $N-1$ time steps, i.e., $\boldsymbol{u_p} = \{u_p^0, u_p^1, \dots, u_p^{N-1}\}$.

For the Air Separation Unit control problem, we will use a sample time $T_s = 6$min. The production rate setpoints to be tracked by the control problem will be defined by optimizing the integrated scheduling and control problem. The set points for the remaining output variables are fixed at $y_p^{sp}(2) = I_p^{sp} = 500ppm$, $y_p^{sp}(3) = \Delta T_p^{sp} = 2.2^oC$ and

$y_p^{sp}(4) = M_{reb}^{sp} = 100kmol$. The MPC problem was implemented in MATLAB using the Model Predictive Control Toolbox.

2.3.3 Production Scheduling

One important consideration in any scheduling formulation is the time representation. In general, it is convenient to model time according to the frequency of change in the main drivers for making scheduling decisions. For the air separation unit problem operating under hourly change electricity prices, it is straightforward to define hourly time slots and assign different price values to different periods [68]. Furthermore, a natural choice for the time horizon for the scheduling problem is the amount of time for which economic forecasts (e.g. energy prices) are reliably available. Therefore, we follow such time representation, which can be expressed as:

$$t_{end}^{n} = t_{start}^{n} + \tau, \qquad \forall n \in N_n \tag{4a}$$

$$t_{start}^{n} = t_{end}^{n-1}, \qquad \forall\, n \neq 1 \tag{4b}$$

$$t_{start}^{1} = 0 \tag{4c}$$

$$t_{end}^{N_e} = T_m \tag{4d}$$

where $n \in \{1, 2, \dots, N_n\}$ is a positive integer value representing the time slots in the scheduling formulation, τ is the discretization period (i.e., 1 hour), t_{start}^{n} and t_{end}^{n} define the starting and ending time of the scheduling period n, respectively, and T_m is the scheduling horizon.

In the scheduling layer, we aim to minimize costs by changing production rate targets, while meeting demand and inventory constraints. We propose a formulation for the air separation unit problem where the only scheduling decisions are the production rate

setpoints $y_p^{sp,n}(1) = F_p^{sp,n}$ defined for each time slot n. The setpoints are constrained by upper and lower bounds (Eq. 5). Furthermore, to account for operational constraints in the system which may prevent feasible transitions, we add rate-of-change constraints to the scheduling problem in order to guide the schedule towards feasible transitions (Eq. 6). Note that $F_p^{sp,initial}$ is the setpoint for the air separation unit before the current scheduling horizon.

$$F_{p,min}^{sp} \leq F_p^{sp,n} \leq F_{p,max}^{sp}, \qquad \forall\, n \in N_n \tag{5}$$

$$F_p^{sp,n} \leq F_p^{sp,n-1} + s, \qquad \forall\, n \neq 1 \tag{6a}$$

$$F_p^{sp,n} \geq F_p^{sp,n-1} - s, \qquad \forall\, n \neq 1 \tag{6b}$$

$$F_p^{sp,1} \leq F_p^{sp,initial} + s \tag{6c}$$

$$F_p^{sp,1} \leq F_p^{sp,initial} + s \tag{6d}$$

Note that the parameter s is problem specific and can be determined by analyzing the performance of the schedule for different values of s. In this work, we assumed that the production rate can change by no more than 20% of the nominal production capacity each hour.

The constraints described so far are modeled independently of the control level performance. However, constraints related to the inventory levels, demand fulfillment and operational costs cannot be accurately predicted without a closed-loop simulation of the control level problem since they depend on the values of control inputs and outputs of the system. The details for the incorporation of the closed-loop simulation as a constraint in the scheduling problem will be discussed in the next section. For now, it is important to highlight that constraints that are somehow related to the control inputs and outputs should

be defined in a time scale comparable to the sample time of the MPC problem. In other words, considering the sample time of MPC is T_s and the scheduling horizon is T_m, the aforementioned constraints and costs of the system will be calculated for $i = T_m/T_s$ time intervals. For convenience, we defined the sample time of the MPC problem as an exact divisor of τ, and therefore an exact divisor of T_m.

For the ASU problem, the scheduling constraints related to control inputs and outputs of the system (adapted from Pattison, Touretzky [41]) are:

$$\alpha^i = \begin{cases} \dfrac{\bar{F}^i}{F_p^i} & if\ F_p^i \geq \bar{F}^i, \forall i \in I \\ 1 & if\ F_p^i < \bar{F}^i, \forall i \in I \end{cases} \tag{7a}$$

$$F_{inv,in}^i = \left(1 - \alpha^i\right) F_p^i, \qquad \forall i \in I \tag{7b}$$

$$M_{inv}^i = M_{inv}^0 \tag{8a}$$

$$M_{inv}^i \geq 0, \qquad \forall i \in I \tag{8b}$$

$$M_{inv}^i \leq M_{inv}^{max}, \qquad \forall i \in I \tag{8c}$$

$$M_{inv}^{T_m} \geq M_{inv}^{min} \tag{8d}$$

$$M_{inv}^i = F_p^i - \bar{F}^i + M_{inv}^{i-1}, \qquad \forall i | i \neq 0 \tag{8e}$$

Equation 7 defines the split, α^i, of the production rate of the process sent to the product storage tank. The inlet flow rate to the inventory is defined as $F_{inv,in}^i$. If the production rate F_p^i is greater than the demand, $\bar{F}^i$, the excess production is sent to inventory. If the production rate is lower than the demand, there is no flow to the inventory. Equation 8a states that the inventory level M_{inv}^i at period $i = 0$ must be equal to the initial inventory level; equations 8b and 8c define upper and lower bounds for the inventory level at every period i; and equation 8d defines a minimum inventory level at the end of the scheduling

horizon, in order to guarantee stability of the schedule and prevent the complete depletion of inventory. In this case, we set M_{inv}^{min} equal to $M_{inv}(0)$, its value at the beginning of the scheduling horizon. Note that is this scheduling framework were to be implemented in a rolling-horizon fashion, it would be beneficial to fix the terminal constraint to ensure recursive feasibility and stability [41]. Finally, equation 8e defines the inventory variation as the difference between the production rate and the product demand, which guarantees that any excess production will be sent to the storage, and any shortage on production will be compensated by the usage of stored nitrogen product.

The objective of the scheduling problem is to minimize production costs, which are mostly related to electricity consumption. The main consumers of electricity in an ASU are the compressors. Furthermore, the turbines are used to generate electricity, and there is a cost penalty for liquefying product to be sent to the storage tank. We assume the network of the compressor, liquefier and turbines is proportional to the flow rate, and therefore the objective of the production scheduling can be formulated as follows:

$$\min_{F_p^{sp}(t)} J_{sched} = \int_0^{T_m} p(t)\big(\gamma_c F_{feed}(t) - \gamma_{t1} F_{feed}(t) - \gamma_{t2} F_p(t) + \gamma_l F_{inv,in}^i(t)\big)dt \quad (9)$$

where $\gamma_c, \gamma_{t1}, \gamma_{t2}$ and γ_l are parameters (with units Watts/(mol/s)) which relate the flow rate in the compressor, turbine 1, turbine 2 and the liquefier, respectively, to the amount of electricity consumed (or generated) by the unit operation, and $p(t)$ is the forecasted electricity price throughout the scheduling horizon. Equation (9) can be approximated in a discrete-time form as:

$$\min_{F_p^{sp,n}} J_{sched} = \sum_i p^i T_s\big(\gamma_c F_{feed}^i - \gamma_{t1} F_{feed}^i - \gamma_{t2} F_p^i + \gamma_l F_{inv,in}^i\big) \quad (10)$$

2.3.4. Integrated scheduling and control

2.3.4.1 Closed-loop simulation-optimization problem

The integrated scheduling and control problem should include the closed-loop behavior of the system as constraints in the scheduling model. We represent this constraint by first defining a control law $\psi(\cdot)$:

$$\psi\left(x_p, y_p^{sp}\right) \coloneqq \arg\min_{\boldsymbol{u_p}} J_{control}(\cdot) \tag{11}$$

where $J_{control}(\cdot)$ was defined in eq. (3). Then, constraints (12) can be included in the scheduling model:

$$u_p^i = \psi\left(x_p, y_p^{sp,\lfloor i/\tau \rfloor}\right) \tag{12a}$$

$$x_p^{i+1} = Ax_p^i + Bu_p^i \tag{12b}$$

$$y_p^i = Cx_p^i \tag{12c}$$

The integrated scheduling and control problem can be summarized as:

Minimize Objective Function (10)

Subject to:

Closed-loop process behavior (Eq. 12)

Bounds and smooth constraints (Eq. 5-6)

Split equations (Eq. 7)

Storage model and constraints (Eq. 8)

(P1)

We refer to this problem as problem P1. Note that the inclusion of a closed-loop process simulation using an advanced process control system in the integrated scheduling and control framework is the main contribution of this paper. In the past, Zhuge and Ierapetritou [69] attempted to integrate scheduling and MPC by deriving explicit control

laws using multi-parametric programming techniques, and including such control laws in the scheduling formulation. However, as the problem size increases the number of critical regions in the multi-parametric problem increases exponentially, which limits the applicability of the framework. The integration of scheduling and MPC also imposes more challenges than the integration of scheduling and dynamic optimization problems. Such problems are usually formulated as Mixed Integer Dynamic Optimization problems (MIDO), which can be solved through sequential techniques or simultaneously through discretization and reformulation of the problem as a Mixed Integer Nonlinear Program (MINLP). However, the formulation in (P1) cannot be classified as a MINLP problem, nor can it be classified as a bilevel optimization problem. The closed-loop process simulation involves solving the MPC optimization problem i times during each execution of the scheduling problem, where $i = T_m/T_s$, and T_s is the sample time for the MPC.

Therefore, we choose to rearrange problem (P1) as a closed-loop simulation optimization problem, where the simulation is equivalent to sequentially solving the model predictive control problem for the entire scheduling horizon, subject to constraints (7). In other words, if the scheduling time horizon is 10 hours, and the MPC execution frequency is 1 minute, the closed-loop simulation would involve solving the MPC problem 600 times. Such simulation is performed assuming there is no model mismatch, and no disturbances affect the system. The input arguments for the simulation are the production flow rate setpoints, $F_p^{sp,n}$. Using such setpoints, the MPC problem is solved and provides values for the production flow rate F_p^i and feed air flow rate F_{feed}^i. Equation 7 is then used to calculate the product flow to inventory, $F_{inv,in}^i$, and since such equality constraint is being used within the simulation, it can be easily evaluated using a logic statement. Finally, the total

cost of operating the system is calculated through Eq. 10. Additionally, storage model and constraints that also depend on the closed-loop simulation of the system are evaluated. The optimization problem uses these simulation results to update the production setpoints using a gradient-based method. Figure 2.2 shows a schematic of the closed-loop simulation optimization framework applied to an air separation unit.

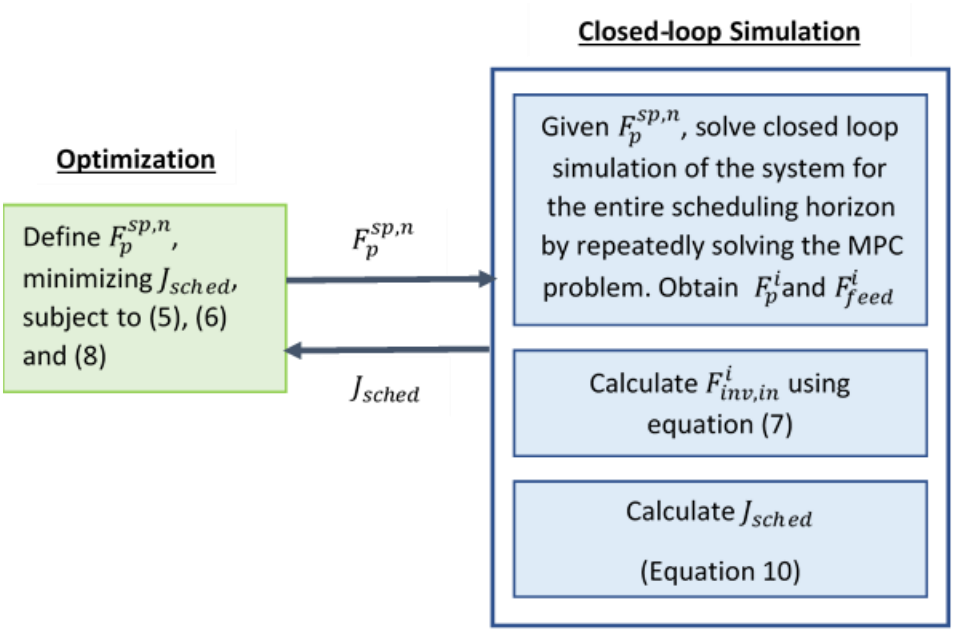

Figure 2.2 - The closed-loop simulation-optimization schematic

Simulation-optimization has become possible in the last decades due to advances in computing power and memory [51, 52]. Main approaches to solve the simulation-optimization problem include response surface methods [70, 71], gradient based procedures [72], and sample path optimization [73, 74]. In this paper, we solve the simulation-optimization problem using a sequential quadratic programming algorithm implemented in MATLAB, in which, at each major iteration, an approximation is made of the Hessian of the Lagrangian function, a quadratic programming subproblem is generated,

and it solution is used to form a search direction for a line search procedure. Hessian and gradients are approximated via finite differences.

We note that we seek locally rather than globally optimal solutions for the simulation optimization problem considered in this work. The convergence of simulation optimization problems to global optima may not be possible or efficient, especially if the simulation is computationally expensive. Conversely, local solutions often represent significant improvements over the heuristics typically used in practice and can be obtained with lower computational effort.

2.3.4.2 The inner and outer control loops

Following the work of Zhuge and Ierapetritou [37], we propose to use two loops for the scheduling and control framework. The outer loop consists of solving the integrated scheduling and control problem described in section 2.3.4.1. Since we assume accurate price forecasts are available, the outer loop is solved with a much lower frequency when compared to the inner control problem. The outer loop provides the control set points that will be tracked by an inner control problem. The inner control consists of an equivalent control strategy to the one used in the closed-loop simulation optimization problem. Additionally, other supervisory and regulatory controllers may be implemented, assuming that such controllers do not affect the scheduling relevant variables. The inner control system ensures that plant-model mismatch and process disturbances are accounted for. Figure 2.3 presents a schematic of the inner and outer control loops.

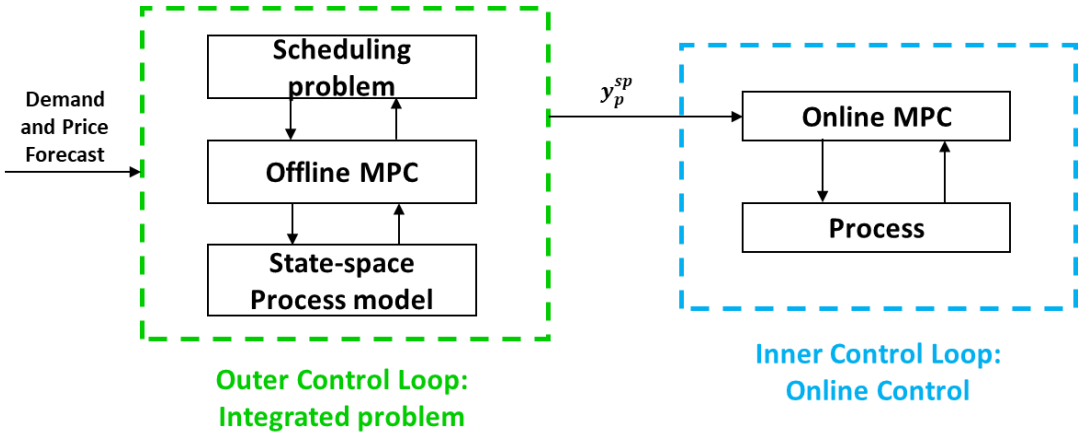

Figure 2.3 - The inner and outer control loops schematic

We note that the outer-loop problem can be optimized in a moving horizon fashion, which can be carried out periodically as new information is made available, or when disruptions to the scheduling layer affect the optimality of the current schedule. However, as observed by Subramanian, Maravelias [75], re-scheduling and moving horizon frameworks do not always yield good closed-loop scheduling solutions, especially in the presence of irregular demand. Defining the ideal scheduling optimization frequency while guaranteeing schedule stability is not straightforward [76], and is subject of ongoing work.

2.3.4.3 Interpretation from the perspective of Economic Model Predictive Control (EMPC)

We note that the problems addressed in this paper could in principle be handled by an economic model predictive controller (EMPC). EMPC has been proposed in an attempt to integrate economic process optimization and process control and incorporates a general cost function or performance index. The implementation strategy of the EMPC follows the same paradigm as conventional MPC. Specifically, EMPC is solved in a receding horizon fashion, where at every sample step the optimization problem is solved and the input

trajectory is defined over the prediction horizon N. The first control action is implemented and, at the next sampling period, the EMPC is re-solved. As in our framework, EMPC acknowledges that steady-state operation may not necessarily be the economically best operation strategy. However, the rigorous design of EMPC system that operate large-scale processes in a dynamically optimal fashion while maintaining stability (safe operation) of the closed-loop process system is challenging as traditional notions of stability may not apply to the closed-loop system under EMPC [77].

Nevertheless, there are some significant differences and benefits of the proposed framework when compared to EMPC. Here, we propose to maintain a certain level of independence for each of the two decision layers. At the control layer, MPC focuses on handling disturbances and ensuring that the process operation is stabilized, while computing the manipulated variables' trajectories over time intervals in the order of seconds or minutes. At the scheduling layer, the integrated problem is solved for a longer time horizon and with lower frequency. EMPC, on the other hand, completely integrates the two decisions and optimizes the economic performance on a time scale comparable to MPC with a time horizon comparable to the scheduling calculation. While such frequent optimization of the economic performance may seem beneficial, operational scheduling decisions usually establish the basis for scheduling of personnel, deliveries, maintenance and relations with third parties that, in a practical operation, cannot be updated frequently as they can lead to "schedule nervousness." Further, it is not clear that such optimization calculations can be completed to optimality in a short amount of time (e.g., with minute frequency).

By maintaining two decision layers, we avoid transferring restrictions associated with the online solution of the control level problem to the integrated scheduling problem. This is particularly essential when choosing to control the system using nonlinear models to represent its dynamic behavior and when extending the scheduling problem to account for discrete decisions (such as starting up and shutting down of units, and accounting for multiple and parallel lines of operation). From an EMPC perspective, addressing this problem requires the extension of the current control theory to include discrete actuators coupled with the efficient solution of an MINLP problem online. For a hybrid simulation-optimization framework, addressing this problem translates to implementing a nonlinear MPC coupled with the solution of a MINLP offline. We acknowledge that some initial efforts have been done to develop the theory of MPC with discrete actuators [78], and the inclusion of discrete decisions to the scheduling layer of the simulation-optimization framework is subject of ongoing work. Therefore, the advantage of the hybrid simulation-optimization framework lies in its independence of the online solution of an MINLP, which may be a significant obstacle given current computational power and state-of-the-art MINLP algorithms.

2.4 Case study

To verify the potential of the proposed framework, we present a realistic operational scenario and compare the results of the optimal schedule to the nominal operation of the plant. We assume that the storage tank has a capacity of 200kmol of product, and the initial inventory level is 50kmol. The energy prices for a period of 48hours are shown in Figure 2.4, and the scheduling horizon starts at 00:00h (midnight) of day 1. Knowing the energy

prices profile, we expect that inventory will deplete during the day (when energy prices are higher). Therefore, we set the minimum inventory level at the end of the scheduling horizon (M_{inv}^{min}) at a quarter of the storage capacity, or $50kmol$. Likewise, the inventory is expected to refill at night. We assume there is a constant demand of 20mol/s over the 48hour horizon. Moreover, we assume that we can accurately predict the energy price 48 hours in advance.

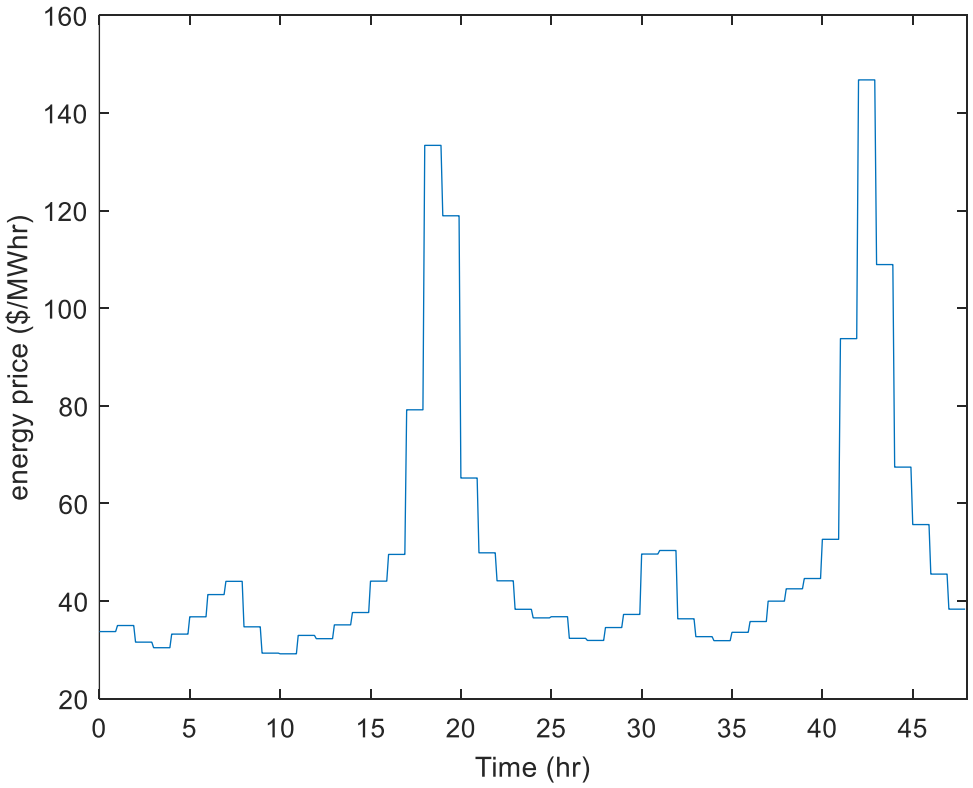

Figure 2.4 - Electricity prices

The 48-hour scheduling problem was then solved using the proposed simulation-optimization framework (Case 1). The results were obtained in 381 seconds using MATLAB R2016b on a 64-bit Windows system with Intel Core i7-6700 CPU at 2.60 GHz and 8 GB RAM. In total, during the optimization procedure, 5,052 simulation calls were made. Note that, in each simulation call, 480 MPC problems were sequentially solved (as

described in section 3.4.1), since the scheduling horizon was 48 hours and the sampling time for the MPC was 6 minutes $\left(i = \frac{T_m}{T_s} = \frac{48h \times 60}{6min}\right)$. MPC problems were implemented in MATLAB using the Model Predictive Control Toolbox. For comparison, the same scheduling problem was also solved using time scale-bridging models following closely the developments of Pattison et al. (2016) (Case 2) and using the full-order dynamic model of the plant (Case 3), also based on the work of Pattison et al., 2016. The time scale-bridging models (SBMs) are nonlinear dynamic models of Hammerstein-Wiener form that approximate the closed-loop process dynamics, and the reader is referred to Pattison et al. (2016) for details about the SBMs and the full-order dynamic model. As opposed to an online controller, the control actions for Case 2 and Case 3 are optimized offline, and follow a heuristic control law that adjusts the manipulated variables through production rate changes, as described in Pattison, Touretzky [41]. Case 2 and Case 3 were solved in 610 seconds and 59,923 seconds respectively, using a sequential dynamic optimization solver in gPROMS 5.0 (Process Systems Enterprise, 1997) on a 64-bit Windows system with Intel Core i7-2600 CPU at 3.40 GHz and 16 GB RAM.

Table 2.1 - Optimal schedule results.

Case	**Predicted Overall Cost**	**Actual Overall Cost**	**Difference from Baseline**	**Solution Time**
Baseline*	$707.91	$707.91	0%	-
Case 1 (MPC)	$697.43	$698.30	1.4%	381 s**
Case 2 (SBM)	$698.60	$698.67	1.3%	610 s
Case 3 (Full-Order)	$698.01	$698.01	1.4%	59,923 s

*****Baseline denotes the constant production rate case.

******Solution time for Case 1 refers to the offline calculations. The online MPC evaluations are practically instantaneous.

The predicted overall operational costs found in the solutions to all three optimization cases are presented in Table 2.1, along with the actual overall operational cost (calculated using the full-order dynamic model). Because it was assumed that electricity prices were known for the 48-hour horizon, deviations between the predicted and actual overall costs are only due to model mismatch between the reduced-order models and the full-order dynamic model. In our proposed framework, plant-model mismatch and disturbances are addressed by communicating schedule results with the online MPC, which tracks the setpoints over time. The MPC (as described in Section 3.2) was implemented online by using the full-order dynamic model to simulate the process and generate the sampled state variable values. The proposed integrated scheduling and MPC framework results in a lower overall operation cost compared to optimization with either the SBMs or

the full-order model, as a better and more flexible control strategy was implemented in the system.

The production rate setpoints found by the simulation-optimization framework and the actual production rate throughout the online MPC implementation are shown in Figure 2.5, along with the production rates found in the other optimization cases. The production rates closely track their predicted values from both the state-space model and the SBMs, leading to minimal deviations from the expected cost in both Case 1 and Case 2. In addition to the backoff impurity constraint, backoff temperature driving force constraint of 1.9 K was used to create the optimal schedule with SBMs (Case 2). As expected, production rates for all three optimal solutions increase when energy prices decrease, and vice versa. The temperature driving force across the reboiler/condenser limits the amount of overproduction possible, and the bound is exactly reached at some time points when the full-order model is used (Case 3). The profiles of the controller outputs from the full-order model for all three optimization cases are shown in Figure 2.6, along with any respective bounds.

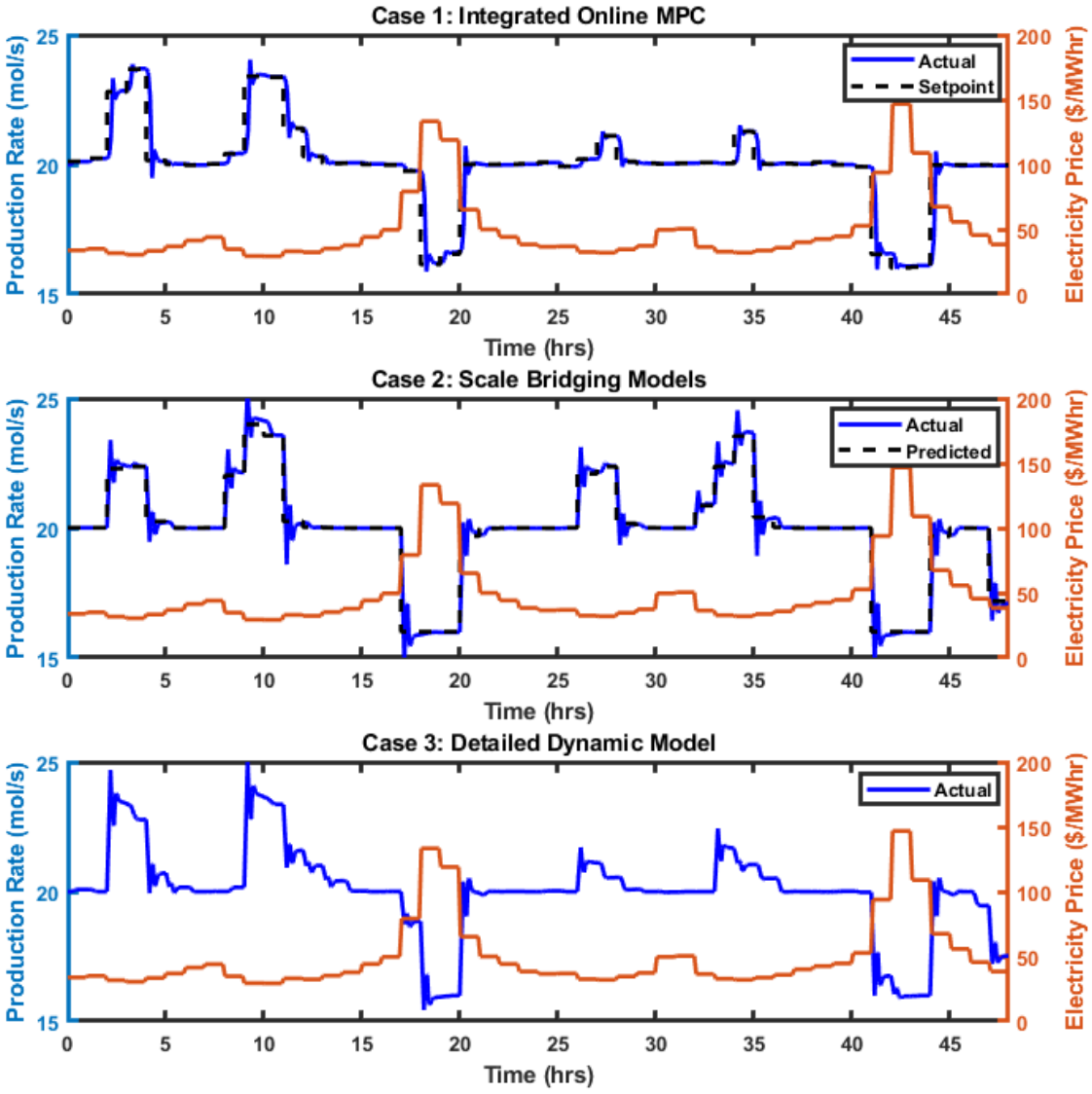

Figure 2.5 - Optimal production rates and electricity prices over time

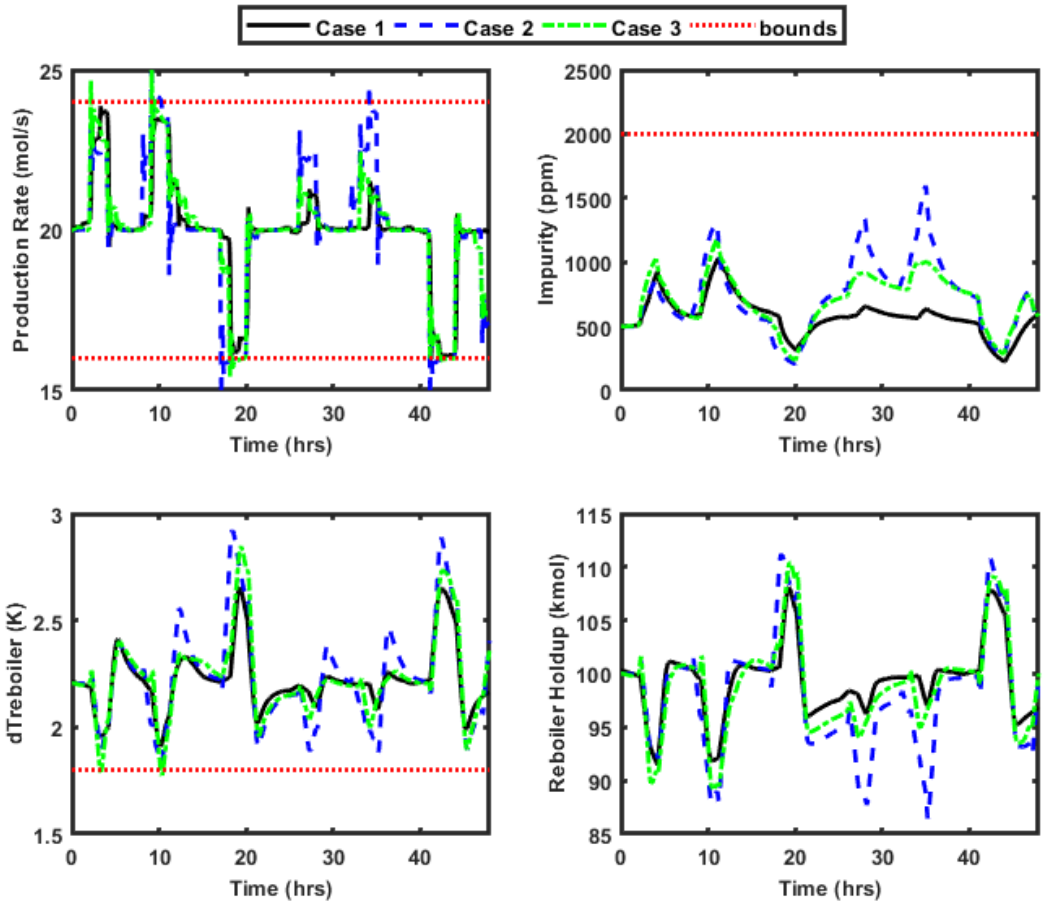

Figure 2.6 - Outputs of the control problem over time

. Displayed profiles were simulated using the full-order dynamic model

None of the variable bounds were violated in simulating the optimal schedules with the full-order dynamic model, although the back-off constraint of 1.9 K temperature driving force in Case 2 was clearly required. The prediction of reboiler holdup by the SBMs in Case 2 was also subject to some inaccuracy – the holdup was returned to only 9.95kmol at the end of the 48-hour window. Note that bounds on the production rate are imposed only for the setpoint; the actual production rates may violate the setpoint bounds. The three optimization cases presented similar economic performance: the variable production rate operation results in electricity cost savings of 1.4%, 1.3%, and 1.4% for the proposed

approach, scheduling with SBMs, and dynamic optimization with the full-order model, in comparison to a constant production profile set at the nominal rate (subject to the same variable electricity prices). These savings are substantial in the context of a well-established, commoditized industry (air separation), which is essentially a utility provider for other manufacturers. Nevertheless, optimization with the full-order model required a lengthy amount of CPU time and is difficult to implement for (typically 2-to-4-day) practical scheduling horizons. While optimization with SBMs saves a significant amount of computational effort, the scheduling calculations are subject to inaccuracies and are done offline: they must be repeated periodically if control-level disturbances are present or new information is available (Pattison et al., 2017). In contrast, the proposed method requires an offline calculation that requires less CPU time than both other described approaches, and the integration with MPC allows for disturbances to be handled online, as the MPC solves almost instantaneously after the optimal schedule is computed. Therefore, the proposed simulation-optimization approach is not only capable of achieving better performance regarding cost reduction, but improves the computational efficiency required for scheduling and re-scheduling calculations.

2.5. Conclusions

In this chapter, we propose a novel framework for the integration of scheduling and model predictive control. The framework includes the initial identification of a set of schedule-relevant variables, W. This step is followed by the identification of state space models which describes the dynamics of each variable belonging to W. We then proposed a model predictive control approach to track the corresponding varying set points over time, while

accounting for quality and process operating constraints, and integrate this control structure to the scheduling layer by solving a closed-loop simulation-optimization problem.

We demonstrate the efficacy of our framework by applying it to an ASU producing high purity nitrogen. The ASU is outfitted with a liquid nitrogen storage tank, which enables the process to adjust the production rate in response to variable electricity prices while satisfying product demand by regasifying the liquid nitrogen inventory. The optimal schedule is obtained in reasonable times and provides operating cost reductions up to 1.4% of the nominal operation cost. Furthermore, we emphasize the value of this framework by comparing it to integrated scheduling and control problems using time-scale bridging models and full-order dynamic models of the process. The proposed simulation-optimization approach is capable of achieving better performance regarding cost reduction, it improves the computational efficiency required for scheduling and re-scheduling calculations, and provides flexibility for handling online control-level disturbances and model mismatch.

The simulation-optimization framework is a novel method for the integration of scheduling and control problems, providing an easy way of implementing integrated solutions and avoiding online computation of MINLP problems. The methodology is integrated in the sense that the scheduling problem is aware of the closed-loop dynamic behavior of the system, however a certain level of independency between scheduling and control problems is maintained in order to avoid that limitations from the control level problem are transferred to the scheduling problem. However, the framework presented here has some limitations: it cannot handle discrete decisions at the scheduling problem, and the class of algorithms that can be used to solve the integrated problem is limited. Therefore,

our future work includes the investigation of alternatives for the solution of the simulation-optimization problem, including the use of metamodels. Such procedures also allow the performance of the simulations offline and/or using parallel computing, which will favor the scalability of the framework.

Nomenclature

Indices

i	Time intervals for the simulation problem
n	Time slot of the scheduling problem
k	Sample number of the MPC problem

Sets

I	Set of time intervals
N_n	Set of scheduling slots
W	Set of scheduling relevant variables

Variables

α	Split fraction
F_p^i	Product flow rate at sample period i
$F_p^{sp,n}$	Product flow rate setpoint at sample period n
F_{feed}^i	Feed air flow rate at sample period i
$F_{inv,in}^i$	Inlet flow rate to the inventory at sample period i
k_{PHX}	Split fraction in the PHX
I_p	Product purity (modeled in term of impurity concentration)
L_{drain}	Liquid drain rate in the reboiler
M_{inv}^i	Inventory holdup at sample period i
M_{reb}	Liquid level in the reboiler
R_{col}	Fraction of vapor product sent to the condenser
$\Delta T_{reboiler}$	Temperature driving force in the reboiler

u_p	Manipulated variables
u_p^{sp}	Setpoint for manipulated variables
x_p	Differential states
y_p	Outputs
y_p^{sp}	Setpoint for output variables

Parameters

$F_p^{sp,initial}$	Initial product flow rate setpoint
$F_{p,min}^{sp}$	Lower bound for the product flow rate setpoint
$F_{p,max}^{sp}$	Upper bound for the product flow rate setpoint
$\bar{F}^i$	Demand rate at sample period i
M_{inv}^0	Initial inventory holdup
M_{inv}^{min}	Minimum inventory holdup
M_{inv}^{max}	Maximum inventory holdup
N	Prediction horizon for MPC
p^i	Electricity price at sample period i
P_f	Terminal penalty matrix
Q	Output penalty matrix
R	Input penalty matrix
s	Limiting factor for changes in setpoint
τ	Scheduling time slot duration
t_{end}^n	End time for scheduling time slot n
t_{start}^n	Start time for scheduling time slot n

T_s	Sample time for the MPC problem
T_m	Scheduling horizon
u_{min}	Lower bound for manipulated variables
u_{max}	Upper bound for manipulated variables
y_{min}	Lower bound for output variables
y_{max}	Upper bound for output variables

Chapter 3

Integration of Planning and Scheduling Problems

Abstract: In this chapter, a framework for the integration of planning and scheduling using data-driven methodologies is proposed. First, the constraints at the planning level related to the scheduling problem are identified. This includes the feasibility of production targets assigned to each planning period (which are equivalent to scheduling horizons). Then, classification methods are used to identify feasible regions from large amounts of scheduling data, and an algebraic equation for the predictor is obtained. The predictor is incorporated in the planning problem, and the integrated problem is solved to optimality. Computational studies are presented to demonstrate the performance of the proposed framework, and results show that the approach is more efficient than current practices in the integration of planning and scheduling problems.

3.1 Introduction

To achieve the integration of decision-making problems, it is essential to estimate and transmit the feasible space from a lower-level problem to an upper-level. In the integration of planning and scheduling, for example, it is critical to model the feasible space of the scheduling problem, and integrate it to the planning model. However, defining the feasible space of scheduling problems is challenging: these problems are usually modeled as Mixed Integer Linear Problems (MILP), and there are no mathematical approaches to rigorously determine the feasible space of a MILP problem in its entirety.

In this chapter, a data-driven feasibility analysis methodology for the integration of planning and scheduling problems is presented. In particular, we investigate the performance of classifications methods in identifying the feasible region of a scheduling problem. Three classification methods will be investigated, namely decision trees, support vector machines and neural networks. Algebraic equations for each of the classification

predictors will be derived, and the equations will be incorporated in the planning problem. The integrated problem is then solved, and the performance of the different classification methods in the final integrated problem is assessed through computational studies. Ultimately, the goal of this work is to demonstrate that the coupling of classical optimization methods and data-driven methodologies enables the solution of problems that so far have been considered intractable. With data analytics and mathematical models, the decisions in an enterprise can be taken in a more informed manner, faster, with higher responsiveness and accommodate increasing fluctuations in demand and complexities in supply chains.

The remaining of this chapter is organized as follows. In section 3.2, well-known mathematical formulations for planning and scheduling problems are presented. In section 3.3, the problem in consideration is summarized, and section 3.4 presents the methodologies proposed to solve the integrated problem. The performance of the proposed framework is demonstrated through case studies in section 3.5. Final conclusions and future work are discussed in section 3.6.

3.2 Background

3.2.1 Scheduling Problem

Scheduling problems arise in almost all industrial sectors, including oil and gas, pharmaceuticals, chemicals, food and beverages, pulp and paper. The goal of a scheduling problem is to determine the assignment of given tasks to be processed on specified resources while minimizing costs, makespan, or maximizing profits within a horizon of days or weeks. The production must be appropriately planned to ensure that equipment,

material, utilities, personnel and other resources are available at the plant when they are needed to realize the production tasks [5]. In general, the scheduling problem can be summarized as:

Given:

i. Production network; e.g., processing units and storage vessel capacities, unit connectivity.
ii. Production recipes; e.g., processing times, rates and utility requirements
iii. Production costs; e.g., raw materials, utilities, etc.
iv. Production targets

Determine:

i. Assignment of tasks to resources (equipment units and utilities)
ii. Sequencing or tasks
iii. Timing of tasks

Traditionally, production scheduling has been done by experienced individuals, using spreadsheets and following best operating practices. However, with increasing complexities and dynamic environments, emerging from the volatility of customer orders, raw material and energy prices, it is very difficult to ensure a profitable production without any optimization support. Several optimization models have been proposed in the literature, most often expressed as mixed-integer programming (MIP) models. For extensive reviews on this subject, the reader may refer to Castro, Grossmann [79], Floudas and Lin [6] and Maravelias [7].

This work employs a discrete state-task network (STN) model for scheduling problems in batch manufacturing plants [80]. However, the frameworks proposed here can

be easily extended for other scheduling formulations and continuous processes. In the STN formulation, the scheduling horizon SH is discretized in time periods $k \in \mathbf{K}$ of duration τ, such that $SH = \tau \times K$. Binary variables X_{ijk} are 1 if task i starts in unit j at time k. Set $\mathbf{I_j}$ comprises tasks i that can be performed in unit j, and set $\mathbf{J_i}$ comprises units j that can execute tasks i. Equation (1a) then determine that only one task i can be assigned to a unit j in any time period k, and equation (1b) forbids tasks that would finish beyond the scheduling horizon. Equation (1c) is related to sequence-dependent changeovers and determines that a task i' following task i in unit j can only be initialized after an appropriate cleaning period $\sigma_{i'ij}$ has been completed. The batch size B_{ijk} of a task is restricted between the minimum V_{ij}^{min} and maximum V_{ij}^{max} capacity (Eq. 1d), while the inventory W_{sk} of state s at time k is limited by the storage capacity C_s (Eq. 1e). Mass balances are enforced by equation (1f), where ρ_{is} are stoichiometric coefficients of state s in task i related to production and consumption. Furthermore, sets $\mathbf{IC}_s$ comprises the tasks i that consume state s, and set $\mathbf{IP}_s$ comprises the tasks i that produce state s. Finally, equation (1g) determines that the production targets P_{st} for states related to finished products $\mathbf{S^{FP}}$ should be achieved by the end of the scheduling horizon. Subscript t is related to the planning problem discretization, and it will be discussed in the next section.

$$\sum_{i \in \mathbf{I_j}} \sum_{k'=k-\tau_{ij}+1}^{k} X_{ijk'} \leq 1, \qquad \forall j \in \mathbf{J}, k \in \mathbf{K} \tag{1a}$$

$$X_{ijk} = 0, \qquad \forall j \in \mathbf{J}, i \in \mathbf{I_j}, k > \mathrm{SH} - \tau_{ij} \tag{1b}$$

$$X_{ijk} + X_{i'jk'} \leq 1, \qquad \forall i, i' \neq 1, j, k, k - \tau_{i'j} - \sigma_{i'ij} < k' \leq k - \tau_{i'j} \tag{1c}$$

$$X_{ijk} V_{ij}^{min} \leq B_{ijk} \leq X_{ijk} V_{ij}^{max}, \qquad \forall j \in \mathbf{J}, i \in \mathbf{I_j}, k \in \mathbf{K} \tag{1d}$$

$$0 \leq W_{sk} \leq C_s, \qquad \forall s \in \mathbf{S}, k \in \mathbf{K} \tag{1e}$$

$$W_{sk} = W_{s,k-1} + \sum_{i \in \mathbf{IP}_S} \sum_{j \in \mathbf{J}_i} \rho_{is} B_{i,j,k-\tau_{ij}} + \sum_{i \in \mathbf{IC}_s} \sum_{j \in \mathbf{J}_i} \rho_{is} B_{ijs}\,, \qquad \forall s \in \mathbf{S}, k \in \mathbf{K} \tag{1f}$$

$$W_{sk} = P_{st}, \qquad \forall s \in \mathbf{S^{FP}}, k = \text{SH} \tag{1g}$$

3.2.2 Planning Problem

Production planning seeks to determine optimal production targets and product inventories for given demand forecasts, while considering a horizon of weeks or months. The planning horizon is usually divided in multiple planning periods with a horizon coinciding with the scheduling horizon (Fig. 3.1). The production targets are therefore defined for each scheduling horizon, and the scheduling problem is solved considering that such targets must be achieved. The planning problem may also consider multiple production and storage facilities, and simultaneously determine targets for each facility. This work, however, focus on a single manufacturing facility, for which the planning problem will be solved considering a large time horizon. The planning horizon PH is discretized into T time steps. The general planning formulation is then posed as follows:

Given:

(i) The demand D_{st} for different products $s \in \mathbf{S^{FP}}$ in each planning period $t \in \mathbf{T}$

(ii) The production and storage capacities

(iii) The production costs Cp_t associated with the different production targets P_{st}

(iv) The holding costs hc_s and unmet demand penalties uc_s associated with each product $s \in \mathbf{S}$

Determine:

(i) The production targets P_{st} for each item $s \in \mathbf{S^{FP}}$ in period $t \in \mathbf{T}$

(ii) Inventory levels Inv_{st} for each product $s \in \mathbf{S^{FP}}$ and period $t \in \mathbf{T}$

(iii)Unmet demand U_{st} for each product $s \in \mathbf{S^{FP}}$ and period $t \in \mathbf{T}$

The objective of the planning problem is to minimize total cost of operating the manufacturing facility, including inventory costs and penalty costs for unmet demand. The planning problem can be written as a mathematical optimization problem as follows:

$$\min CT = \sum_{i \in T} (Cp_t + Ch_t + Cu_t) \tag{2a}$$

$$F(P_{st}) \leq 0, \qquad \forall t, s \in \mathbf{S^{FP}} \tag{2b}$$

$$Cp_t = C(P_{st}), \qquad \forall t, s \in \mathbf{S^{FP}} \tag{2c}$$

$$Ch_t = \sum_{s \in S^{FP}} hc_s I_{st}\,, \qquad \forall t \tag{2d}$$

$$Cu_t = \sum_{s \in S^{FP}} uc_s U_{st}\,, \qquad \forall t \tag{2e}$$

$$Inv_{st} = Inv_{st-1} + P_{st} - D_{st} + U_{st}, \qquad \forall t, s \in \mathbf{S^{FP}} \tag{2f}$$

The feasibility of production targets is implicitly modeled via eq. (2b), and production cost Cp_t in period t is expressed via eq. (2c). Both equations are a function of production targets P_{st}. Holding cost Ch_t and backlog cost Cu_t are calculated in equations (2d) and (2e). Inventory and backlog flow balance constraints for product s at the end of period t are maintained by eq. (2f). It can be noted that constraints (2b) and (2c) are presented here in a general form and depend on the process network in consideration. Furthermore, it can be noted that equations (2a, 2d-2f) are linear equations. The nature of the final optimization problem will depend on the form of equations (2b) and (2c). Complex representations of these equations can bring nonlinearities and discrete decisions to the

planning problem, while simpler approximations can generate a simple linear optimization problem at the planning level.

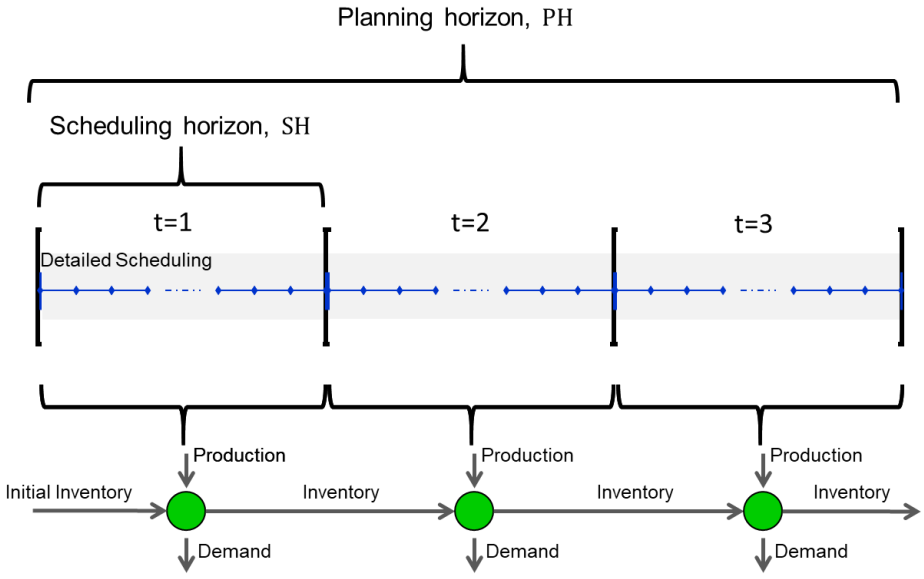

Figure 3.1 - Planning and scheduling horizons and information flow

3.2.3 Integrated Approaches

In planning problem 2, equations 2b and 2c are associated with the feasibility of production targets and cost of production, respectively. These equations depend on the characteristics of the process network and the solution of the scheduling problem. Therefore, to accurately provide feasibility and production cost information, the integration of planning and scheduling models becomes necessary.

As suggested in Maravelias and Sung [12], the integrated planning and scheduling formulations proposed in the literature can be classified into three categories: (a) detailed scheduling models, (b) relaxations/aggregations of scheduling models, and (c) surrogate models derived through off-line analysis of manufacturing facilities. The first group includes intuitive ideas such as replacing the resource and production cost constraints by a

monolithic scheduling model over the entire planning horizon. Clearly, such methods result in complex mathematical formulations that can become intractable for large time horizons and high-dimensional problems [12]. The second group tries to handle the complexity problem by removing some of the scheduling constraints [13-16], or by aggregating some of the decisions of the original scheduling formulation [17-20]. The third group proposes to use surrogate models to generate constraints that define the feasible region of the scheduling model and the production cost as function of production targets [21-24]. Such methods can generate more accurate and computationally tractable descriptions of resource and production costs constraints.

The work proposed here falls within the third category of integrated approaches, using data-driven models for the representation of scheduling information. For simplicity, production costs will be approximated by linear relationships, while strategies to define the feasibility of production targets will be extensively investigated. The basic idea behind this work relies on projecting the feasible region of a schedule onto the space of production targets. Although the work proposed in Sung and Maravelias [23] provides a conceptually attractive framework to obtain such projection when handling convex feasible regions, the extension to non-convex problems presented in Sung and Maravelias [24] is not trivial. Therefore, a more practical and easily implementable framework to obtain the feasible region of non-convex problems is developed and presented here. The framework will focus on machine learning methodologies to identify the scheduling feasible regions, and therefore it takes advantage of large amounts of historical information available in enterprises. In particular, the potential of supervised learning methods for the integration of planning and scheduling is investigated. Finally, it should be noticed that this work does

not rely on the existence of a scheduling formulation to solve the integrated problem, even though scheduling models can be used to generate data in cases where historical information is not available.

3.2.4 Feasibility analysis

The concept of feasibility analysis will play a major role in the integration of planning and scheduling problems. Given design variables d, the feasibility analysis problem attempts to determine if a process can meet all constraints f_j for all the values in the uncertain space θ by adjusting control variables z. Mathematical formulations for this problem were first proposed by Grossmann and co-workers [81, 82], and solution strategies were proposed by Halemane and Grossmann [83] and Grossmann and Floudas [84]. More recently, surrogate-based feasibility analysis was proposed by Boukouvala and Ierapetritou [85] and Wang and Ierapetritou [86]. A comprehensive review on this subject can be find in Grossmann, Calfa [53].

The work proposed here explores a new direction for the solution of the feasibility problem, interpreting feasibility analysis as a classification problem. Such interpretation will enable the use of machine learning methods to identify the feasible region of a system. These methods are usually suitable to handle large amounts of data and therefore will take advantage of all the data that has been made available in an enterprise. Although implemented in planning and scheduling case studies, we foresee that the feasibility analysis methodology developed here can be extended to other areas within the process systems engineering research.

From a conceptual point of view, the feasibility analysis of a scheduling problem is conducted by assuming that the process network, batch recipes, equipment sizes and storage capacities are the design variables; the production targets for the scheduling horizon are the uncertain parameters; and the allocation of resources and sequencing and timing of tasks are the control variables. The feasibility analysis method developed here will attempt to identify the exact region in the uncertainty space in which all scheduling constraints are respected (Fig. 3.2a). Using available sample points and its respective labels, classifiers will be derived to predict the class (feasible or infeasible) of any point in uncertainty space.

To analyze the performance of each classifier, the feasibility metrics proposed by Wang and Ierapetritou [86] are adopted in this work. The whole range of the uncertainty parameters is first divided in four regions (Fig. 3.2b): CF (Correct Feasible region) represents the overlapped feasible regions; CIF (Correct InFeasible region) represents overlapped infeasible region; ICF (InCorrect Feasible region) represents the area predicted by the classifier as feasible but in fact is a infeasible region; ICIF (InCorrect InFeasible region) represents the area predicted by the classifier as infeasible but in fact is a feasible region. With these definitions, four metrics for model accuracy are defined:

$$\begin{aligned} \mathrm{CF\%} &= \frac{\mathrm{CF}}{\mathrm{CF}+\mathrm{ICIF}} \times 100 \\ \mathrm{CIF\%} &= \frac{\mathrm{CIF}}{\mathrm{CIF}+\mathrm{ICF}} \times 100 \\ \mathrm{NC\%} &= \frac{\mathrm{ICF}}{\mathrm{ICF}+\mathrm{CF}} \times 100 \\ \text{Total Error} &= \frac{\mathrm{ICF}+\mathrm{ICIF}}{\mathrm{CF}+\mathrm{CIF}+\mathrm{ICF}+\mathrm{ICIF}} \times 100 \end{aligned} \tag{3}$$

The first two metrics (CF% and CIF%) describe how well the uncertainty parameter space has been correctly explored and classified with respect to feasibility. CF% represents the percentage of the feasible region in the original space that has been correctly identified by the classifier; and CIF% represents the percentage of the infeasible region in the original space that has been correctly identified by the classifier. The third metric, NC%, represents the percentage of feasible region that have been overestimated by the classifier. Therefore, NC% evaluates the conservativeness of the feasible region predicted by the classifier. The fourth metric, Total Error, measures the total number of misclassifications in relation to the size of the testing set. A classifier can accurately approximate the feasible region if CF% and CIF% are close to 100%, and NC% and Total Error are close to 0.

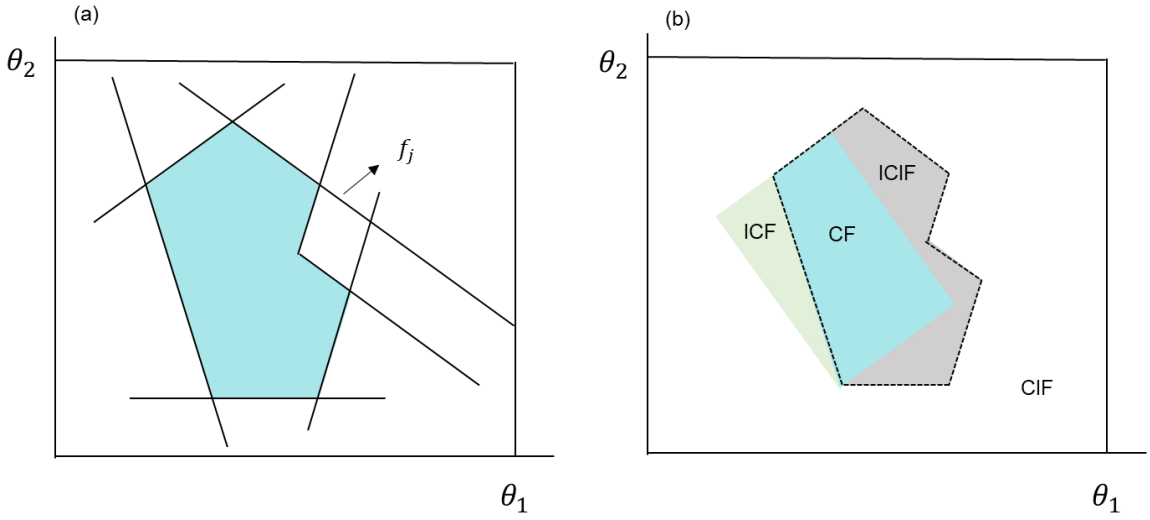

Figure 3.2 - Feasible region visualization

On the left, a feasible region in the uncertain space is represented by the blue shaded area. On the right, an illustration of the model accuracy metrics for the given feasible region.

3.2.5 Classification methods

As mentioned in the previous section, machine learning methodologies will be used to identify the feasible region of a scheduling problem. Machine learning uses three types of techniques: (i) supervised learning, which trains a model on known input and output data so that it can predict future outputs; (ii) unsupervised learning, which finds hidden patterns or intrinsic structures in input data; (iii) reinforcement learning, which attempts to maximize a reward given the current state of the system [87]. Supervised learning methods can be further classified in classification and regression techniques. While classification methods predict discrete responses, classifying input data into categories, regression techniques predict continuous responses and attempt to approximate unknown functions. In this work, we use supervised learning methods for the identification of feasible regions. In particular, we focus on three classification methods, namely decision trees, support vector machines and neural networks. Further details will be given in the next sections.

It is important to highlight that this work will not address issues of data pre-processing or feature selection, two important steps in practical machine learning implementations. We are mainly interested in understanding how the different classifiers will affect the planning optimization model. The objective of obtaining a final model for the planning model that could be solved efficiently was the main criteria on the selection of the three aforementioned classification methods in this investigation. Methods such as k-nearest neighbors or gaussian process classifiers present a complex structure for the predictor, which would likely introduce too many complexities in the planning problem, and therefore are not considered in this work.

3.3 Problem Definition

Consider the production planning problem for a single manufacturing facility operating a batch process, for which production targets P_{st} should be defined for each planning period $t \in \mathbf{T}$ in the planning horizon PH. Each planning period t corresponds to a scheduling horizon SH, which is further discretized in K time periods of duration τ. Once production targets P_{st} are defined, they are transferred to the scheduling problem, and appropriate scheduling decisions should be taken to achieve the production targets by the end of the scheduling horizon. Given are the inventory capacities, holding costs, penalties for unmet demand, and a demand forecast for the entire planning horizon. Demand is assumed to be deterministic. The planning problem is defined by Equation 2. To achieve integration with the scheduling problem and obtain an overall feasible solution, Eqs. (2b) and (2c) should be appropriately defined.

In this work, it is assumed that the production costs associated to production targets (Eq. 2c) can be represented by linear relationships between production targets and raw material consumption (Eq. 4). In Eq. 4, rc_s represents the cost of raw material $s \in \mathbf{S}^R$, where $\mathbf{S}^R$ is the subset of states related to raw materials. Furthermore, $\gamma_{ss'}$ relates the consumption of raw material $s' \in \mathbf{S}^R$ to the production of finished products $s \in \mathbf{S}^{FP}$.

$$Cp_t = \sum_{s \in \mathbf{S}^{FP}} \sum_{s' \in \mathbf{S}^R} rc_{s'} \gamma_{ss'} P_{st} \tag{4}$$

Remark: It is reasonable to approximate production costs as a linear combination of production targets if the raw material prices are the main drivers of scheduling cost. However, this assumption is not accurate in the presence of significant changeover costs, variable processing times, utility costs and dynamic utility requirements/availabilities. In

such cases, the production costs are likely to present a discontinuous relationship with the production targets, as demonstrated by Li and Ierapetritou [88]. The approximation of discontinuous function is a challenging problem, and the identification of the points of discontinuity is not trivial [89-92]. This problem however is out of the scope of this work.

The focus of this work is to obtain accurate representations of $F(P_{st})$ in Eq. 2b which describes the feasibility of production targets. If this function is approximated by taking into consideration the detailed behavior of the scheduling problem, the final planning problem is considered to be integrated with scheduling. Solutions for this problem are expected to be feasible from both planning and scheduling perspectives, i.e., the production targets set for each period of the planning problem should be achievable within the scheduling horizon.

To obtain representations of $F(P_{st})$, it is assumed that either historical information about the feasibility of different production targets is available, or a scheduling model can be used to generate data and feasibility information. In other words, we are given instances in the production space $\mathbf{x}_m = \{P_1^m, P_2^m, \dots, P_d^m\}$ (production targets) and labels y_m, $m = \{1,2, \dots M\}$ classifying each instance as feasible $[-1]$ or infeasible $[1]$. Note that d is the dimension of the problem (the number of products) and M is the number of labeled instances. The set $\mathbf{St} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots (\mathbf{x}_M, y_M)\}$ will be referred to as the training set. Using the training set $\boldsymbol{St}$, we will attempt to obtain a classifier that accurately predicts the class of any unlabeled instance in the production space. The performance of this classifier will be evaluated by comparing predicted labels and actual labels in a testing set $\mathbf{Test} = \{(\mathbf{x}_{T1}, y_{T1}), (\mathbf{x}_{T2}, y_{T2}), \dots, (\mathbf{x}_{TN}, y_{TN})\}$. This data will be used to calculate the feasibility metrics $\mathrm{CF\%}, \mathrm{CIF\%}$ and $\mathrm{NC\%}$ (Eq. 3). Finally, an algebraic equation for this

classifier will be obtained, and this equation will be integrated in the planning problem. The nature of the final planning model (LP, NLP, MILP) will depend on the nature of the classifier. A summary of the proposed framework is shown in Figure 3.3.

In the case studies presented in this paper, no historical information about the feasibility of production targets is available. Therefore, the training and testing sets will be generated by first using Latin Hypercubic Sampling to define instances in the production space. Each instance (or each production target) will be fed to a scheduling problem in the form of Equation 1, with a dummy objective ($\text{minimize}\, obj = 0$). The solution of this problem will determine if a production target is feasible or infeasible. The training and testing sets are then built using these results, finalizing step 1.

In the next section, the different classification methods and the algebraic form of the predictors (steps 2 and 4 in the proposed framework) will be discussed in detail.

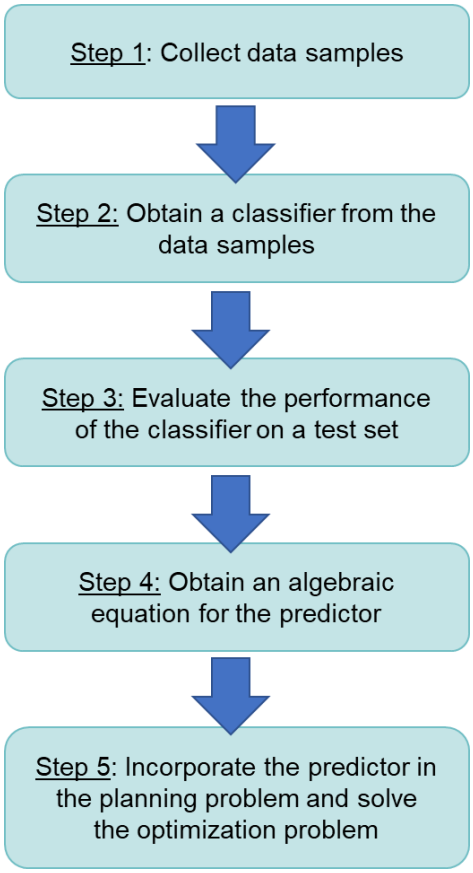

Figure 3.3 - Data driven feasibility analysis framework

3.4 Frameworks for the integration of planning and scheduling

In this work, five different methodologies will be used for the integration of planning and scheduling problems. The first three methodologies are the contributions of this work and follow the framework shown in Figure 3.3. For each methodology, a different classification method will be used to estimate the feasible region of a scheduling problem, and the derived classifiers will be incorporated in the planning model. The proposed methodologies will be described in sections 4.1 to 4.3. The fourth and fifth methodology will be used for comparison purposes and are described in section 4.4.

3.4.1 Integration using decision trees

The first methodology for the integration of planning and scheduling problems uses decision trees to identify the feasible region of a schedule. Decision trees are a non-parametric supervised learning method that creates a model to predict the target variable by learning simple and logical decision rules. In a decision tree, instances are classified based on feature values. Each node in a decision tree represents a feature in an instance to be classified, and each branch represents a value that the node can assume. Instances are classified starting at the root node and sorted based on their feature value [87].

Consider, for example, the decision tree in Figure 3.4. In this example, each instance is formed by two features, x_1 and x_2. At the root node, the decision tree checks if the feature x_2 in an instance is less or equal to 6.23. If so, the decision path will follow the left branch, and the next node will evaluate if $x_1 \leq 18.46$. On the other hand, if $x_2 > 6.23$, the decision path follows the right branch, and the next node evaluates if $x_1 \leq 6.06$. The procedure is repeat until a leaf node is reached, where a label $[-1,1]$ will be assigned for the given instance.

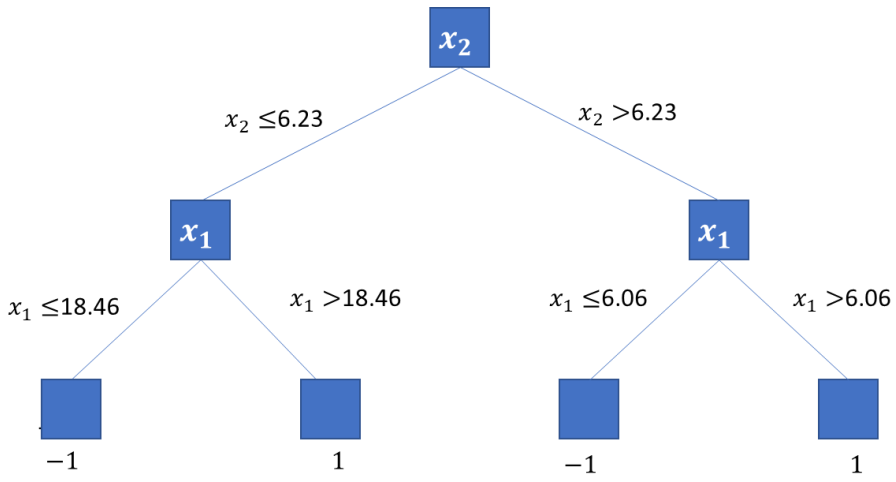

Figure 3.4 - A decision tree

Decision trees are simple to understand and to interpret, but they can be unstable (small variations in data can generate completely different trees) and are prone to overfitting. In this work, decision trees will be generated in python using the library scikit-learn [93]. This library follow the CART algorithm [94] to generate the trees. In this algorithm, given the training vectors $\mathbf{x}_m \in \mathbf{R}^d$, $m = \{1,2,\dots,M\}$ and label vector $\mathbf{y} = \{y_1, y_2, \dots, y_M\}$, the design space is recursively partitioned such that samples with the same labels are grouped together. The algorithm for generating the tree follows:

Let the data at node at the parent node be represented by N_P. Each candidate split $\theta = (j, t_{split})$ consisting of a feature j and threshold t_{split} partitions the data into $N_L(\theta)$ and $N_R(\theta)$ subsets:

$$\begin{aligned} N_L(\theta) &= (\mathbf{x}, y)|x_j \leq t_{split} \\ N_R(\theta) &= N_p \backslash N_L(\theta) \end{aligned} \tag{5a}$$

The improvement (gain) $I(\cdot)$ generated by a split of a parent node N_p, into left ($N_L(\theta)$) and right children ($N_R(\theta)$) is given by Eq. (5b).

$$I(N_p) = G(N_p) - qG(N_L) - (1-q)G(N_R) \tag{5b}$$

where q is the fraction of instances going left and $G(\cdot)$ is a measure of impurity. Common measures of impurity include the Gini measure, given by:

$$G(N) = 1 - p(N_m)^2 - \big(1 - p(N_m)\big)^2 \tag{5c}$$

where $p(N_m)$ is the relative frequency of class 1 in node N_m.

The algorithm to generate the decision tree will then select parameters θ that maximize the improvement I at node N_p, and recurse for nodes N_L and N_R until the maximum allowable depth of the tree is reached, the minimum number of instances in a node is reached, or until all the instances in a leaf node belong to the same class.

A decision tree for the training set **St** is generated using the python library skicit-learn. Once the tree is generated, the performance of the tree is evaluated by predicting labels of a testing set **Test**, and comparing the predictions with the actual labels. The overall prediction error is calculated, as well as metrics $\mathrm{CF\%}, \mathrm{CIF\%}$ and $\mathrm{NC\%}$. If the performance of the tree is satisfactory, an algebraic equation for the predictor is derived.

To derive the algebraic form of the predictor, we first note that the value of -1 will be attributed to feasible points. Then, we define the decision rules (or the paths) that will end in a leaf node of class -1. The combination of paths leading to feasible regions can be written using logic operators. For example, for the decision tree in Fig. 5 we have:

$$Y_1 = (x_1 \leq 18.46)$$

$$Y_2 = (\, x_2 \leq 6.23\,)$$

$$Y_3 = (x_1 \leq 6.06)$$

$$(Y_1 \wedge Y_2) \vee \big(Y_3 \wedge (\neg\, Y_2)\big) \tag{6a}$$

The logic equation can be rewritten in the *conjunctive normal form*, in which the logic statement should be represented by a conjunction of clauses $Q_1 \wedge Q_2 \wedge \ldots \wedge Q_s$ (i.e., connected by and operators $\wedge$), where the clauses correspond to logic propositions that only involve the OR operator. For details, see [95]. The following is equivalent to Eq. (6a):

$$(Y_1 \vee Y_3) \wedge (Y_2 \vee Y_3) \wedge \big(Y_1 \vee (\neg Y_2)\big) \tag{6b}$$

Equation (6b) then substitutes equation (2b) on the planning problem. The resulting integrated problem corresponds to a Disjunctive Programming Model that can be re-written as a Mixed Integer Linear Problem (MILP), and solved using GAMS/CPLEX.

3.4.2 Integration using support vector machines

The second methodology for the integration of planning and scheduling problems uses support vector machines (SVM) to identify the feasible region of a schedule. Support vector machines attempt to find an optimal hyperplane that separates all data points of one class from those of the other class. Given M training points in $\mathbf{R}^d$, $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots \mathbf{x}_\mathrm{M}\}$, each point is associated with one of two classes characterized by a value of $y_m = \pm 1, m = 1, \ldots M$. If the training data is linearly separable, then a pair $(\mathbf{w}, b)$ exists such that:

$$\begin{aligned} &\mathbf{w}^T\mathbf{x}_m + b \geq +1, \text{ for all } \mathbf{x}_m, y_m = +1 \\ &\mathbf{w}^T\mathbf{x}_m + b \leq -1, \text{ for all } \mathbf{x}_m, y_m = -1 \end{aligned} \tag{7a}$$

The SVM problem translates to finding an affine function $g(\mathbf{x}) = \mathbf{w}\mathbf{x} + b$ that classifies the points, i.e., $y_m(\mathbf{w}\mathbf{x}_m + b) \geq 1$ and $y_m \in \{-1,1\}$. The optimal hyperplane $\mathbf{w}\mathbf{x} + b = 0$ is the unique one which separates the training data with a maximal margin. In other words, it determines the direction $\mathbf{w}/|\mathbf{w}|$ where the distance between the projections of the training vectors of two different classes is maximal. It can be shown [96] that the optimal separating hyperplane is found by minimizing the squared norm of w:

$$w^* = \min \frac{1}{2}\left|\left|\mathbf{w}\right|\right|^{\mathbf{2}} \tag{7b}$$

$$s.t. y_m(\mathbf{w}\mathbf{x}_m + b) \geq 1$$

By deriving the Lagrangian of this problem, we obtain the dual function:

$$\max_{\alpha} \sum_m \left[\alpha_m - \frac{1}{2}\sum_n \alpha_m \alpha_n y_m y_n (\mathbf{x}_m \cdot \mathbf{x}_n)\right] \tag{7c}$$

$$s.t.\begin{cases}\sum_m \alpha_m y_m = 0 \\ \alpha \geq 0\end{cases}$$

The resulting problem is a convex optimization problem, which is solved with respect to α. The solution of this problem enables the computation of $\boldsymbol{w}^*$ and b (Eq. 7d), and therefore a predictor in the form $f(x) = sign(\mathbf{w}^T\mathbf{x} + b)$ is obtained.

$$\begin{aligned}\mathbf{w}^* &= \sum_m \alpha_m y_m x_m \\ b &= \frac{1}{N_{sv}} \sum_{m \in SV} (y_m - \mathbf{w} \cdot \mathbf{x}_m)\end{aligned} \tag{7d}$$

where N_{SV} refers to the number of support vectors, which are the sample points that lie on the plane $y_m(\mathbf{w}\mathbf{x}_m + b) = 1$.

If the data points are not linearly separable due to existence of outliers or misclassified instances, the problem can be addressed by using a soft margin concept. This is done by introducing positive slack variables $\xi_m,\ m = 1, \ldots M$ in the constraints, as follows:

$$\begin{aligned}&\mathbf{w}^T\mathbf{x}_m + b \geq +1 - \xi_m, \text{ for all } \mathbf{x}_m, y_m = +1 \\ &\mathbf{w}^T\mathbf{x}_m + b \leq -1 + \xi_m, \text{ for all } \mathbf{x}_m, y_m = -1 \\ &\xi_m \geq 0\end{aligned} \tag{7e}$$

In this case, the dual optimization problem becomes:

$$\max_\alpha \sum_m \left[\alpha_m - \frac{1}{2}\sum_n \alpha_m \alpha_n y_m y_n (\mathbf{x}_m \cdot \mathbf{x}_n)\right]$$
$$s.t.\begin{cases}\sum_i \alpha_m y_m = 0 \\ 0 \leq \alpha \leq C\end{cases} \tag{7f}$$

Nevertheless, most real-problems involve non-separable data for which no hyperplane exists that successfully categorizes the data. We therefore seek to find a transformation of the input space that maps the data to a space Z, and construct an optimal separating hyperplane in this space. We call $\mathbf{z}_m = \phi(\mathbf{x}_m)$, where $\phi(\cdot)$ is a transformation function. By substituting $\mathbf{x}_m$ by its transformation in Eq. (7f), we obtain the following problem:

$$\max_{\alpha} \sum_{m} \left[\alpha_m - \frac{1}{2} \sum_{n} \alpha_m \alpha_n y_m y_n (\phi(\mathbf{x}_m) \cdot \phi(\mathbf{x}_n)) \right]$$
$$s.t. \begin{cases} \sum_{i} \alpha_m y_m = 0 \\ 0 \leq \alpha \leq C \end{cases} \tag{7g}$$

Therefore, in order to solve the SVM problem for data mapped into Z space, we need to define the dot product $\phi(\mathbf{x}_m) \cdot \phi(\mathbf{x}_n)$. We do not, however, need to map each point $\mathbf{x}_m$ to Z space. The dot product $\phi(\mathbf{x}_m) \cdot \phi(\mathbf{x}_n)$ can be defined using results from the theory of reproducing kernels. The theory states that there is a space Z and a function ϕ mapping $\mathbf{x}$ to Z such that $K(\mathbf{x}_1, \mathbf{x}_2) =< \phi(\mathbf{x}_1), \phi(\mathbf{x}_2) >$, where $K(x_1, x_2)$ is a class of functions denominated Kernel Functions. The dot product takes place in the Z space. Examples of kernel functions include the radial basis function (Eq. 7h), which is used in this study.

$$K(\mathbf{x}_1, \mathbf{x}_2) = \exp(-\gamma \left|| \mathbf{x}_1 - \mathbf{x}_2 \right||^2) \tag{7h}$$

The kernel-based support vector machine translates to computing matrix $K(\cdot,\cdot)$ for the training set and solving the quadratic optimization problem in (7g). The classification function (eq. 7i) can be used to classify untested points.

$$f(\mathbf{x}) = sign\left(\sum_{\alpha_v>0} \alpha_v y_v K(\mathbf{x}_v, \mathbf{x}) + b\right), \qquad v = 1, \dots, N_{SV} \tag{7i}$$

For any support vector $\alpha_v > 0$.

Note that the predictor (7i) takes the form of a non-convex nonlinear function. We will identify this predictor from the training data **St** in our problem, evaluate the performance of the predictor using metrics $\mathrm{CF\%}, \mathrm{CIF\%}$ and $\mathrm{NC\%}$, and incorporate this predictor in the planning optimization problem. Equation (2b) takes the form:

$$\sum_{\alpha_v>0} \alpha_v y_v K(\mathbf{x}_v, \mathbf{x}) + b \leq 0 \tag{7j}$$

and the resulting optimization problem corresponds to a nonlinear nonconvex optimization problem, that is solved to local optimality using GAMS/CONOPT.

3.4.3 Integration using neural networks

The third methodology for the integration of planning and scheduling problems uses artificial neural networks (ANN) to identify the feasible region of a schedule. A neural network is an interconnected group of nodes (Fig. 3.5). Considering a neural network with a single hidden layer as shown in Figure 3.5, the neural network will take inputs, multiply each input by a weight $\mathbf{w}_{input} \in \mathbf{R}^{1\times N}$ and add a bias term $\mathbf{b}_{input} \in \mathbf{R}^{1\times N}$, where N is the number of nodes in the hidden layer. The results will feed a hidden layer, where an activation function is applied to the net input. In this work, the activation function has the form of the logistic sigmoid function, $f(x) = 1/(1 + \exp(-x))$. Each node in the hidden layer generates an output which in its turn will be multiplied by weights $\mathbf{w}_h$ and bias $\mathbf{b}_h$ to generate the output layer. The training of the neural network then consists in determining

weights and bias values that minimize the mean squared error between predicted outputs and the actual output of the system.

When training a neural network, the weights and bias are initially set to random values. Then, a forward pass on the neural network calculate predicted outputs, and a loss function is computed to prediction errors. Next, weights are updated using a gradient descent step, i.e.,

$$w \leftarrow w - \eta \left(\frac{\alpha \partial R(w)}{\partial w} + \frac{\partial Loss}{\partial w} \right) \tag{8a}$$

where η is a learning rate which controls the step-size in the parameter search space, $Loss$ is the loss function used for the network, and R is the predictor. The procedure is repeated until a certain number of iterations has been reached, or when the improvement in loss is below a certain threshold.

Neural networks have an inherent ability to learn and approximate complex and nonlinear functions. There are efficient identification algorithms for the estimation of parameters in the neural network, and neural nets usually have a simple structure for the predictor. However, different random weights initializations can lead to different accuracies, and neural networks require tuning a number of hyperparameters such as the number of hidden neuros, layers and iterations.

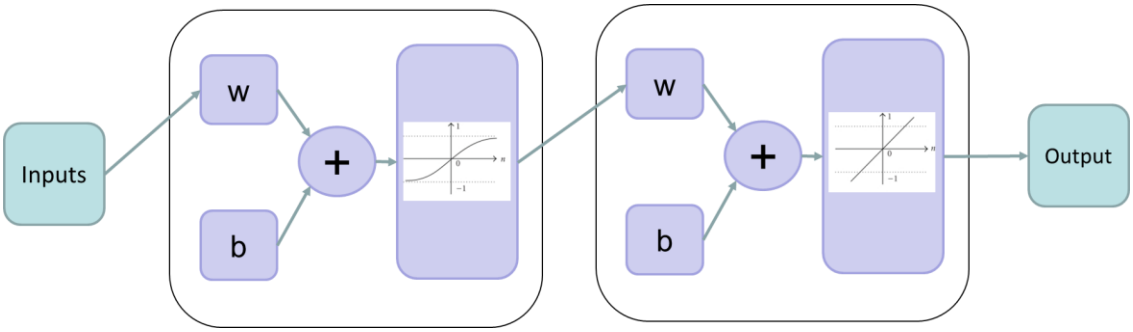

Figure 3.5 - A neural network representation

In this work, a neural network is identified for training set **St** using the scikit-learn library in python. Once the neural network is trained, the performance of the predictor is evaluated using metrics CF%, CIF% and NC% for a testing set **Test**. The predictor of a neural network with one hidden layer, a single output and logistic activation function takes the form of equation 8b. The predictor is then incorporated in the planning optimization problem, substituting equation 2b by Eq. 8c. Note that we enforce the prediction to be less than 0 since feasible points are classified as -1 and infeasible points are classified as 1.

$$f(\mathbf{x}) = \mathbf{w}_h \times \left(\frac{1}{1 + \exp\left(\mathbf{w}_{input} \times \mathbf{x} + \mathbf{b}_{input}\right)} \right) + b_h \tag{8b}$$

$$\mathbf{w}_h \times \left(\frac{1}{1 + \exp\left(\mathbf{w}_{input} \times \mathbf{x} + \mathbf{b}_{input}\right)} \right) + b_h \leq 0 \tag{8c}$$

The resulting planning problem corresponds to a nonlinear nonconvex optimization problem that is solved to local optimality using GAMS/CONOPT.

3.4.4 Benchmark methodologies

Two methodologies are used in the integration of planning and scheduling for comparison purposes. First, a detailed scheduling model is solved for the entire planning horizon, and this methodology is referred as "Detailed scheduling for PH". This problem provides the overall optimal solution for the integrated problem. However, the resulting integrated problem corresponds to a large-scale MILP problem, which can become intractable when handling large-dimensional problems.

The second methodology consists on obtaining an over estimation of the feasible region by solving several scheduling problems that maximize the production of final

products s in given "directions" w_s^i, $i = \{1,2,3, \dots M\}$. In other words, problem (1) is solved with an objective given by Eq. 9a:

$$\max_{\mathrm{P_s}} z = \sum_s w_s^i P_{st} \tag{9a}$$

where $t = \{1\}$. The solution of this problem results in feasible production targets P_{st}^i and an OE inequality in the form:

$$OE^i = \sum_s w_s^i P_{st} \leq \prod^i \tag{9b}$$

where $\prod^i$ is the best upper bound of problem 1. If problem 1 is solved to optimality, then the best upper bound is the optimal objective value z. The methodology follows an iterative approach to define directions w_s^i, and the final set of OE^i inequalities is incorporated in the planning problem. This methodology will be referred as "Integration with OE inequalities", and the reader should refer to the work by Sung and Maravelias [23] for details on the algorithm. This methodology represents the state-of-the-art in the integration of planning and scheduling, and the resulting integrated problem will be a linear programing model that can be solved very efficiently. However, inaccuracies in the approximation of non-convex feasible regions may result in poor integrated solutions.

3.5 Case studies

In this section, three case studies will be solved following the five methodologies presented in section 3 and 4. First, a simple two-dimensional case will be presented to illustrate all the steps. Then, a three-dimensional problem and a more challenging seven-dimensional problem will be investigated.

3.5.1 Two-dimensional problem

Consider the process network shown in Figure 3.6. Raw materials RMA and RMB are converted by two parallel units into products A and B. The two units can operate at up to 1kg/h ("slower" unit) and 3kg/h ("faster" unit), respectively. For both units, 2h of cleanup between products A and B are required. We aim to define the feasible region of a 6-hour scheduling problem, discretized in hourly periods, and integrate planning and scheduling problems considering a 30-day planning horizon. We assume that demand forecasts for the entire planning horizon are available. Product recipes, unit capacities and inventories capacities are given. The objective of the planning problem is to define production targets for each 6-hour period in the planning horizon. The demands, capacities, holding costs and penalty costs for this problem are given in Appendix A.

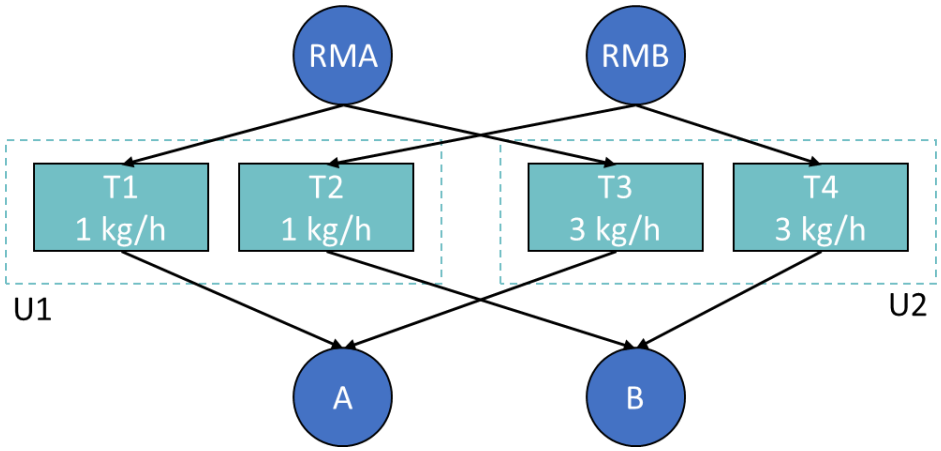

Figure 3.6 - Process network for the scheduling problem

To generate "historical data", an MILP model for the scheduling problem in the form of equation (1) is defined, and the problem is solved for different production targets $\mathbf{P}^i, i = \{1,2,\dots,m\}, m = 100$. The production targets are generated by sampling the production space using Latin Hypercube Sampling. Then, each target is labeled as feasible or infeasible ($y^i = -1$ and $y^i = 1$, respectively), according to the results of the MILP, and

a training set $\mathbf{S} = \{(\mathbf{P}^1, y^1), (\mathbf{P}^2, y^2), \ldots (\mathbf{P}^m, y^m)\}$ is obtained. The data from the training set is standardized, and three different classifiers are generated. Then, the classifiers are used to predict the labels of a testing set containing 3600 points in the production space, and the performance of the classifiers is evaluated using the feasibility metrics CF%, CIF% and NC%. Furthermore, an over-estimation of the feasible region is also generated according to the methodology "Integration with OE inequalities". The performance of this over-estimation is also evaluated, and the results are shown in Table 3.1 and Figure 3.7. In table 2, the CPU time required to generate the classifier for each novel methodology is mainly related to the generation of the training set, and therefore would show significantly lower values if historical information was available. The computational time for pre-processing the data (standardization) and training the classifiers is in the order of milliseconds. Figure (3.7a) was generated using an additional set of 3600 points obtained by a grid sampling strategy, and the feasibility of each sample (or production target) was verified by solving a scheduling problem in the form of Equation 1, with a dummy objective (minimize $obj = 0$). Finally, Figure 3.8 shows the structure of the generated decision tree. Note that P_A refers to product A and P_B refers to product B in Fig. 3.8.

All classifiers were trained using scikit-learn library in python and default configurations.

Table 3.1 - Performance of different classifiers in the prediction of the feasible region.

	CF (%)	**CIF (%)**	**NC (%)**	**Total Error (%)**	**CPU(s) (sampling)**	**CPU (s) (training)**
OE	100	76.71	29.18	14.88		3.67
Decision Tree	89.37	99.25	1.47	4.32	12.86	0.001
SVM	91.25	98.75	2.37	3.96	12.86	0.001
ANN	94.13	98.75	2.30	2.92	12.86	0.009

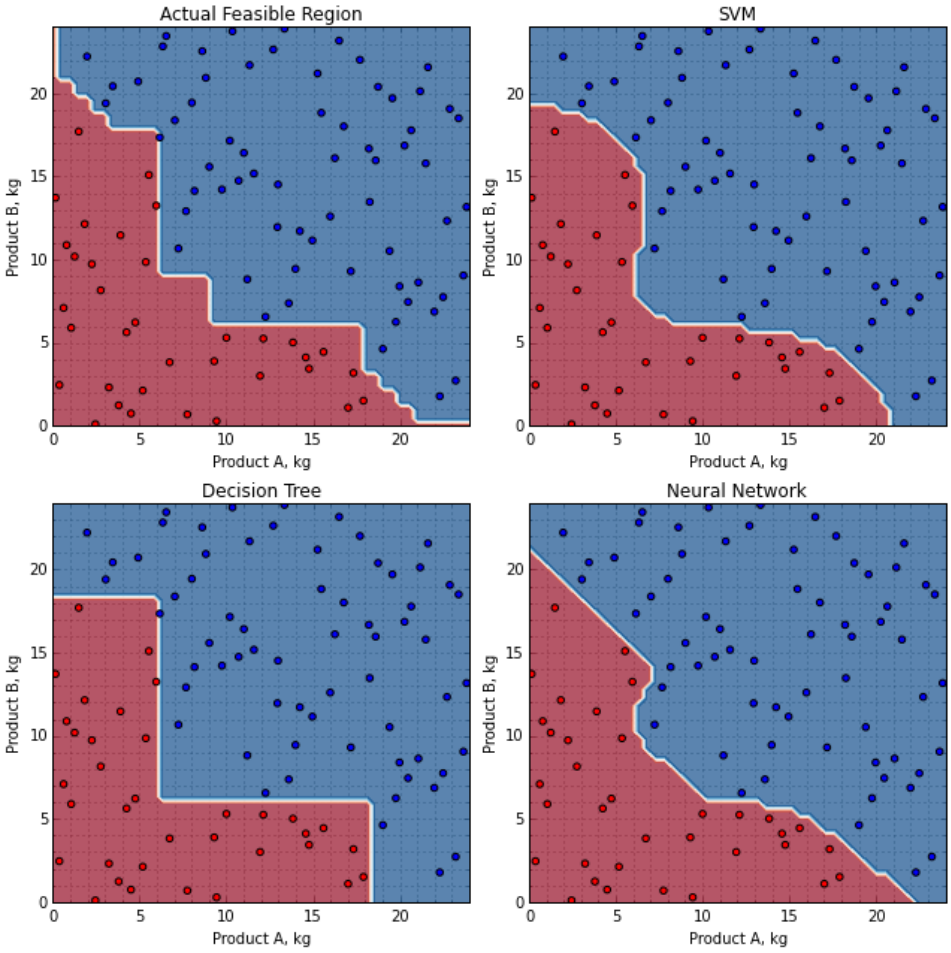

Figure 3.7 - Performance of different classification methods.

Red represents the feasible region, and blue represents infeasible regions. On the top left, the actual feasible region. On the top right, the feasible region predicted by SVM. On the bottom left, the feasible region predicted by decision trees. On the bottom right, the feasible region as predicted by neural networks.

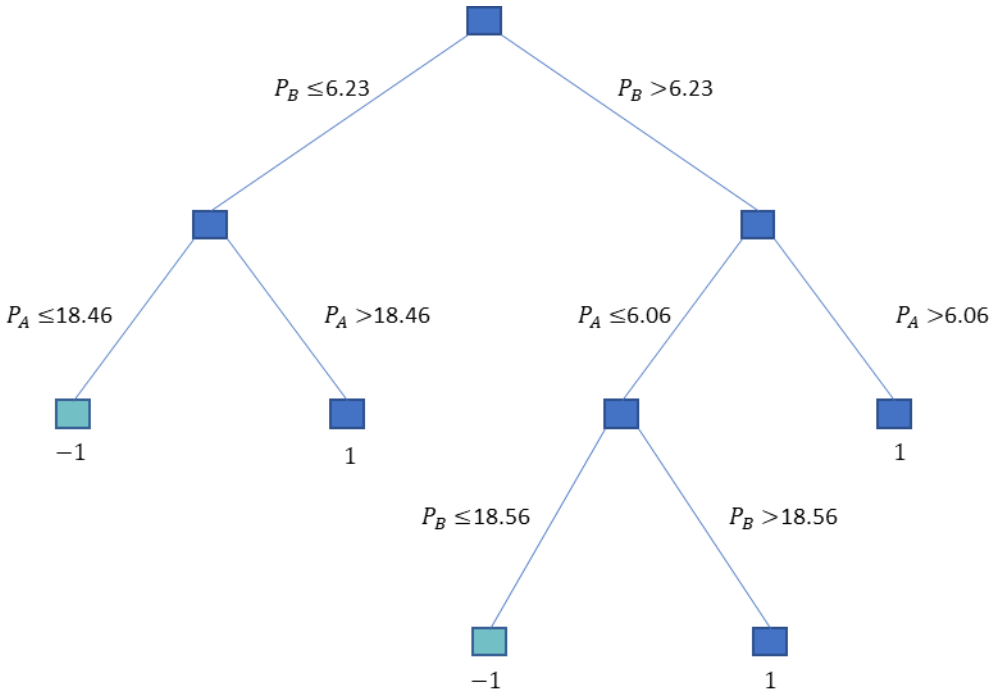

Figure 3.8 - Decision tree for the study in consideration

As seen in Figure 3.7, all classification methods misclassified several points in the feasibility boundaries. However, they present an overall good performance in identifying the feasible region. Therefore, we proceed to integrate the predictors with the planning problem and solve the integrated problem to local optimality. The explicit constraints representing the decision tree, SVM, and neural network predictors are given by Eqs. (10), (11) and (12) respectively

$$P_{At} \leq 18.46 \tag{10a}$$

$$P_{Bt} \leq 18.56 \tag{10b}$$

$$P_{At} \leq 6.06 + M(1 - y_{1t}) \tag{10c}$$

$$P_{Bt} \leq 6.23 + M(1 - y_{2t}) \tag{10d}$$

$$y_{1t} + y_{2t} = 1 \tag{10e}$$

$$\sum_{\alpha_v>0} \alpha_v y_v \exp(-\gamma \left|\left|\mathbf{x}_v - \mathbf{P}_t\right|\right|^2) + b \leq 0 \tag{11}$$

$$\mathbf{w}_h \times \left(\frac{1}{1 + \exp\left(\mathbf{w}_{input} \times \mathbf{P}_t + \mathbf{b}_{input}\right)} \right) + b_h \leq 0 \tag{12}$$

Where $\mathbf{P}_t$ is the vector of production targets at planning period t, i.e., $\mathbf{P}_t = \{P_{1t}, P_{2t}\}$. Note that Eq. 10 was obtained following the procedure described in Section 3.4.1, and the disjunctions were translated to algebraic constraints using Big-M formulation [95]. Therefore, y_{1t} and y_{2t} are binary variables in this formulation. The support vector predictor (Eq. 11) was formed by 10 support vectors, and the values for α_v, y_v, x_v, b and γ are given in Appendix B. The neural networks were trained using 1 hidden later and 10 hidden nodes, and the values for $\mathbf{w}_h$, $\boldsymbol{w}_{input}$, $\mathbf{b}_h$ and $\mathbf{b}_{input}$ are given in Appendix B.

Once the solutions (production targets) for each planning period are obtained, the targets are transferred to the scheduling model, which solves equation (1) with a modification in equation (1h) and the following objective function:

$$\min \sum_s (hc_s \times h_s + uc_s \times u_s)$$

$$W_{sk} = P_s + h_s - u_s, \qquad \forall s \in \mathbf{S^{FP}}, k = \text{SH} \tag{13}$$

Therefore, we solve the scheduling problem attempting to achieve the production targets P_s, penalizing any excess of production h_s and unmet production targets u_s.

The scheduling problems for the entire planning horizon will be solved in a sequential manner. We highlight here that all state information at the end of one scheduling horizon (final inventories, current usage of units and last processing task in each unit) will be transferred to the following scheduling problem. Therefore, we can enforce changeover constraints between different scheduling periods with appropriate linking constraints.

The integrated problem is also solved using the two benchmark methods, and the results are shown in Table 3.2.

Table 3.2 - Integrated planning and scheduling results

Strategy	Holding Cost	Unmet demand	Production cost	Total cost	CPU(s)
Detailed scheduling for PH	$ 649.78	$ 0.00	$ 7,234.19	$ 7,883.97	10
Integration with OE ineq.	$ 0.00	$ 9,495.21	$ 7,115.50	$ 16,610.71	~0
Integration with decision tree	$ 1,087.95	$ 2,317.38	$ 7,205.23	$ 10,610.55	0.03
Integration with SVM	$ 1,334.00	$ 2,515.99	$ 7,202.76	$ 11,051.75	0.29
Integration with ANN	$ 616.05	$ 2,721.65	$ 7,200.17	$ 10,538.87	0.08

The results for the integrated problem show how the proposed approach for the integration of planning and scheduling may fail to find the global optimal solution. This could be a result of poor approximations of the feasible region, the result of obtaining a local optimum in the solution of the NLP problem, or the result of the choice of features in the feasibility problem.

It should be also noticed that we assumed throughout this entire work that the feasible space was a function of the production targets. However, production targets are not the only variables influencing the feasible space. The initial state of the system (the last task performed on each unit, for example) will also affect the feasibility of a schedule. The significance of this effect in the final integrated problem will depend on scheduling horizon, process network and the relation between demand and operating capacity of the plant. These complex relationships will be subject of future work.

Nevertheless, the novel integrated problem was solved in reasonable time and is expected to scale well with the problem dimensionality. Due to the combinatorial nature of

MILP, the detailed scheduling model size grows exponentially with problem dimensionality, which is not the case of the nonlinear programming models proposed here. The proposed approach also demonstrates great potential when compared to OE inequalities approach. Although they result in lower computational times in the solution of the integrated problem, the integration with OE inequalities will result in several infeasible production targets that will be penalized at the scheduling level, generating large penalties for unmet demands.

3.5.2 Three-dimensional problem

In this section we consider a three-dimensional case study adopted from [23] that uses the process network shown in Figure 3.9. Six processing units and two raw materials are used to produce products A, B and C through 10 different tasks. The cleanup time between different tasks in unit 1, unit 5 and unit 6 is 1 hours. The cleanup time between different tasks in unit 4 is 2 hours. A 24h scheduling horizon and 60-day planning horizon are considered. The scheduling problem is discretized in hourly intervals. Relevant data for this problem is presented in Appendix A. We aim to solve the integrated planning and scheduling problem for given demand forecasts, determining production targets for each planning period of 24h.

To solve the integrated approach, we first generate historical data by solving the MILP scheduling problem for 1000 different production targets $\mathbf{P}^i, i = \{1,2, \dots 1000\}$, and obtaining a feasible/infeasible classification. We form the training set $\mathbf{S} = \{(\mathbf{P}^1, y^1), (\mathbf{P}^2, y^2), \dots (\mathbf{P}^m, y^m)\}$, standardize the data, and generate three classifiers for this data, including a decision tree, a support vector machine and a neural network. All

classifiers were trained using scikit-learn library in python and default configurations. Then, the classifiers are used to predict the labels of a testing set containing 1000 points in the production space, and the performance of the classifiers is evaluated using the feasibility metrics CF%, CIF% and NC%. Furthermore, an over-estimation of the feasible region is also generated according to the methodology "Integration with OE inequalities". The performance of this over-estimation is also evaluated, and the results are shown in Table 3.3. The computational times shown in Table 3.3 for the three novel methodologies are mainly related the generation of the training set, since the training of classifiers was performed in the order of milliseconds. Finally, Figure 3.10 shows the structure of the generated decision tree. The labels for each ramification of the decision tree have been omitted, as the sole purpose of Fig. (3.10) is to illustrate the structure of the tree.

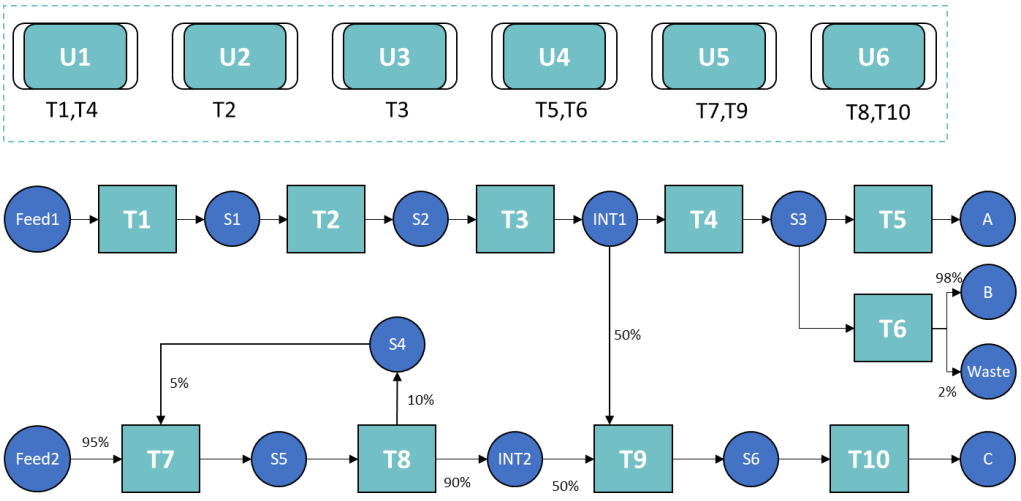

Figure 3.9 - Process network for the scheduling problem

Table 3.3 - Performance of different classifiers

	CF (%)	CIF (%)	NC (%)	Total Error (%)	CPU(s) (Sampling)	CPU(s) (Training)
OE	100	96.41	6.02	2.30		12.08
Decision Tree	93.31	95.16	8.47	5.5	150	0.003
SVM	98.89	97.19	4.82	2.2	150	0.018
ANN	98.60	97.50	4.32	2.1	150	0.13

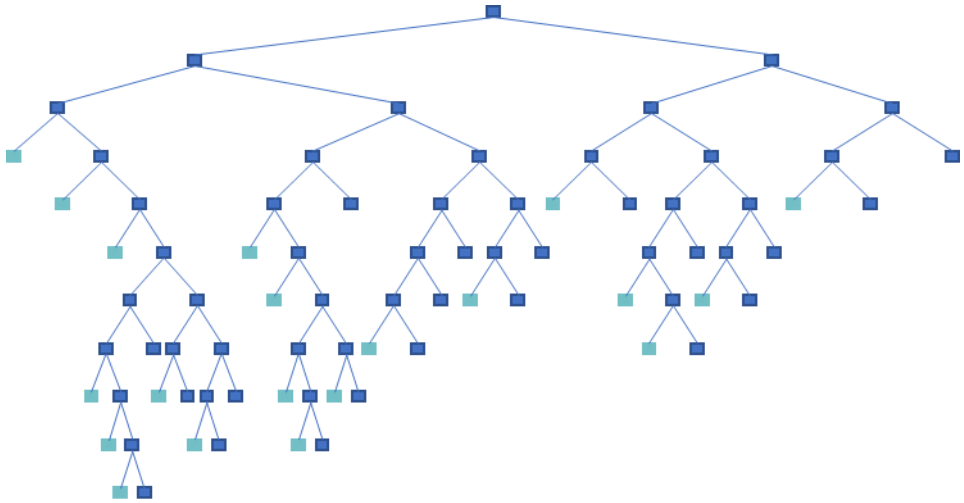

Figure 3.10 - Decision tree for the study in consideration

In this study, we note how a small change on dimension can affect the structure of the decision tree. As the complexity of the decision tree increase, it becomes more troublesome to obtain the disjunctive equations to be integrated in the scheduling problem. Furthermore, the performance of the decision tree according to the feasibility metrics is much inferior to the performance of SVMs and Neural Networks. In general, decision trees are not expected to perform well in highly nonlinear and complex problems [87].

Therefore, we will not integrate decision trees and planning problems for this case study. The integrated results for SVMs and Neural Networks, as well as the two benchmark methodologies are shown in Table 3.4. Once again, the costs for the proposed frameworks are lower when compared to the OE inequality methodology, but still fail in finding the global solution obtained with the detailed scheduling problem.

Table 3.4 - Integrated planning and scheduling results

Strategy	**Holding Cost***	**Unmet demand***	**Production cost***	**Total cost***	**CPU (s)**
Detailed scheduling for PH	$ 0	$ 0	$ 47.80	$ 47.80	17.23
Integration with OE ineq.	$ 3.18	$ 4.42	$ 47.74	$ 55.35	0
Integration with SVM	$ 3.56	$ 2.09	$ 47.78	$ 53.43	20.19
Integration with ANN	$ 3.42	$ 2.09	$ 47.78	$ 53.28	0.66

*Values in thousands of dollars.

3.5.3 Seven-dimensional problem

In this case study we target a more realistic network with many products and a multistage structure, such as the ones found in the food and chemical industries. This problem was adopted from [23] and involves a process network PN1 (Figure 3.11), which has six processing units and seven final products. In the first stage, orders (batches) passing through units U1 or U2 take 2 hours. In the second stage, orders passing through unit U3 take 1 hour. In the third stage, the processing time is 2 hours for tasks on unit U4 and 1 hour for tasks on units U5 and U6. The cleanup time between different tasks performed in the same unit is 1 hour. Product A can be produced by task T7 or T8. Products E and F are produced by different tasks but involve the same precursor on the same unit. Relevant data

for this problem is provided in Appendix A. For the integrated scheduling and planning problem, a 52-week horizon will be considered, with scheduling horizon of 168 hours. The scheduling problem is discretized in hourly intervals. Given demand forecasts, our goal is to determine production targets for each planning period by solving the integrated problem.

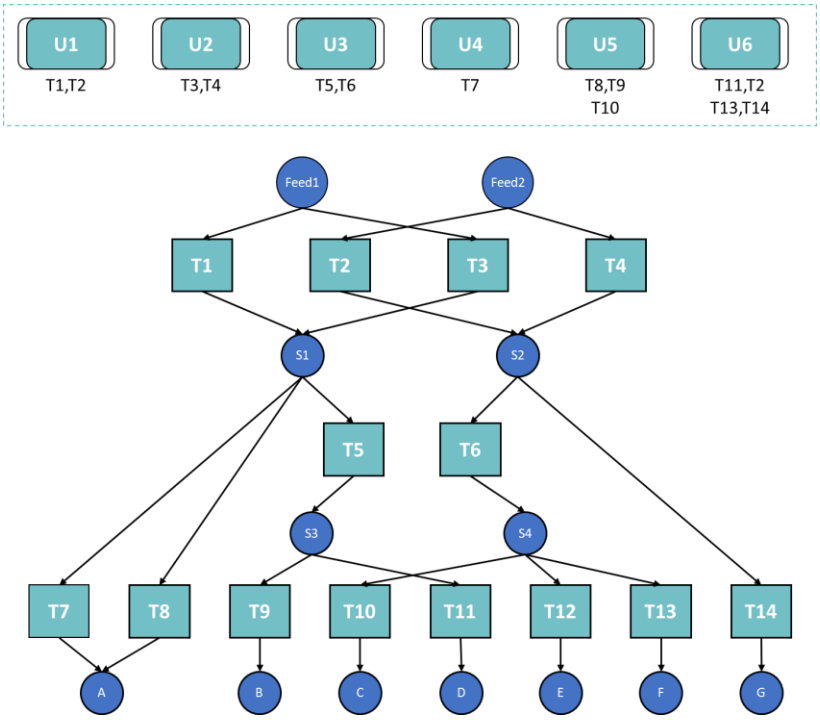

Figure 3.11 - Process network for the scheduling problem

To solve this problem, 5000 production targets were generated, the feasibility of each production target was verified, and the final training set was formed. The training set was then used to build classifiers, and the performance of the classifiers evaluated in a 1000 points testing set is shown in Table 3.5. The predictors for the SVM and Neural Networks were then incorporated in the planning problem, and the integrated problem was solved to local optimality. The solutions were transmitted to a scheduling problem, which attempts to achieve the production targets within the scheduling horizon. The final results for the

integrated problem as well as the solution of benchmark methodologies are shown in Table 3.6.

Table 3.5 - Performance of different classifiers in the prediction of the feasible region

	CF (%)	CIF (%)	NC (%)	Total Error (%)	CPU (sampling)	CPU (s) (training)
Decision Tree	74.09	66.83	23.80	28.89	~11 hours	0.184
SVM	93.44	91.34	6.07	7.43	~11 hours	4.883
ANN	93.78	91.09	6.22	7.33	~11 hours	2.345

Table 3.6 - Integrated planning and scheduling results

Strategy	Holding Cost*	Unmet demand*	Production cost*	Total cost*	CPU(s)
Detailed scheduling for PH	$ 0	$ 40,57	$ 0	$ 40,574	600
Integration with OE ineq.	$ 0,253	$ 1,461	$ 0,473	$ 2,187	0.03
Integration with SVM	$ 0,791	$ 0,043	$ 0,496	$ 1,330	60.19
Integration with ANN	$ 0,458	$ 0,712	$ 0,485	$ 1,655	0.31

*In million dollars.

In this study, the detailed scheduling model fails to provide reasonable solutions within a resource limit of 600 seconds. In fact, the detailed scheduling model only found a better solution than the SVM model after 8500 seconds. After 10 hours, the best solution found corresponded to a total cost of $ 0.861 MM, and the solution still presented a 9.23% optimality gap. The potential of the integrated problems using classification methods becomes clear in this study, and it is expected to show even more benefits as dimensions increase.

3.5 Conclusions

In this work, a framework for the integration of planning and scheduling problems was presented. The solution of the integrated problem determined production targets for each planning period. The goal of the integrated approach was to provide solutions (production targets) that were feasible from both planning and scheduling perspectives. The framework then consists on using classification methods to define the feasible region of a scheduling problem projected onto the product space from large amounts of historical data, and incorporating the built classifiers in the planning problem. Three different classification methods were investigated, namely decision trees, support vector machines and neural networks. The nature of the final integrated problem depended on the nature of the classifier; a MILP problem was obtained when integrating planning and decision tree classifiers; an NLP problem was obtained when integrating SVM or Neural Networks and planning problems. The integrated problems were solved for three different case studies, and the solutions were compared to other integrating methodologies found in the literature. As the dimension of the case-studies increase, the proposed framework generated better and more efficient solutions when compared to other methodologies. Ultimately, the framework shows the potential of machine learning and big-data analytics on the solution of integrated problems. Furthermore, the proposed feasibility analysis methodology can be implemented to a variety of problems in the process systems engineering, and it will be implemented on the integration of scheduling and control in the next chapter.

Perhaps the main drawback of this framework is an approximation error that can still be considered large, even for the 7-dimension problem. Practitioners should evaluate the trade-off between the possibility of finding a better solution when using the monolithic

approach, the computational time that this approach entails, and the practicality that an approximated integrated solution offers. To further prove the value of the proposed framework, future work should include the solution of multi-facility planning problems.

This work did not address the issue of feature selection. Throughout this work, it was assumed that the feasibility of the scheduling problem was a function of the production targets, a straightforward assumption based on ad-hoc knowledge. Nevertheless, the case studies demonstrated that other features may impact the feasible region of the scheduling problem, which will be mainly related to the initial state of the system. The significance of this impact will depend on the scheduling horizon, process network and relationships between demand and operating capacity of the plant. Future work should investigate how feature selection methodologies can capture the effects of different features in the feasible space and improve the performance of classifiers.

Future work should also focus on the different activation functions and kernels that can be used in neural networks and support vector machines, respectively. The choice of kernels and activation functions will normally fall within the context of model selection in machine learning. Nevertheless, the use of the predictors in optimization problems defines a new criterion for model selection, since the complexity of the predictor affects the complexity of the final planning problem. In our case studies, for example, the solution of the integrated problem using neural networks was consistently more efficient than the integration with SVM. Therefore, an in-depth investigation of the structure of the classifies and how they affect the complexity of the nonlinear optimization problem should be conducted.

Finally, future work should address the solution of the integrated problem considering uncertainties in demand and uncertainties in the feasible region. More robust classifiers can be built by manipulation of parameters or by simple considerations when evaluating the predictor. The work could investigate how robust decisions can impact the final operation of a manufacturing facility.

Nomenclature

Sets

$i \in \mathbf{I}$	Processing tasks
$j \in \mathbf{J}$	Units
$k \in \mathbf{K}$	Scheduling time points/periods
$s \in \mathbf{S}$	States
$t \in \mathbf{T}$	Planning time points/periods
m	Set of data points used to generate classifiers

Subsets

$\mathbf{I}_j$	Set of tasks i that can be performed in unit j
$\mathbf{J}_i$	Set of units j that can performed task i
$\mathbf{IC}_s$	Set of tasks i that consume state s
$\mathbf{IP}_s$	Set of tasks i that produce state s
$\mathbf{S}^{\mathbf{FP}}$	Set of states related to finished products
$\mathbf{S}^{\mathbf{R}}$	Set of states related to raw materials

Scheduling Variables

B_{ijk}	Batch size of task i processed in unit j starting at time point k
X_{ijk}	Binary variable which is 1 if task i is processed in unit j starting at time point k
W_{sk}	Inventory level of state s at time point k

Scheduling Parameters

C_s	Storage capacity for state s
P_{st}	Production targets for state s
SH	Scheduling horizon
V_{ij}^{min}	The minimum capacity for unit j processing task i
V_{ij}^{max}	The maximum capacity for unit j processing task i
ρ_{is}	Stoichiometric coefficients related to production/consumption of state s in task i
τ	Scheduling discretization time

Planning variables

Ch_t	Holding cost at period t
Cp_t	Processing cost at period t
Cu_t	Unmet demand penalty at period t
Inv_{st}	Inventory levels for product s in period t
P_{st}	Production targets for state s in planning period t
U_{st}	Unmet demand U_{st} for each product s and period t

Planning Parameters

D_{st}	Demand for product s in planning period t
hc_s	Holding cost for state s
uc_s	Unmet demand penalty for product s
rc_s	Cost of raw material s
$\gamma_{ss\prime}$	Stoichiometric coefficients related to production of state s and raw material s'
PH	Planning horizon

Data Set

$\mathbf{x}_m$	Vector of production targets used to train the classifiers
$\mathbf{X}$	The set of vectors $\mathbf{x}_m$
y_m	Label that classifies vector $\mathbf{x}_m$ in feasible or infeasible
$\mathbf{y}$	Label vector, $\mathbf{y} = \{y_1, y_2, \dots, y_M\}$ used to train the classifiers
d	Number of products (features) in the planning problem
M	Number of data points used to obtain classifiers
$\mathbf{St}$	Training set
$\mathbf{Test}$	Testing set

Decision Trees variables

$G(\cdot)$	Measure of impurity
j	Features, or the components of vector $\mathbf{x}^m$
t_{split}	A threshold for partitioning the data
$I(\cdot)$	Improvement gain
N_p	Data at the parent node
N_L	Data at the left node
N_R	Data at the right node
$p(N_m)$	Relative frequency of class 1 in node N_m
q	Fraction of instances going left
θ	Candidate split

Support Vector Machines variables

b	Bias
N_{sv}	Number of support vectors
v	Set of support vectors
$\mathbf{w}$	Weight
α	Lagrangian multiplier
ξ	Slack variables

Neural Network variables

$\mathbf{b}_{input}$	Bias in the input layer
$\mathbf{b}_h$	Bias in the output layer
$Loss$	Loss function
N	Number of nodes in the hidden layer
R	Predictor
$\mathbf{w}_{input}$	Weights in the input layer
$\mathbf{w}_h$	Weights in the output layer

Chapter 4

Integration of Scheduling and Robust Model Predictive Control Using Surrogate Models

Abstract: In order to achieve optimal operational conditions, the integration of decision-making across different layers of a company and the consideration of uncertain parameters in view of dynamic market conditions are essential. In this study, we propose a framework for the integration of scheduling and control for nonlinear systems under process uncertainties. The proposed approach includes the use of a tube-based robust model predictive control to handle disturbances affecting the control layer of the problem; the use of classification methodologies to determine the feasible space of operation of the process; and the use of surrogate models to derive the closed-loop input-output behavior of the dynamic system. Case studies are utilized to illustrate the performance of the proposed framework and evaluate the impact of control-level disturbances in scheduling solutions.

4.1 Introduction

Dynamic market conditions and increasing pressure for competitive performance in the process industries have led to significant efforts in the pursuit of optimal operation conditions. In addition to improved process designs, these circumstances have spurred the development of strategies aimed at vertical integration and coordination of decision-making across all the layers of the chemical supply chain. Several strategies have been proposed in the literature to achieve the integration of scheduling and control, but they are usually limited to small dimensional problems [28]. Furthermore, there are few studies about the integration of scheduling and control problems under uncertainties [4], an issue that is a very important concern in the daily operation of real plants.

In a manufacturing facility, disturbances such as flow and rate temperature variations, stream quality fluctuations and dynamic model mismatches can impact the

process operation and considerably affect the feasibility of a scheduling solution if they are not taken into account in the integrated scheduling and control problem. Some attempts to incorporate uncertainty considerations in the integrated problem includes the work by Mitra, Gudi [97], who proposed to analyze the trade-off between profit maximization and reliability of solutions in view of parametric uncertainties; and the work by Patil, Maia [98], who proposed to quantify the worst-case variability in controlled variables via frequency response analysis in the problem of design, scheduling and control integration. A thorough review on the integration of scheduling and control under uncertainties is given by Dias and Ierapetritou [4].

In this chapter, a novel framework for the integration of scheduling and model predictive control under process uncertainties is proposed. First, we implement deterministic and robust model predictive control strategies to continuous production processes that are affected by disturbances in their operation. Then, we simulate the system operation while varying the production setpoints and generate sets of data that mimic historical data usually available in production sites. Each data point holds the information of whether any quality and production constraints were violated within a determined time period and determined production setpoints. Next, we use classification methods to approximate the feasible region of operation of a system as a function of production setpoints. The classification predictors, coupled with surrogate models that capture the closed loop behavior of the system, are incorporated in the scheduling problem, and the integrated problem is optimized. In this work, we explore the use of neural networks as surrogate models, due to its inherent capability of predicting highly nonlinear functions

and handling high-dimensional problems, two characteristics that are critical when deriving surrogate models from large sets of industrial data.

The remaining of this chapter is organized as follows. In section 4.2, a background in uncertainty, model predictive control formulations, and neural networks is provided, building the theoretical basis of the proposed framework. In section 4.3, the problem in consideration is summarized, and section 4.4 presents the methodology proposed to integrate scheduling and control problems. The performance of the proposed framework is demonstrated through case studies in section 4.5. Final conclusions and future work are discussed in section 4.6.

4.2 Background

4.2.1 Uncertainties in process operation

The nature of uncertainties, its sources, and the appropriate method to deal with them has been a topic of discussion by scientists for a long time [99]. In practice, it is useful to categorize the uncertainties within a problem and systematically address each category according to its nature and its importance to the problem. In the context of integrated scheduling and control using surrogate models, we adapt the classification of uncertainties proposed by Pistikopoulos [100] and identify three main sources of uncertainties:

(i) Uncertainties related to the *control level*, including model-inherent uncertainty (such as kinetic constants, physical properties, mass/heat transfer coefficients); and process-inherent uncertainty (such as flow rate and temperature variations)

(ii) Uncertainties related to the *scheduling level*, including external uncertainties (such as product demands, raw material and utilities costs, and product prices);

and discrete uncertainties (such as equipment breakdown and personnel absence)

(iii) Uncertainties related to the inaccuracies of the surrogate models, which will be derived to represent the closed-loop behavior of the control level in the integrated scheduling and control problem.

Furthermore, uncertainties can be modeled in different ways based on the availability of information. Li and Ierapetritou [101] classify the models of uncertainties as:

(i) Bounded description, where upper and lower bounds are defined for an uncertain parameter representing the range of all possible realizations of the parameter

(ii) Probability description, where uncertainties are characterized by a probability distribution function

(iii) Fuzzy description, where fuzzy sets are used to model the uncertainty

Finally, approaches for dealing with uncertainties in scheduling can be classified as preventive or reactive [101]. In reactive approaches, solutions for scheduling and control problems are based on nominal models, and are updated in response to the occurrence of uncertainties. Preventive approaches, on the other hand, incorporate the model of uncertainty in the scheduling and control formulations and generate robust solutions prior to the occurrence of a disturbance.

This work addresses uncertainties at the control level of the integrated problem. It assumes the only description available about the uncertainties are their lower and upper bounds, and it takes a preventive approach to handle the uncertainties. More specifically,

robust optimization is employed at the control problem in order to handle uncertainties and avoid constraint violations.

4.2.2 Model Predictive Control

The basic idea of model predictive control (MPC) is to use a dynamic model to forecast system behavior and select control actions that will drive the system to a desired state. At each control interval, the MPC is implemented by: (1) measuring the current state of the system, $\mathbf{x}_{initial}$; (2) setting future control inputs $\mathbf{U} = \{\mathbf{u}^0, \mathbf{u}^1, \dots, \mathbf{u}^{N-1}\}$ for a prediction horizon N by solving an online optimization problem, where tracking errors are minimized; (3) applying the first element of the optimization solution, $\mathbf{u}^0$, to the system and going back to step 1 at time $k = k + 1$.

The MPC optimization problem for a nonlinear dynamic system can be formulated as:

$$\min_{\mathbf{U}=\{\boldsymbol{u_k}\}_{\boldsymbol{k=1}}^{\boldsymbol{N}}} J_{std}(\mathbf{x}_0, \mathbf{U}) = \frac{1}{2}\sum_{k=0}^{N-1}\left(|\mathbf{x}_k - \mathbf{x}^{sp}|_Q^2 + |\mathbf{u}_k - \mathbf{u}^{sp}|_R^2\right) + \left|\mathbf{x}_N - \mathbf{x}_N^{sp}\right|_{P_f}^2 \qquad (1)$$

$$s.t. \begin{cases} \mathbf{x}_0 = \mathbf{x}_{initial} \\ \mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k) \\ \mathbf{x}_k \in \mathbb{X}, \forall k \\ \mathbf{u}_k \in \mathbb{U}, \forall k \end{cases}$$

where k is a nonnegative integer denoting the sample number, which is connected to time by $time = k\Delta$ and Δ is the sample time. N is the prediction horizon, $\mathbf{x}^{sp} \in \mathbb{R}^n$ and $\mathbf{u}^{sp} \in \mathbb{R}^m$ are the setpoints, and Q and R, are tuning parameters penalizing deviations from the setpoint and control moves, respectively. We allow the final state penalty to have a different weighting matrix P_f, for generality. Finally, $\mathbf{U}$ is the input sequence for $N-1$ time steps, i.e., $\mathbf{U} = \{\mathbf{u}^0, \mathbf{u}^1, \dots, \mathbf{u}^{N-1}\}$. We assume throughout this paper that the state is

perfectly measured and the outputs of the system $\mathbf{y}_k$ are modeled as $\mathbf{y}_k = I\mathbf{x}_k$, where I is the identity matrix. The outputs $\mathbf{y}_k$ are omitted in all system representations.

An important feature of MPC is its ability to cope with constraints on input and state variables, and therefore MPC has been applied to a variety of industries where satisfaction of constraints is particularly important to increase efficiency and meet safety requirements [9]. In general, input constraints are modeled as hard constraints and state constraints are modeled as soft constraints.

If uncertainties affect the dynamic system, a robust MPC strategy might be more successful in tracking setpoints and avoiding constraint violations. Robust MPC was first proposed by Campo and Morari [102], who considered the possibility of model mismatch and assumed that the system behavior could be described by a set of linear time invariant (LTI) models instead of a single LTI. They proposed to minimize the worst-case tracking error for the family of linear plants, and showed how to recast the resulting minimax optimization problem as a linear program. This approach was further extended by Allwright and Papavasiliou [103] and Zheng and Morari [104], and can be classified as open-loop robust MPC formulation. Open-loop MPC solves optimization problems in which the decision variable is a sequence of control actions, and ignores the fact that the controller will react to the uncertainty in the next steps, which may lead to infeasibilities and conservative solutions. Therefore, closed-loop robust MPC was proposed by Kothare, Balakrishnan [105] and Lee and Yu [106], and the optimization problem was solved for a control policy, which is a sequence of control laws, overcoming the drawbacks of open-loop robust MPC.

While closed-loop MPC (or feedback MPC) is superior to standard MPC in the presence of uncertainty, the associated optimal control problem is vastly more complex than the optimal control problem employed in deterministic MPC [107]. Hence a lot of research effort has been devoted to different forms of feedback MPC that sacrifice optimality for simplicity. Tube-based MPC falls within this category of problems [108, 109]. Tube-based MPC takes into consideration both open-loop and feedback control responses, in the presence of uncertainty, a bundle or *tube of trajectories*, where each trajectory corresponds to a particular realization of the uncertainty. The control of uncertain systems can therefore be viewed as the control of tubes rather than trajectories. Tube-based MPC therefore requires a modified form of the online control problem in which the constraints are tightened in order to constrain the trajectories of the uncertain system to lie in a tube centered on the nominal trajectories. The reader is referred to Rawlings and Mayne [107] for a comprehensive discussion of the theory and stability of tube-based MPC.

For a system with bounded additive disturbance, $\mathbf{w} \in \mathbb{W} \subset \mathbb{R}^n$, satisfying the following difference equation:

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k) + \mathbf{w} \tag{2}$$

in which $\mathbf{x}_k \in \mathbb{X}$ and $\mathbf{u}_k \in \mathbb{U}$, and the sets $\mathbb{X}$ and $\mathbb{U}$ are polyhedral. Let the nominal system be described by:

$$\mathbf{z}_{k+1} = f(\mathbf{z}_k, \mathbf{v}_k) \tag{3}$$

The algorithm for the implementation of tube-based MPC can be summarized as follows [107]:

Initialization. At time 0, set $i = 0, \mathbf{x}_0 = \mathbf{x}_{initial}$ and $\mathbf{z}_0 = \mathbf{x}_0$. Solve $\overline{\mathbb{P}}_N$ for N time steps to obtain the nominal closed-loop state and control sequences $\mathbf{V}^* = \mathbf{V}^*(z_0) =$

$\{\mathbf{v}_0^*(z_0), \mathbf{v}_1^*(z_0), \ldots, \mathbf{v}_{N-1}^*(z_0)\}$, and $\mathbf{Z}^* = \mathbf{Z}^*(\mathbf{z}_0) = \{\mathbf{z}_0^*(\mathbf{z}_0), \mathbf{z}_1^*(\mathbf{z}_0), \ldots, \mathbf{z}_{N-1}^*(\mathbf{z}_0)\}$, and set $\mathbf{u} = \bar{\kappa}_N(\mathbf{z}_0) = \mathbf{v}_0^*(z_0)$.

Step 1 (Compute control): At time i, compute $\mathbf{u} = \kappa_N(\mathbf{x}_0, \mathbf{z}_0)$ by solving $\mathbb{P}_N(\mathbf{x}_0, \mathbf{z}_0)$.

Step 2 (Control): Apply $\mathbf{u}$ to the system being controlled.

Step 3 (Update x): Set $\mathbf{x}_{initial} = \mathbf{x}_{k+1}$ where $\mathbf{x}_{k+1} = f(\mathbf{x}_0, \mathbf{u}) + \mathbf{w}$ is the successor state

Step 4 (Update z, $\mathbf{v}^*$ and $\mathbf{z}^*$): Compute $\mathbf{v}^* = \bar{\kappa}_N(\mathbf{z}_N^*)$ and $\mathbf{z}^* = f(\mathbf{z}_N^*, \mathbf{v}^*)$ by solving $\mathbb{P}_N(\mathbf{z}_N^*)$. Set $\mathbf{z} = \mathbf{z}^*(1)$. Set $\mathbf{V}^* = \{\mathbf{v}_1^*, \ldots, \mathbf{v}_{N-1}^*, \mathbf{v}^*\}$ and set $\mathbf{z}^* = \{z_1^*, \ldots, z_N^*, z^*\}$.

Step 5 (Repeat): Set $i = i + 1$. Go to step 1.

Problems $\overline{\mathbb{P}}_N$ and $\mathbb{P}_N$ are given by Eq. (4) and (5), respectively.

$$\overline{\mathbb{P}}_N(z_0) = \min_{\mathbf{V}=\{\mathbf{v}_k\}_{k=1}^N} \frac{1}{2}\sum_{k=0}^{N-1}\left(|\mathbf{z}_k - \mathbf{z}^{sp}|_{\bar{Q}}^2 + |\mathbf{v}_k - \mathbf{v}^{sp}|_{\bar{R}}^2\right) + \left|\mathbf{z}_N - \mathbf{z}_N^{sp}\right|_{\bar{P}_f}^2 \tag{4}$$

$$s.t. \begin{cases} \mathbf{z}_0 = \mathbf{z}_{initial} \\ \mathbf{z}_{k+1} = f(\mathbf{z}_k, \mathbf{v}_k) \\ \mathbf{z}_k \in \mathbb{Z}, \forall k \\ \mathbf{v}_k \in \mathbb{V}, \forall k \end{cases}$$

$$\mathbb{P}_N(\mathbf{z}_0, \mathbf{x}_0) = \min_{\mathbf{U}=\{\mathbf{u}_k\}_{k=1}^N} \frac{1}{2}\sum_{k=0}^{N-1}\left(|\mathbf{x}_k - \mathbf{z}_k|_Q^2 + |\mathbf{u}_k - \mathbf{v}_k|_R^2\right) + |\mathbf{x}_N - \mathbf{v}_N|_{P_f}^2 \tag{5}$$

$$s.t. \begin{cases} \mathbf{x}_0 = \mathbf{x}_{initial} \\ \mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k) \\ \mathbf{x}_k \in \mathbb{X}, \forall k \\ \mathbf{u}_k \in \mathbb{U}, \forall k \end{cases}$$

The tightened constraint sets $\mathbb{Z}$ and $\mathbb{V}$ are defined as $\mathbb{Z} = \alpha\mathbb{X}$ and $\mathbb{V} = \beta\mathbb{U}$ where $\alpha, \beta \in (0,1)$ are chosen in order to avoid constraints violation in the robust control implementation, and tested by Monte Carlo simulation of the controlled system. For each choice of α and β, the controller provides a degree of robustness that can be adjusted by modifying the "tuning" parameters α and β.

In this work, both standard and tube-based control strategies will be implemented in the control problems under investigation, and the impact of the choice of the controller in the scheduling problem will be analyzed.

3.2.3 Neural Networks

Neural networks and machine learning concepts have been discussed in sections 3.4.3. Here, we briefly overview the use of neural networks in classification and regression problems. For a regression problem, the predictor of a neural network with one hidden layer, a single output and logistic activation function takes the form of Eq. 6a. For a classification problem, the predictor of a neural network with the same structure takes the form of Eq. 6b. Note that classification problems will classify inputs into one of two categories, labeling them to belong to category +1 or -1.

$$f(\mathbf{x}) = \mathbf{w}_h \times \left(\frac{1}{1 + \exp\left(\mathbf{w}_{input} \times \mathbf{x} + \mathbf{b}_{input}\right)} \right) + b_h \tag{6a}$$

$$f(\mathbf{x}) = \text{sign} \left(\mathbf{w}_h \times \left(\frac{1}{1 + \exp\left(\mathbf{w}_{input} \times \mathbf{x} + \mathbf{b}_{input}\right)} \right) + b_h \right) \tag{6b}$$

In this work, neural networks will be used (i) as a classification method, in order to determine the feasible region of operation of the control problem (following the methodologies presented in Chapter 3); and (ii) as a regression method, in order to identify the closed-loop behavior of a subset of control variables that need to be represented in the scheduling problem. The details of the implementation of the neural networks in the integrated scheduling and control problem will be discussed in section 3.4.

4.3 Problem definition

Consider the integrated production scheduling and control problem for a continuous operation, for which production setpoints $\mathbf{x}_t^{sp}$ and $\mathbf{u}_t^{sp}$ should be defined for each scheduling period $t \in \mathrm{T}$, where T is the number of scheduling slots of the integrated problem. A rigorous and general representation of the integrated scheduling and control problem is given by problem (7).

$$\min_{\mathbf{x}_t^{sp},\mathbf{u}_t^{sp}} \sum_t \sum_k Cost_t(\mathbf{x}_{kt}, \mathbf{u}_{kt}) \tag{7a}$$

$$l(\mathbf{x}_{kt}, \mathbf{u}_{kt}) = 0 \tag{7b}$$

$$\mathbf{x}_{0,t+1} = \mathbf{x}_{K,t}, \qquad \mathbf{u}_{0,t+1} = \mathbf{u}_{K,t} \tag{7c}$$

$$\mathbf{x}_{k+1,t} = f(\mathbf{x}_{kt}, \mathbf{u}_{kt}) + \mathbf{w} \tag{7d}$$

$$\mathbf{u}_{k,t} = \psi(\mathbf{x}_{kt}, \mathbf{u}_{k-1,t}, \mathbf{x}_t^{sp}, \mathbf{u}_t^{sp}) \tag{7e}$$

$$\mathbf{x}_{kt} \in \mathbb{X}, \mathbf{u}_{kt} \in \mathbb{U} \tag{7f}$$

In this formulation, each scheduling slot t of duration τ is discretized into K control steps $k = \{0,1,..,K\}$, such that $\tau = K\Delta$, and Δ is the sample time of the control problem. It is assumed throughout this paper that the state of the system $\mathbf{x}_{kt}$ is perfectly measurable, and that the system outputs $\mathbf{y}_{kt}$ are connected to $\mathbf{x}_{kt}$ by $\mathbf{y}_{kt} = I\mathbf{x}_{kt}$, where I is the identity matrix. Therefore, $\mathbf{y}_{kt}$ is omitted for clarity. $\psi(\cdot)$ is the control law defined by the solution of (1) (if a standard MPC is employed) or (5) (if tube-based MPC is employed), and hence it defines the control actions $\mathbf{u}_{kt}$. Note that $\mathbf{u}_{kt}$ will depend on the previous manipulated variables $\mathbf{u}_{k-1,t}$ only if rate of change constraints are utilized in the MPC formulation. This is not the case for the MPC formulations implemented in this work. However, we include this possible dependency in Equation (7) for completeness. The objective function (7a)

includes the production costs of operating the system. Constraint (7b) ensures that that demands are met at all times, and that the amount of product stored does not deplete/exceed the physical capacity of the storage system. $l(\cdot)$ may also involve discrete decisions such as the assignment of products to scheduling slots, or the assignment of equipment to tasks. Constraint (7c) links the final state of the system at scheduling period t to the initial state of the system at scheduling period $t+1$. Constraint (7d) is the dynamic representation of the system, subject to an additive unmeasured bounded disturbance $\mathbf{w} \in \mathbb{W}$. Finally, constraint (7f) defines that the state and input variables should be contained in the polyhedral sets $\mathbb{X}$ and $\mathbb{U}$ at all times.

While problem (7) is an adequate formulation of the integrated problem, it represents a complex bilevel optimization problem that includes non-convexities and non-linearities. Therefore, this problem is very challenging to solve in reasonable computational times. The use of surrogate models on the integration of scheduling and control diminishes the complexity of the problem since it enables its reformulation as:

$$\min_{\mathbf{x}_t^{sp}, \mathbf{u}_t^{sp}} \sum_t Cost_t(\bar{\mathbf{x}}_t^s, \bar{\mathbf{u}}_t^s) \tag{8a}$$

$$l(\bar{\mathbf{x}}_t^s, \bar{\mathbf{u}}_t^s) = 0 \tag{8b}$$

$$\tilde{\mathbf{x}}_{t+1}^0 = \tilde{\mathbf{x}}_t, , \tilde{\mathbf{u}}_{t+1}^0 = \tilde{\mathbf{u}}_t \tag{8c}$$

$$\tilde{\mathbf{x}}_t = \phi^{\tilde{x}}(\mathbf{x}_t^{sp}, \mathbf{u}_t^{sp}, \tilde{\mathbf{x}}_t^0, \tilde{\boldsymbol{u}}_t^0) \tag{8d}$$

$$\tilde{\mathbf{u}}_t = \phi^{\tilde{u}}(\mathbf{x}_t^{sp}, \mathbf{u}_t^{sp}, \tilde{\mathbf{x}}_t^0, \tilde{\boldsymbol{u}}_t^0) \tag{8e}$$

$$\bar{\mathbf{x}}_t^s = \phi^{\bar{x}}(\mathbf{x}_t^{sp}, \mathbf{u}_t^{sp}, \tilde{\mathbf{x}}_t^0, \tilde{\boldsymbol{u}}_t^0) \tag{8f}$$

$$\bar{\mathbf{u}}_t^s = \phi^{\bar{u}}(\mathbf{x}_t^{sp}, \mathbf{u}_t^{sp}, \tilde{\mathbf{x}}_t^0, \tilde{\boldsymbol{u}}_t^0) \tag{8g}$$

$$\xi\left(\mathbf{x}_t^{sp}, \mathbf{u}_t^{sp}, \tilde{\mathbf{x}}_t^0, \tilde{\boldsymbol{u}}_t^0\right) \leq 0 \tag{8h}$$

In this problem, the scheduling model is represented by Eqs. (8a) to (8b). Eq. (8c) is the linking constraint, which connects the final state and input of the system at scheduling period t to the initial state and input of the system at scheduling period $t+1$. The closed-loop behavior of the dynamic system is represented by Eqs. (8d) to (8h). In other words, equations (8d) to (8h) are surrogate models approximating the original full-space control problem previously described by Eqs. (7d) to (7f).

A few assumptions have been made in this reformulation:

(i) It can be inferred from (7) that both $\mathbf{x}_{kt}$ and $\mathbf{u}_{kt}$, $\forall k = \{1,2,..,K\}$ are functions of the setpoints of the system, $\mathbf{x}_t^{sp}$ and $\mathbf{u}_t^{sp}$, as well as its initial conditions, $\mathbf{x}_{0t}, \mathbf{u}_{0t}$. Therefore, the average values $\bar{\mathbf{x}}_t = \sum_k \mathbf{x}_{kt}$ /K and $\bar{\mathbf{u}}_t = \sum_k \mathbf{u}_{kt}$ /K, as well as the final state of the system at each scheduling slot, $\mathbf{x}_{Kt}$, can be accurately described by functions $\Phi(\mathbf{x}_t^{sp}, \mathbf{u}_t^{sp}, \mathbf{x}_{0t}, \mathbf{u}_{0t})$ in a deterministic system (i.e., $\mathbf{w} = 0$); and reasonably approximated by $\phi(\mathbf{x}_t^{sp}, \mathbf{u}_t^{sp}, \mathbf{x}_{0t}, \mathbf{u}_{0t})$ in a non-deterministic system if $\mathbf{w}$ is sufficiently small.

(ii) If the scheduling discretization time is appropriately chosen, it is reasonable to expect that the production costs of the system ($Cost_t$), as well as the scheduling constraints $l(\cdot)$ can be modeled as functions of a subset of $\bar{\mathbf{x}}_t$ and $\bar{\mathbf{u}}_t$, namely $\bar{\mathbf{x}}_t^s \subseteq \bar{\mathbf{x}}_t = \sum_k \mathbf{x}_{kt}$ /K and $\bar{\mathbf{u}}_t^s \subseteq \bar{\mathbf{u}}_t = \sum_k \mathbf{u}_{kt}$ /K

(iii) We assume it is possible to determine a classification model $\xi(\cdot)$ as a function of $\left(\mathbf{x}_t^{sp}, \mathbf{u}_t^{sp}, \mathbf{x}_{0t}, \mathbf{u}_{0t}\right)$, such that $\xi(\cdot)$ results in +1 if any violation of state and input constraints, $\mathbf{x}_k \in \mathbb{X}$ and $\mathbf{u}_k \in \mathbb{U}, \forall k = \{0,1,\dots,K\}$ occur during simulations of the control system; and $\xi(\cdot)$ assumes the values -1 otherwise.

(iv) We define $\tilde{\mathbf{x}}_0^t \approx \mathbf{x}_{0t}$, $\tilde{\mathbf{x}}_t \approx \mathbf{x}_{Kt}$, $\tilde{\mathbf{u}}_0^t \approx \mathbf{u}_{0t}$, $\tilde{\mathbf{u}}_t \approx \mathbf{u}_{Kt}$

(v) We recognize that the setpoints of a system $\mathbf{x}_t^{sp}$ and $\mathbf{u}_t^{sp}$ may present strong correlations if they are defined based on the steady state of the system. In such cases, including both $\mathbf{x}^{sp}$ and $\mathbf{u}^{sp}$ as inputs, or features, in the surrogate model training is redundant and may cause degeneracies [110]. Therefore, we will include either $\mathbf{x}^{sp}$ or $\mathbf{u}^{sp}$ in the surrogate model training. With this strategy, the dimensionality of problem (8) is further reduced.

The integrated scheduling and control problem using surrogate models consists of the use of detailed dynamic models of the system, $\mathbf{x}_{k+1,t} = f(\mathbf{x}_{kt}, \mathbf{u}_{kt}) + \mathbf{w}$ and control strategies (described in section 4.2.2) to simulate the closed-loop behavior of the control problem; and the use of surrogates to extract input-output relationships from the simulation data in order to approximated equations (8d) to (8g). In practice, the integrated problem would define scheduling decisions and production setpoints for the entire scheduling horizon, and transmit these decisions to the control layer, which would in fact implement the solutions.

In the next section, a framework to achieve the integration of scheduling and robust model predictive control is discussed in detailed.

4.4. Framework for the integration of scheduling and control

The integrated scheduling and control framework proposed in this work consists of four basic steps, as shown in Figure 4.1.

Step 1: Implement standard Model Predictive Control and Tube-based Model Predictive Control to the systems under consideration

Step 2: Obtain historical process operating data representative of typical production schedules by simulating the control system

Step 3: For each control strategy and related simulation data, obtain surrogate models $\phi^{\tilde{x}}, \phi^{\bar{x}}$ and $\phi^{\bar{u}}$ using neural network regressors to approximate $\tilde{\mathbf{x}}_t, \bar{\mathbf{x}}_t^s$ and $\bar{\mathbf{u}}_t^s$

Step 4: Obtain model $\xi(\cdot)$ using neural network classification methods to classify data points $\left(\mathbf{x}_t^{sp}, \mathbf{u}_t^{sp}, \mathbf{x}_{0t}\right)$ as feasible $(\xi(\cdot) = -1)$ or infeasible $(\xi(\cdot) = +1)$ using the simulation data

Step 5: Solve the integrated scheduling and control problem by incorporating the neural network predictors in the scheduling optimization problem

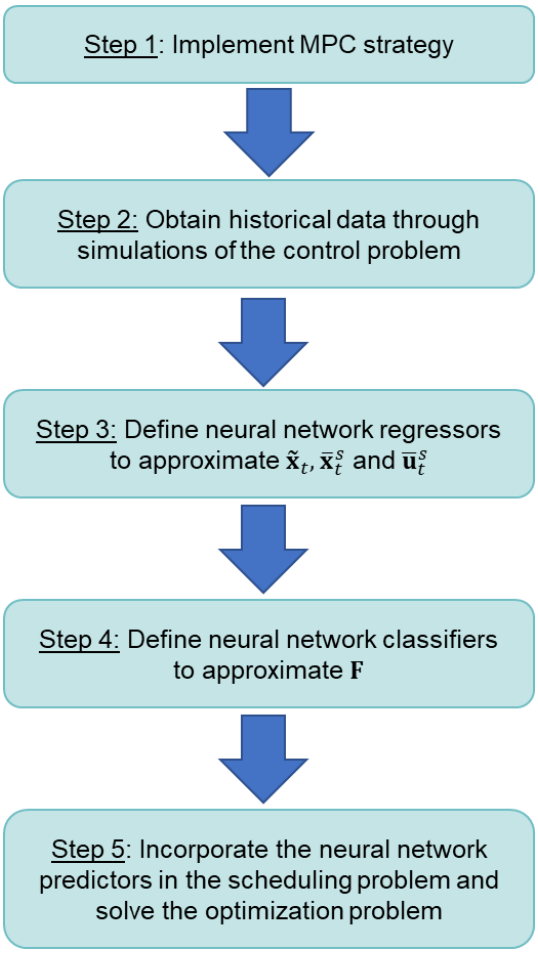

Figure 4.1 - Framework for the integration of scheduling and control

In Step 2, the control problem will be simulated for randomly generated values of $\left(\mathbf{u}_i^{sp}, \mathbf{x}_{0i}\right)$, $\mathbf{x}_{01} \in \mathbb{X}$ and $\mathbf{u}_i^{sp} \in \mathbb{U}$, where i denotes an integer, $i = \{1,2, \dots, Sx\}$, and Sx is the number of sample points. For every sample point i, the setpoint of the state variables $\mathbf{x}_i^{sp}$ will be defined based on the steady state conditions of the system and such that $\mathbf{x}_i^{sp} \in \mathbb{X}$. The inputs of the simulation i will consist of $S_i = \left(\mathbf{x}_i^{sp}, \mathbf{u}_i^{sp}, \mathbf{x}_{0i}\right)$. For each sample point, the simulation will take place for a τ period of time, where τ is the scheduling discretization interval. At the end of the simulation, the values of $\tilde{\mathbf{x}}_i, \bar{\mathbf{x}}_i^s$ and $\bar{\mathbf{u}}_i^s$ are computed and stored as $\mathbf{y}_i = (\tilde{\mathbf{x}}_i, \bar{\mathbf{x}}_i^s, \bar{\mathbf{u}}_i^s)$. A value for $\mathbf{f}_i$ is also stored, where $\mathbf{f}_i$ takes the value of +1 if, at any point throughout the simulation, state and input constraints are violated. $\mathbf{f}_i = -1$ otherwise. For every sample point i, the disturbance $\mathbf{w}$ will follow a different trajectory that is unknown to the controller. Therefore, it is not unexpected that two set of data points i and j, such that $\left(\mathbf{x}_i^{sp}, \mathbf{u}_i^{sp}, \mathbf{x}_{0i}\right) = \left(\mathbf{x}_j^{sp}, \mathbf{u}_j^{sp}, \mathbf{x}_{0j}\right)$, generate different simulation outputs $\mathbf{y}_i \neq \mathbf{y}_j$. The regressive nature of the neural network in step 3 will handle such cases with no further issues.

Furthermore, it is expected that tube-based MPC will take conservative control actions when compared to standard MPC. The different control performances will generate different outputs, even though simulations will be performed for the same values of disturbances $\mathbf{w}$ and the same data points $\left(\mathbf{x}_i^{sp}, \mathbf{u}_i^{sp}, \mathbf{x}_{0i}\right)$, $i = \{1,2, \dots S\}$. While conservative, it is expected that tube-based MPC will result in feasible operation for a wider range of data points, and the effects of this feasible region in the scheduling problem are analyzed.

Steps 3 and 4 will be carried out based on the simulation data generated in step 2, using neural network regression and classification methodologies. The implementation was

performed in MATLAB using the Deep Learning Toolbox. All the neural networks were designed with one hidden layer and logistic activation function.

In Step 5, the predictors for the neural networks (which take the form of Equation (6a) for the regression case, and Equation (6b) for the classification case), are incorporated in the scheduling model, providing an integrated scheduling and control formulation. The integrated problem will be solved to local optimality, and the solution of the problem will be implemented in the control layer.

In the next section, the proposed framework is implemented in two case studies, demonstrating its performance and the differences between the preventive and deterministic approaches regarding the disturbances at the control problem.

4.5 Case studies

4.5.1 Integration of scheduling and control for a CSTR

We will first consider a simple, low-dimension, scheduling problem, in order to demonstrate the concepts of the framework. Consider the problem of integrating scheduling and control in a continuously operated stirred tank reactor followed by a well-mixed storage tank as shown in Figure 4.2. An irreversible, first-order reaction $R \rightarrow P$ occurs in the liquid phase, and the reactor temperature is regulated with external cooling. Mass and energy balances are given by Eq. 9a and 9b; respectively.

$$\frac{dc}{dt} = \frac{F0 \cdot (C0 - C)}{\pi \cdot r^2 \cdot h} - k0 \cdot C \cdot \exp\left(-\frac{E}{R \cdot T}\right) \tag{9a}$$

$$\frac{dT}{dt} = \frac{F0 \cdot (T0 - T)}{\pi \cdot r^2 \cdot h} - \frac{\Delta H}{\rho \cdot Cp} k_0 \exp\left(-\frac{E}{R \cdot T}\right) + \frac{2 \cdot U}{h \cdot \rho \cdot Cp}(Tc - T) + w \tag{9b}$$

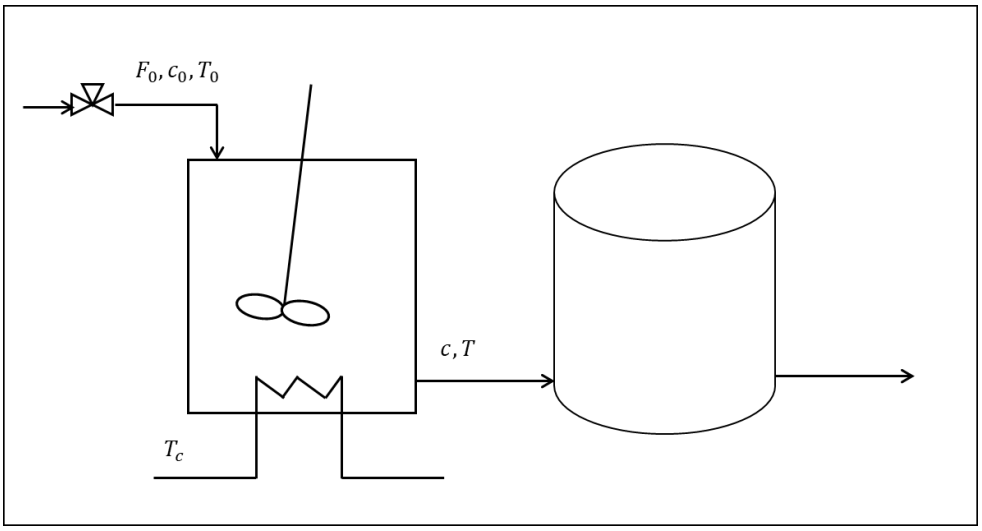

Figure 4.2 - CSTR and storage tank schematic

The state vector $\mathbf{x}_t$ includes the molar concentration of R, c, and the reactor temperature, T, i.e., $\mathbf{x}_t = [c_t, T_t]$. The manipulated variables are the inlet flow rate $F0$, and the coolant liquid temperature Tc, and therefore $\mathbf{u}_t = [Tc_t, F0_t]$. Additionally, we assume an unmeasured disturbance w affects the reactor temperature. The disturbance takes the form $w_k = A\sin(\omega \cdot k \cdot \Delta)$, where A and ω are independent uniformly distributed random variables, taking values in the sets $[0, 5]$ and $[-1,1]$, respectively. However, such model is unknown to the controller. The parameters of the CSTR are shown in Table 4.1.

The scheduling of the CSTR consists of setting different setpoints for each hour in a 24-hour scheduling horizon, while considering a varying cost of raw material. The cost of raw material varies at each scheduling slot as shown in Figure 4.3. Deliveries to the customer are made every 4 hours. A bound on the hourly average impurity concentration of the outlet stream is imposed at all times according to customer specifications. Furthermore, a storage tank with limited capacity can hold finished products throughout the entire scheduling horizon. It is assumed that the storage tank is well-mixed and that no further reaction occurs in the tank.

Table 4.1 - Parameters of the CSTR dynamic model

Parameter	Nominal Value
$C0$	$1\ \mathrm{kmol/m^3}$
$T0$	$350\ \mathrm{K}$
h	$0.219\ \mathrm{m}$
$k0$	$7.2 \times 10^{10}\ \mathrm{min^{-1}}$
E/R	$8750\ \mathrm{K}$
U	$54.94\ \mathrm{kJ/(min \cdot m^2 \cdot K)}$
ρ	$1000\ \mathrm{kg/m^3}$
Cp	$0.239\ \mathrm{kJ/(kg \cdot K)}$
ΔH	$-50000\ \mathrm{kJ/kmol}$

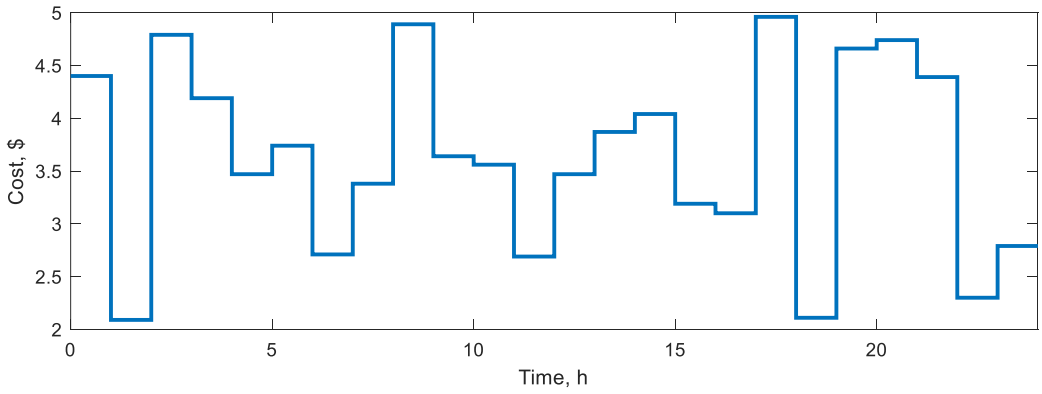

Figure 4.3 - The cost of raw material varying with time

The discrete-time scheduling model of this problem (corresponding to equations (8a) and (8b)) is given by:

$$\min_{\mathbf{x}_t^{sp},\mathbf{u}_t^{sp}} \sum_t p_t^r \cdot \bar{u}_t^2 \cdot \tau \tag{10a}$$

$$I_t = I_{t-1} + \bar{u}_t^2 \cdot \tau - D_t, \qquad \forall t \tag{10b}$$

$$I_0 = 0 \tag{10c}$$

$$0 \leq I_t \leq \bar{I}, \qquad \forall t \tag{10d}$$

$$\bar{x}_t^1 \leq \bar{C} \tag{10e}$$

The objective of this problem is the minimization of costs, given by Eq. (10a). The cost is a function of the cost of raw materials p_t^r, and the raw material consumption in each slot, $\bar{u}_t^2 = \overline{F0}_t$. Equation (10b) corresponds to mass balances determining the inventory levels at the end of each scheduling slot. The inventory level I_t is a function of the inventory level at the previous scheduling slot, I_{t-1}; a function of the production levels, $\bar{u}_t^2 \cdot \tau$; and a function of the demand at scheduling slot t, D_t. Equations (10c) assigns the initial inventory level to 0, and Eq. (10d) determines lower and upper bounds for the inventory. Equation (10e) sets an upper bound to the average impurity concentration in the outlet stream of the CSTR, a quality requirement that is allowed to vary from one scheduling horizon to the other.

Following the proposed framework, we first implement standard and robust MPC strategies to the CSTR in consideration. The MPC strategies were implemented using MPCTools/CasADi in MATLAB interface [111, 112]. The performance of standard and tube-based MPC in a deterministic operation ($w = 0$) is shown in Figure 4.4. The performance of standard and robust MPC in a non-deterministic operation is shown in Figure 4.5. In this case, 10 different values for A and ω were randomly generated, and the

simulation was repeated for each value of $w = A\sin(\omega \cdot k \cdot \Delta)$. In both cases, the system was simulated for 1 hour, the sampling time of the control problem was $0.25s$, and the prediction horizon of the control problem was $N = 15$. The initial state of the system was given by $\mathbf{x}_0 = [0, 300]$, and the setpoints were given by $\mathbf{x}^{sp} = [0.53, 348.42]$, $\mathbf{u}^{sp} = [300.5, 0.1]$.

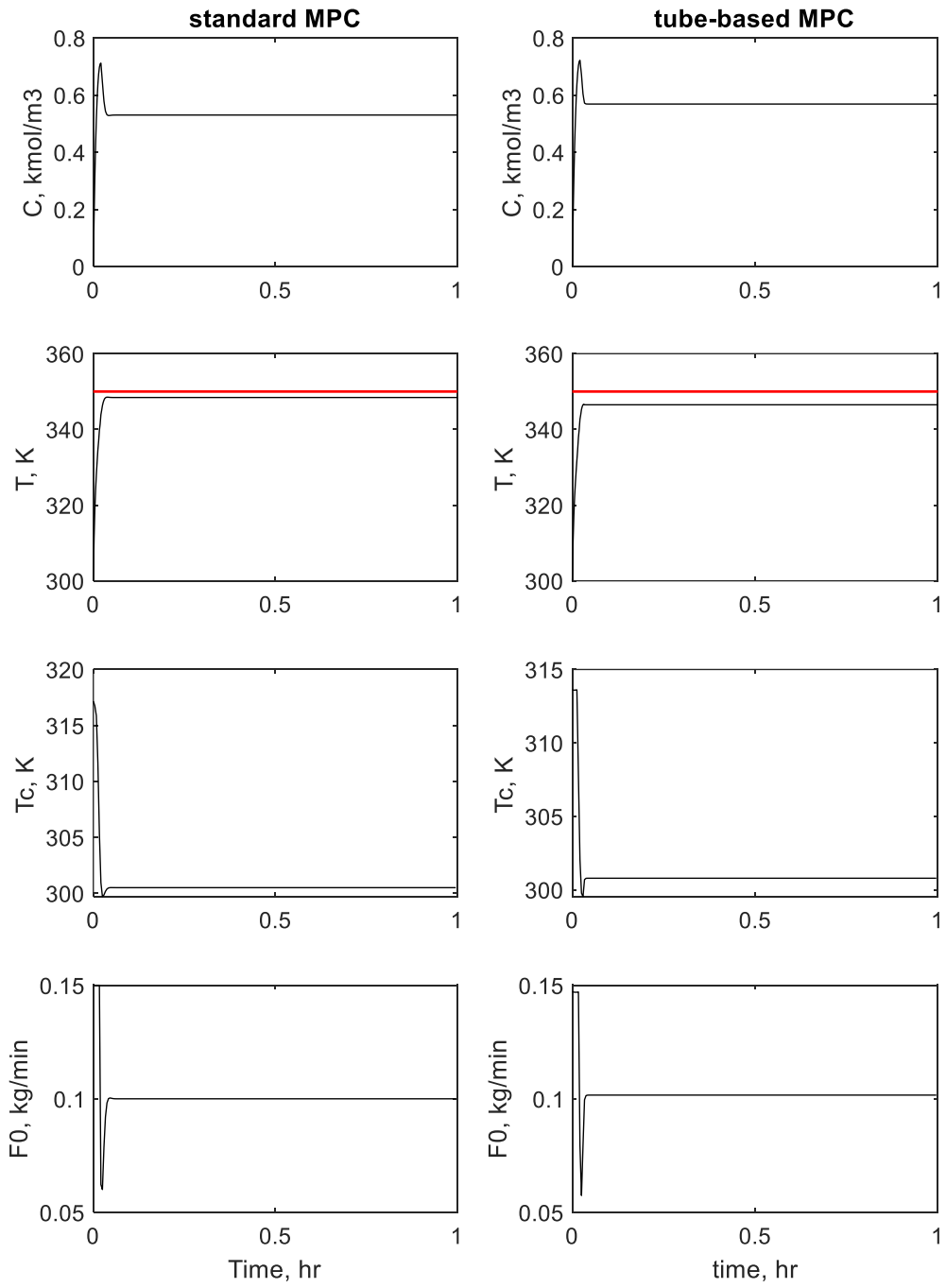

Figure 4.4 - The state and input trajectories of a deterministic system

On the left, a standard MPC was implemented. On the right, a tube-based MPC was implemented.

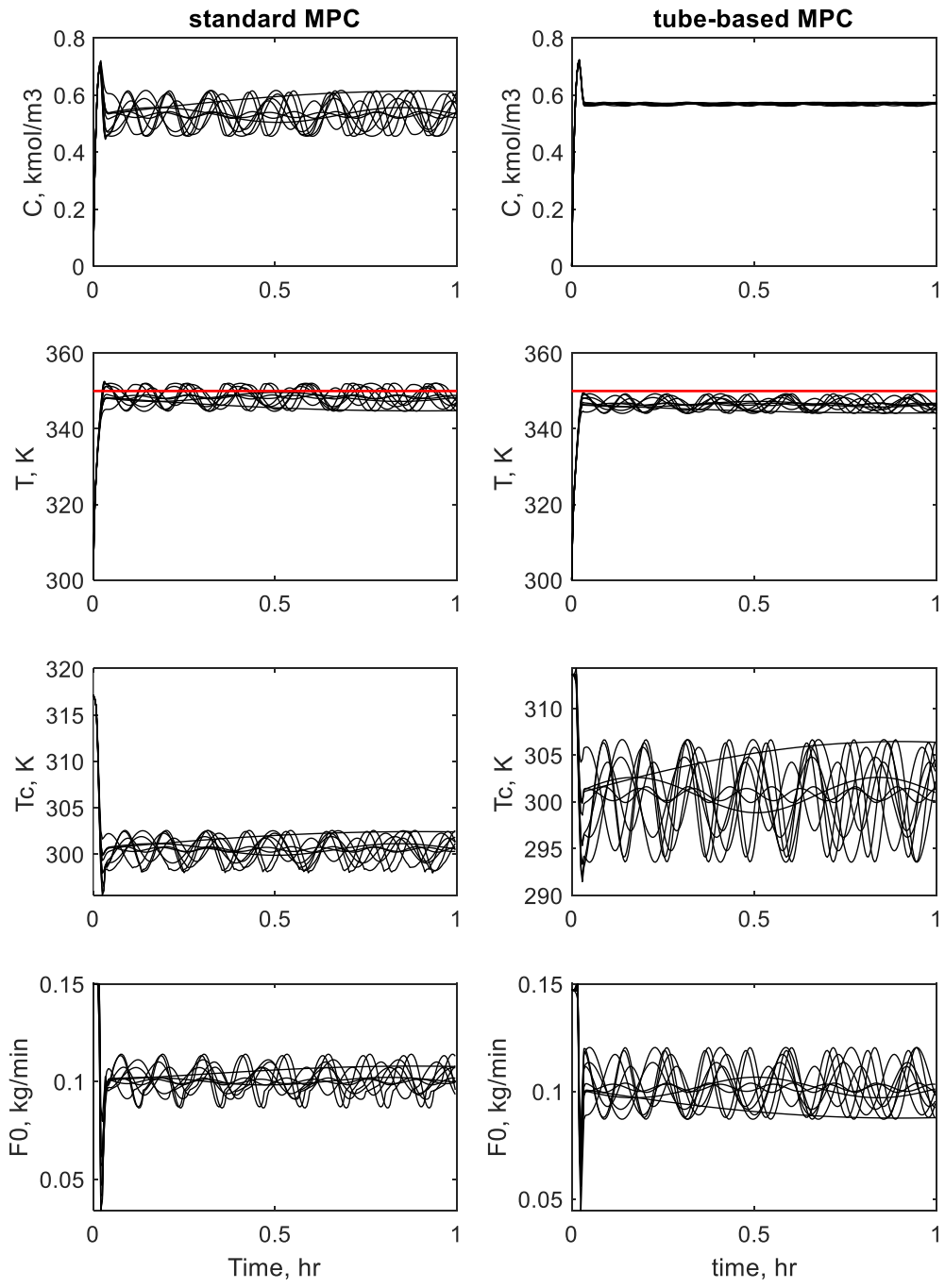

Figure 4.5 - Simulations of state and input trajectories for a system affected by disturbances. On the left, a standard MPC was implemented. On the right, a tube-based MPC was implemented.

Tube-based MPC sacrifices setpoint tracking in order to obtain feasible operation. This can be concluded by analyzing the average value of state and input variables in the deterministic simulation. For standard MPC, $\bar{\mathbf{x}}$ and $\bar{\mathbf{u}}$ take the values of $[0.53, 348.42]$ and $[300.5, 0.1]$, respectively. For tube-based MPC, $\bar{\mathbf{x}}$ and $\bar{\mathbf{u}}$ take the values of $[0.5685, 346.5]$ and $[300.78, 0.102]$, respectively. In this case study, the higher levels of impurity in the flow stream will also affect the economics of the problem. However, tube-based MPC outperforms standard MPC in the non-deterministic simulations when analyzing constraint violation. Tube-based MPC is also able to maintain a smoother trajectory of the state of the system, albeit at the cost of aggressive control actions.

Following the proposed framework, once the MPC strategies are implemented, a series of simulations are performed in order to obtain historical process operating data. In this case, 1000 data points $\left(\mathbf{u}_i^{sp}, \mathbf{x}_{0i}\right)$ were randomly generated. For each data point i, $\mathbf{x}_i^{sp}$ was obtained as the steady state of the dynamic system (10), given $\mathbf{u}_i^{sp}$. Since the CSTR described by problem (9)` may present multiple steady states, $\mathbf{x}_i^{sp} = [c_i^{sp}, T_i^{sp}]$ was chosen to minimize c_i^{sp} while satisfying $\mathbf{x}_i^{sp} \in \mathbb{X}$. The inputs for simulation i were then defined as $S_i = \left(\mathbf{x}_i^{sp}, \mathbf{u}_i^{sp}, \mathbf{x}_{0i}\right)$. Each simulation run was equivalent to one hour of operation, as $\tau =$ 1 hour in the scheduling problem. Therefore, the performance of the control and the dynamic system was simulated for a period equivalent to 6 weeks of operation. In each run, both standard and tube-based MPC were simulated. At the end of each simulation i, the values for $\mathbf{y}_i^{\mathbf{std}} = (\tilde{\mathbf{x}}_i, \bar{\mathbf{x}}_i^s, \bar{\mathbf{u}}_i^s)$ and $\mathbf{y}_i^{\mathbf{tube}} = (\tilde{\mathbf{x}}_i, \bar{\mathbf{x}}_i^s, \bar{\mathbf{u}}_i^s)$ were computed and stored. Values for $\mathbf{f}_i^{std}$ and $\mathbf{f}_i^{tube}$ were also stored, where the values of +1 or -1 were assigned to $\mathbf{f}_i^{std}$ and $\mathbf{f}_i^{tube}$ depending on the existence of constraint violations in each one of the control strategies simulations.

Next, the data sets $\mathbf{S} = \{S_1, S_2, \dots S_{1000}\}$, $\mathbf{Y}^{std} = \{\mathbf{y}_1^{std}, \mathbf{y}_2^{std}, \dots, \mathbf{y}_{1000}^{std}\}$ and $\mathbf{Y}^{tube} = \{\mathbf{y}_1^{tube}, \mathbf{y}_2^{tube}, \dots, \mathbf{y}_{1000}^{tube}\}$ were used to train neural networks to estimate $\bar{x}_t^1$, $\bar{u}_t^2$ and $\mathbf{x}_{0t}$ as functions of $\left(\mathbf{u}_i^{sp}, \mathbf{x}_{0i}\right)$ for each control strategy. Two important points were considered in this step. First, $\mathbf{x}_i^{sp}$ was not utilized in the neural network training procedure since it is strongly correlated to $\mathbf{u}_i^{sp}$, and therefore its addition to the input set would be redundant. Second, the surrogate model building procedures were restricted to the subsets $\bar{x}_t^1 \subseteq \bar{\mathbf{x}}_t$ and $\bar{u}_t^2 \subseteq \bar{\mathbf{u}}_t$ since the average raw material consumption, $\bar{u}_t^2$, and the average impurity concentration out of the reactor, $\bar{x}_t^1$ are the only subset of variables relevant to the scheduling problem. The neural networks were trained using MATLAB Deep Learning Toolbox, and predictors in the form of Eq. (6a) were obtained.

The data sets $\mathbf{S} = \{S_1, S_2, \dots S_{1000}\}$ and $\mathbf{F}^{std} = \{\mathbf{f}_1^{std}, \mathbf{f}_2^{std}, \dots, \mathbf{f}_{1000}^{std}\}$ and $\mathbf{F}^{tube} = \{\mathbf{f}_1^{tube}, \mathbf{f}_2^{tube}, \dots, \mathbf{f}_{1000}^{tube}\}$ were then used to train neural network classifiers to estimates feasibility functions $\xi^{std}(\cdot)$ and $\xi^{tube}(\cdot)$. The neural networks were trained using MATLAB Deep Learning Toolbox, and predictors in the form of Eq. (6b) were obtained. Finally, two integrated scheduling and control problems were formulated: 1. The standard integrated problem, obtained by incorporating the surrogates trained with standard MPC simulation data in the scheduling model; and 2. The robust integrated problem, obtained by incorporating the surrogates trained with tube-based MPC simulation data in the scheduling model. The resulting integrated problems belong to the class of Nonlinear Programming Problems (NLP), and were solved using GAMS CONOPT on a 64-bit Windows system with Intel Core i7-6700 CPU at 2.60 GHz and 8GB RAM. The optimization results are shown in Table 4.2.

The optimization solution includes setpoints $\mathbf{u}_t^{sp}$ that are used to simulate the system, and calculate the actual costs of operation. Difference between predicted costs and actual costs of operation are expected because of the approximation due to the neural network regressors, and because of the disturbances affecting the system. The actual costs of operation are included in Table 4.2, and Figure 4.6 shows the state and control dynamic behavior for the entire scheduling horizon.

The robust strategy results in a slightly higher operational cost when compared to the standard strategy. However, the benefits regarding the feasibility of the scheduling solution from a control perspective are incontestable. Tube-based MPC manages to avoid constraints violation throughout most of the operation, while standard MPC fails to respect both the control and the scheduling constraints. It is safe to say that tube-based MPC outperforms standard MPC from both scheduling and control perspectives.

Table 4.2 - Optimal schedule results for case study 1

Case	**Solution Time**	**Predicted Cost ($)**	**Actual cost ($)**	**Difference (%)**
Standard	0.608 s	$ 367.42	$ 369.66	0.61%
Robust	1.216 s	$ 369.81	$ 370.58	0.21%

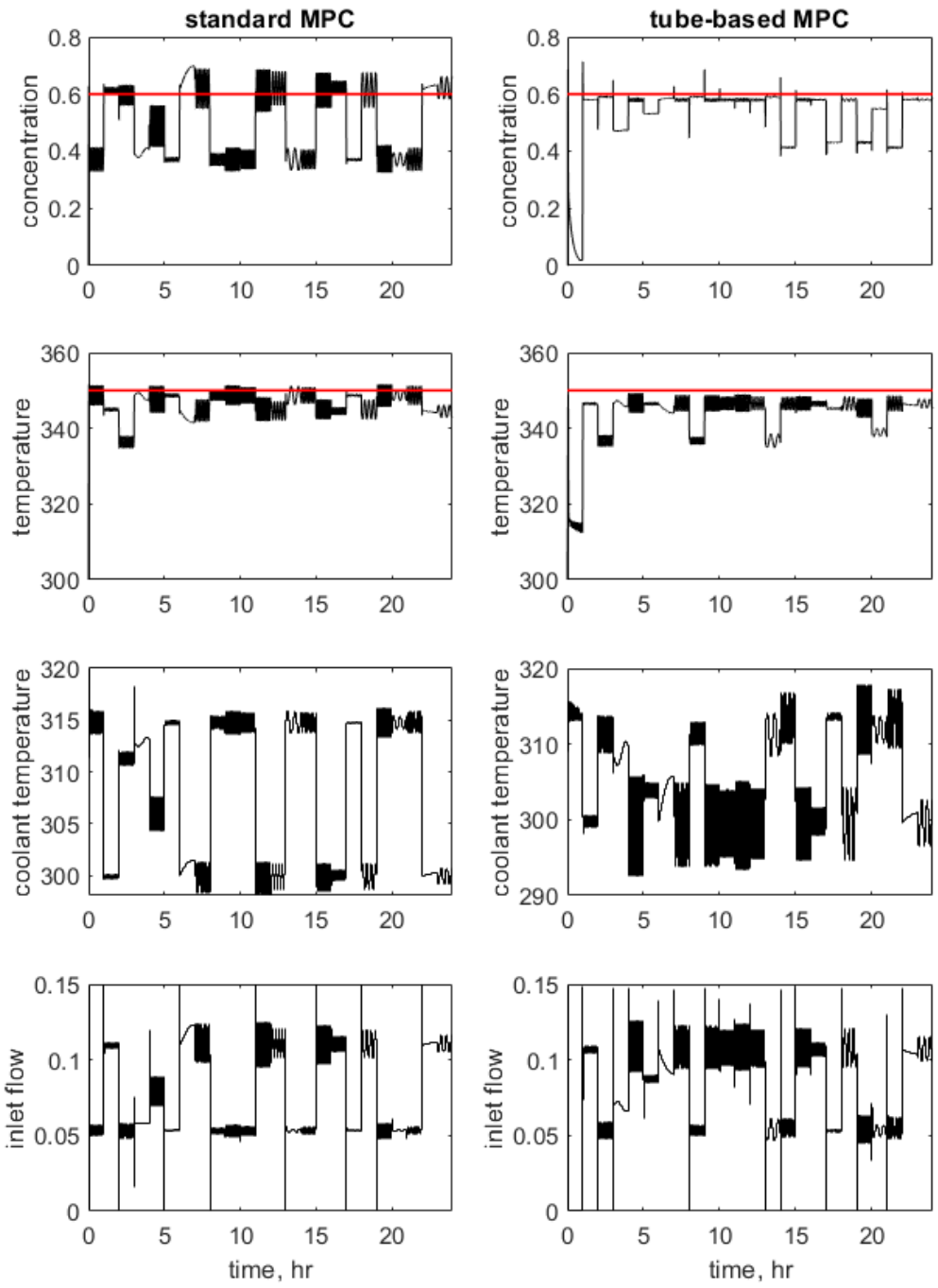

Figure 4.6 - State and input closed-loop behavior of the system in a 24hr simulation. Two control strategies (standard MPC and tube-based MPC) are tracking setpoints defined by the scheduling optimization

4.5.2 Milk powder optimization using a Spray Dryer

In this section, a more complex scheduling and control problem is addressed. Consider the operation of a spray dryer, such as the one presented by Petersen, Poulsen [113]. The spray dryer is used to produce milk powder, and the main challenges in its operation are related to bringing the residual moisture of the powder below some specification, while avoiding that powder sticks to the chamber walls and reducing energy consumption. The spray dryer under consideration is depicted in Figure 4.7.

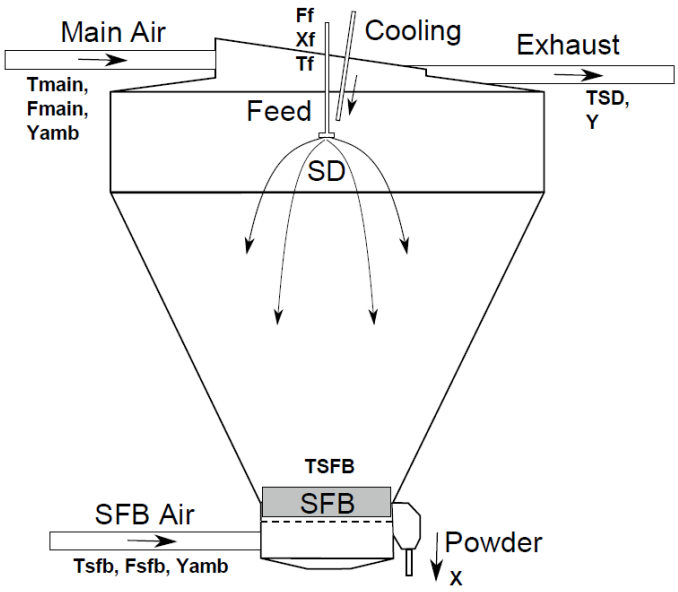

Figure 4.7 -A schematic of the spray dryer

Hot inlet air feeds the spray dryer chamber (SD) around the high-pressure nozzles. The nozzles disperse the liquid feed into droplets. Water from the droplets evaporates by absorbing heat from the hot air. In this process, the residual moisture of the droplets decreases. The dried product then enters a static fluid bed (SFB) where it is further dried. Mass and energy balances for this system are given in Eq. 11.

$$m_a \frac{dY_{ab}}{dt} = (F_{main} + F_{sfb})(Y_{amb} - Y_{ab}) + F_{add}(Y_{add} - Y_{ab}) + R_{aw} \quad (11a)$$

$$m_b \frac{dX_{ab}}{dt} = F_s(X_f - X_{ab}) - R_{aw} \tag{11b}$$

$$C_a \frac{dT_{SD}}{dt} = -\lambda R_{aw} + F_{main}\Delta h^a_{da,main} + F_{sfb}\Delta h^a_{da,sfb} + F_{add}\Delta h^a_{da,add} + F_{main}Y_{amb} \\ F_{sfb}Y_{amb}\Delta h^a_{v,sfb} + F_{add}Y_{add}\Delta h^a_{v,add} + F_s\Delta h^a_s + F_s X_f \Delta h^a_w - Q_{ab} - Q_a \tag{11c}$$

$$C_b \frac{dT_{SFB}}{dt} = F_{sfb}\Delta h^b_{da} + F_{sfb}\Delta h^b_v + F_s\Delta h^b_s + F_s X_{ab}\Delta h^b_w + Q_{ab} - Q_b \tag{11d}$$

where

$$\Delta h^a_{i,j} = \int_{TSD}^{T_j} C_{p,i}(T)dT, \qquad i = \{da, v, w, s\},\ \ j = \{main, sfb, add\} \tag{11e}$$

$$\Delta h^b_{i,j} = \int_{TSFB}^{T_j} C_{p,i}(T)dT, \qquad i = \{da, v, w, s\},\ \ j = \{main, sfb, add\} \tag{11f}$$

$$Q_{ab} = k_1(T_{SD} - T_{SFB}) + k_2 X_f + k_3 T_f - k_4 \tag{11g}$$

$$Q_a = k_5(T_{SD} - T_{amb}) \tag{11h}$$

$$Q_b = k_6(T_{SFB} - T_{amb}) \tag{11i}$$

It is assumed that the air and the product are in equilibrium, i.e., that the temperature of the air, T_{SD} and T_{SFB}, and the temperature of the product are identical. $\Delta h^a_{\{.\}}$ and $\Delta h^p_{\{.\}}$ are the enthalpy changes for the humid air and powder inlets and outlets on the SD and SFB stages. C_a and C_b are the heat capacities of the hold-up of air and powder. λR_{aw} is the heat of evaporation and Q_{ab} describes the heat exchange between the SD and SFB stages. Q_a and Q_b are heat losses to the surroundings. F_{main} and F_{sfb} are the dry base inlet air flows. The parameters Y_{add}, F_{add} and T_{add} are used to compensate for air leakages and un-modeled inlet air flows such as nozzle cooling air. F_s is the flow of feed solids, $F_s = F_f S_f$. X_f and T_f are the dry base feed concentration and feed temperature. m_a is the mass of dry air and m_b is the mass of dry powder. The parameters for this problem are given in Table 4.3.

Table 4.3 - Parameters for the spray dryer model

Symbol	Value	Unit
C_a	61.634	kJ/K
C_b	148.26	kJ/K
k_1	0.2725	kW/K
k_2	1.5017	kW
k_3	0.0605	kW
k_4	27.276	kW
k_5	0.24735	kW/K
k_6	-0.03198	kW/K
F_{add}	248.54	kg/h
Y_{add}	9.4566	g/kg
T_{add}	60.018+273.15	K
T_f	324.66	K
S_f	0.4897	

We assume that the evaporation takes place in the SD stage only with the drying rate determined from conditions in the SFB. The product drying rate is governed by the thin layer equation:

$$R_{aw} = \frac{k_7 k_8}{k_8 + F_s}\left(\frac{T_f}{T_0}\right)^{k_9}\left(X_{ab} - X_{eq}\right)m_b \tag{12}$$

where m_b is the mass of dry powder. The equilibrium moisture, X_{eq}, describes the moisture content at which water cannot be evaporated from the powder any longer, and it is given by Eq. 13.

$$X_{eq} = C \cdot K \cdot X_m \cdot \frac{RH}{(1 - K \cdot RH)(1 - K \cdot RH + C \cdot K \cdot RH)} \quad (13)$$

where $X_m = 0.030723$, $C = 2.6535 \cdot 10^{-7} \exp(\frac{6292.1}{T})$ and $K = 0.057882 \exp\left(\frac{945.16}{T}\right)$. X_m, C and K are Guggenheim-Anderson-de-Boer constants related to monolayer and multilayer properties. The relative humidity RH is calculated from T_{SFB} and Y_{ab}.

One important consideration in the spray dryer operation is the stickiness of the produced particles. Sticky particles form depositions on the walls of the spray dryer, and therefore should be avoided during operation. Stickiness can be predicted by glass transition temperature given by:

$$T_g = \frac{T_{gp} + kZT_{gw}}{1 + kZ} \quad (14)$$

in which $T_{gp} = 144.8^o C$ and $T_{gw} = -137^o C$. The obtained glass transition temperatures, T_g^{SD} and T_g^{SFB}, are the upper limiting temperatures of which deposits form on the chamber walls of the spray dryer. The moisture content of the powder is given by Eq. (15)

$$Z = \begin{cases} \left(A_p + B_p T_{SD}\right) \exp C_p RH(T_{SD}, Y_{ab}) & for\ SD \\ X_{ab} & for\ SFB \end{cases} \quad (15)$$

in which $A_p = 0.193$, $B_p = -0.000435$ and $C_p = 4.51$.

Control and state constraints of the problem include the maximum capacity of the feed pump, which limits the feed flow such that $0kg/h \leq F_f \leq 120kg/h$. The risk of powder explosions and the risk of scorched particles creates upper limits on the allowable inlet temperatures and, consequently, $T_{main} \leq 120^o C$ and $T_{sfb} \leq 200^o C$. To avoid depositions of sticky particles on the spray dryer surfaces, the temperatures T_{SD}, T_{SFB} must be below the glass transition temperatures in the SD stage, $T_{SD} \leq T_g^{SD}$, and the SFB stage,

$T_{SFB} \leq T_g^{SFB}$. Finally, the powder moisture must be below a maximum limit, $X_{ab} \leq X_{max} = 35$ g/kg.

In summary, the state variables of the system include the powder moisture X_{ab}, the air humidity in the outlet of the spray dryer, Y_{ab}, the temperature of the spray dryer chamber, T_{SD}, and the temperature on the static fluid bed, T_{SFB}. Therefore, $\mathbf{x} = [Y_{ab}, X_{ab}, T_{SD}, T_{SFB}]$. The input variables include the powder feed flow, F_f, as well as the temperature of the main air inlet, T_{main}, and the temperature of the SFB air inlet, T_{sfb}. Therefore, $\mathbf{u} = [F_f, T_{main}, T_{sfb}]$. The constraints include soft bounds on state variables and hard bounds on the input variables. In the MPC implementation, Eq. 14 was linearized around the nominal state in order to keep the constraint set polyhedral.

We further assume that the ambient humidity, Y_{amb}, acts as a disturbance to the system, taking the form $Y_{amb} = A\sin(\omega k)$, where A and ω are random values generated within the intervals $[0,0.0002]$ and $[0,1]$, respectively.

Before implementing MPC, the dynamic model of this system was linearized around the nominal setpoint $\mathbf{x}^{sp} = [19.7, 34.8, 59,68]$ and $\mathbf{u}^{sp} = [90, 128,117]$. The resulting system assumed the generic form a state space system with additive disturbance, $\mathbf{x}_{k+1} = A\mathbf{x}_k + B\mathbf{u}_k + \mathbf{w}_k$. Both standard and tube-based MPC were implemented using MPCtoolbox/CasADi, using a Matlab interface [111, 112]. Simulations for the non-deterministic system for 10 random choices of A and ω are shown in Figure 4.8. The initial state of the system was $\mathbf{x}_0 = [19, 34.7, 71, 65]$, and the setpoints were $\mathbf{x}^{sp} = [21.7, 34.8, 58, 67]$ and $\mathbf{u}^{sp} = [80, 119, 117]$.

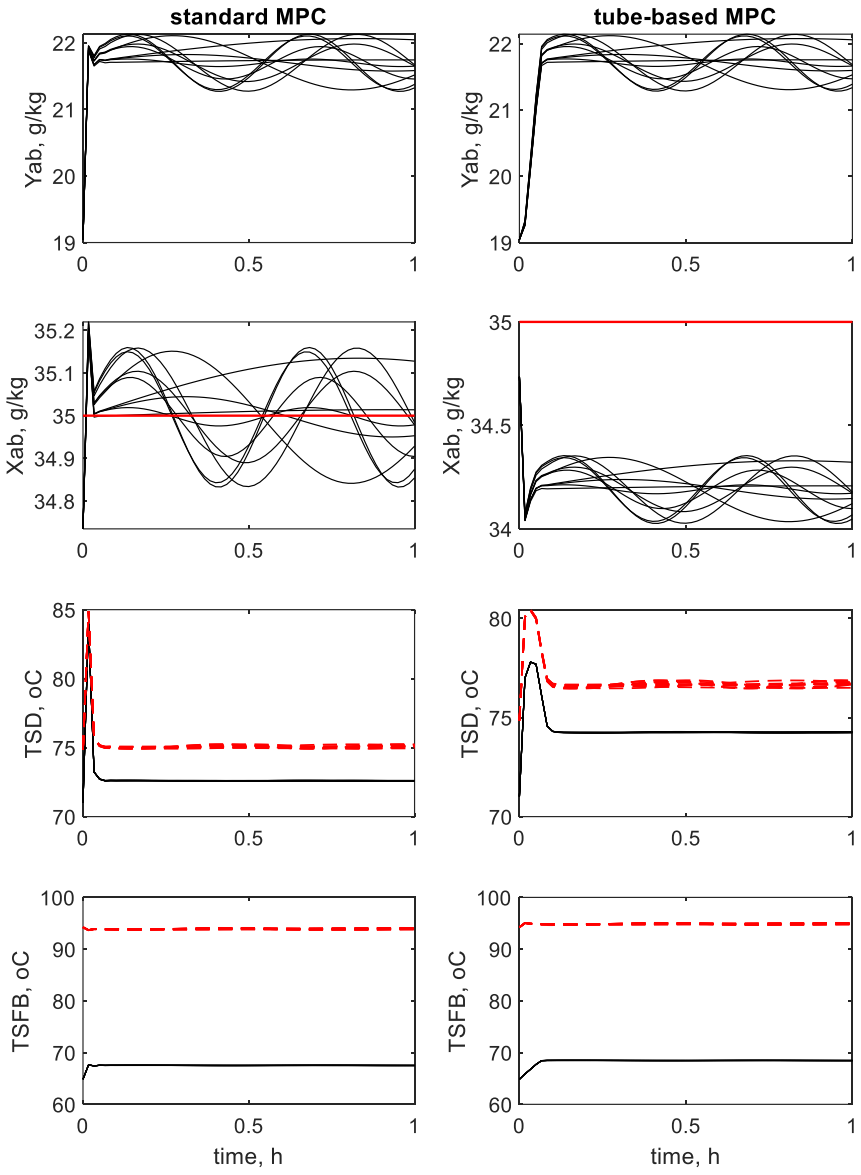

Figure 4.8 - Simulatiom of the state trajectory for a system affected by disturbances On the left, a standard MPC was implemented. On the right, a tube-based MPC was implemented

In this problem, we assume that the only independent setpoint is the feed flow rate, F_f^{sp}. The remaining setpoints are defined as steady states of the system meeting all control and state constraints previously defined, while minimizing the air temperature setpoints T_{main}^{sp} and T_{sfb}^{sp}. The system was then simulated for 1000 sample points $S_i =$

$\left(\mathbf{x}_i^{sp}, \mathbf{u}_i^{sp}, \mathbf{x}_{0i}\right)$, and the simulation data for $\mathbf{y}_i^{\mathbf{std}} = (\tilde{\mathbf{x}}_i, \bar{\mathbf{x}}_i^s, \bar{\mathbf{u}}_i^s)$, $\mathbf{y}_i^{\mathbf{tube}} = (\tilde{\mathbf{x}}_i, \bar{\mathbf{x}}_i^s, \bar{\mathbf{u}}_i^s)$, $\mathbf{f}_i^{std}$ and $\mathbf{f}_i^{tube}$ was stored. The data sets $\mathbf{S} = \{S_1, S_2, \dots S_{1000}\}$, $\mathbf{Y}^{std} = \{\mathbf{y}_1^{std}, \mathbf{y}_2^{std}, \dots, \mathbf{y}_{1000}^{std}\}$, $\mathbf{Y}^{tube} = \{\mathbf{y}_1^{tube}, \mathbf{y}_2^{tube}, \dots, \mathbf{y}_{1000}^{tube}\}, \mathbf{F}^{std} = \{\mathbf{f}_1^{std}, \mathbf{f}_2^{std}, \dots, \mathbf{f}_{1000}^{std}\}$ and $\mathbf{F}^{tube} = \{\mathbf{f}_1^{tube}, \mathbf{f}_2^{tube}, \dots, \mathbf{f}_{1000}^{tube}\}$ were used to build surrogate models of the system, that were further incorporated in the scheduling model given by Eq. 15.

$$\min_{\mathbf{u}_t^{sp}} \sum_t p^f \cdot \bar{u}_t^1 \cdot \tau + \sum_t p_t^h \Delta H \tag{15a}$$

$$I_t = I_{t-1} + \bar{u}_t^1 \cdot \tau - D_t, \qquad \forall t \tag{15b}$$

$$I_0 = 0 \tag{15c}$$

$$0 \leq I_t \leq \bar{I}, \qquad \forall t \tag{15d}$$

$$\Delta h_{da,main,t}^a = \int_{TSD}^{\bar{T}_{main,t}} C_{p,da}\,(T) dT \tag{15e}$$

$$\Delta h_{v,main,t}^a = \int_{TSD}^{\bar{T}_{main,t}} C_{p,v}\,(T) dT \tag{15f}$$

$$\Delta h_{da,t}^b = \int_{TSD}^{\bar{T}_{sfb,t}} C_{p,da}\,(T) dT \tag{15g}$$

$$\Delta h_{v,t}^b = \int_{TSD}^{\bar{T}_{sfb,t}} C_{p,v}\,(T) dT \tag{15h}$$

$$\Delta H_t = F_{main}(\Delta h_{da,main,t}^a + Y_{amb} \Delta h_{v,main,t}^a) + F_{sfb}(\Delta h_{da,t}^b + Y_{amb} \Delta h_{v,t}^b) \tag{15i}$$

The scheduling problem for the spray dryer consists of selecting setpoints $\mathbf{u}_t^{sp}$ in order to minimize the costs of operating the system. A single product is produced by the spray dryer, and the production flow can vary over time. The cost related to the raw material consumption, as well as costs related to energy consumption of the system. The energy consumption is directly related to the spray dryer inlet air temperatures, since these

temperatures must be heated in an energy intensive operation prior to SD feeding step. The energy price profile is shown in Figure 4.9. Equations (15e) to (15i) represent the energy consumption calculations, and the parameters for these equations were obtained from Green and Perry [67]. The energy prices vary at every scheduling slot of duration $\tau = 1$ hour. Therefore, it is expected that production levels will rise whenever the energy consumption cost is low, building up inventory to be consumed when energy prices spike. Equations (15b) to (15d) set material balances for the storage unit, inventory capacity constraints and initial inventory levels. A 1-week scheduling horizon is considered, with scheduled deliveries D_t at the end of each day, i.e., $t \in \{24, 48, 72, 96, 120, 144, 168\}$. We assume a steady demand of 900kg/day.

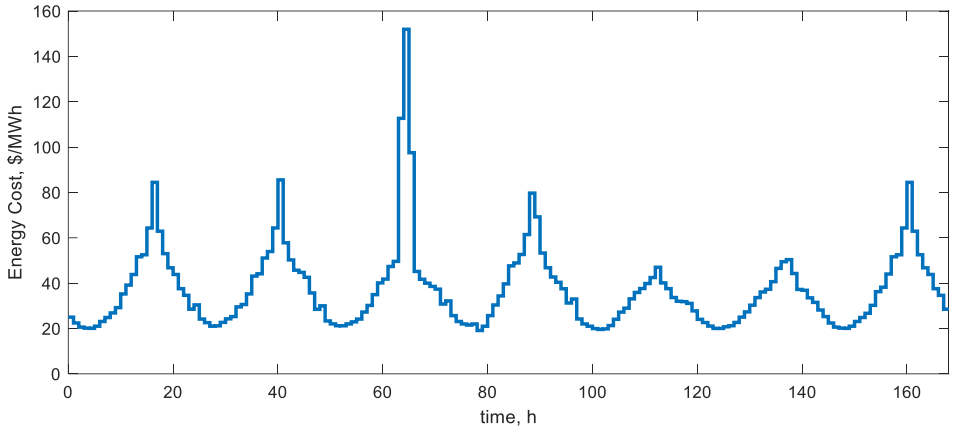

Figure 4.9 - The energy cost profile

Note that $\bar{\mathbf{u}}_t = \left[\bar{F}_{f,t}, \bar{T}_{main,t}, \bar{T}_{sfb,t}\right]$ are needed for scheduling calculations. Predictors for these variables are built based on the sets of data generated in the simulation, as well as a classification model that determines the feasible space of operation of standard and tube-based MPC. The surrogate models are incorporated in the scheduling problem, and the problem is solved to local optimality using GAMS IPOPT on a 64-bit Windows

system with Intel Core i7-6700 CPU at 2.60 GHz and 8GB RAM. The optimization results are shown in Table 4.4.

The solution of the optimization problem is implemented in the control simulation, and the results are shown in Table 4.4 and Figure 4.10. As in case study 1, differences between actual and predicted costs can be noted, and such differences are attributed to inaccuracies of the neural network regressors and disturbances affecting the system. The profile of the state variables reflects the variations in production flows, which decreases whenever spikes on the energy costs can be observed. The results show that, while robust MPC is seem as a conservative control solution when compared to deterministic MPC, the differences from a scheduling and economical point of view may be insignificant. Furthermore, while tube-based MPC manages to avoid constraints violation throughout the entire operation, standard MPC fails to respect the quality requirements of the finished product, i.e. $X_{ab} \leq 35$ g/kg at several points. It was expected that the feasibility constraints in the integrated problem would avoid such violations. However, uncertainties can make the feasible region of standard MPC unpredictable.

Table 4.4 - Optimal schedule results for case study 2

Case	Solution Time	Predicted Cost ($)	Actual cost ($)	Difference (%)
Standard	72 s	$ 6609.89	6627.43	0.27%
Robust	60 s	$ 6618.71	6626.93	0.12%

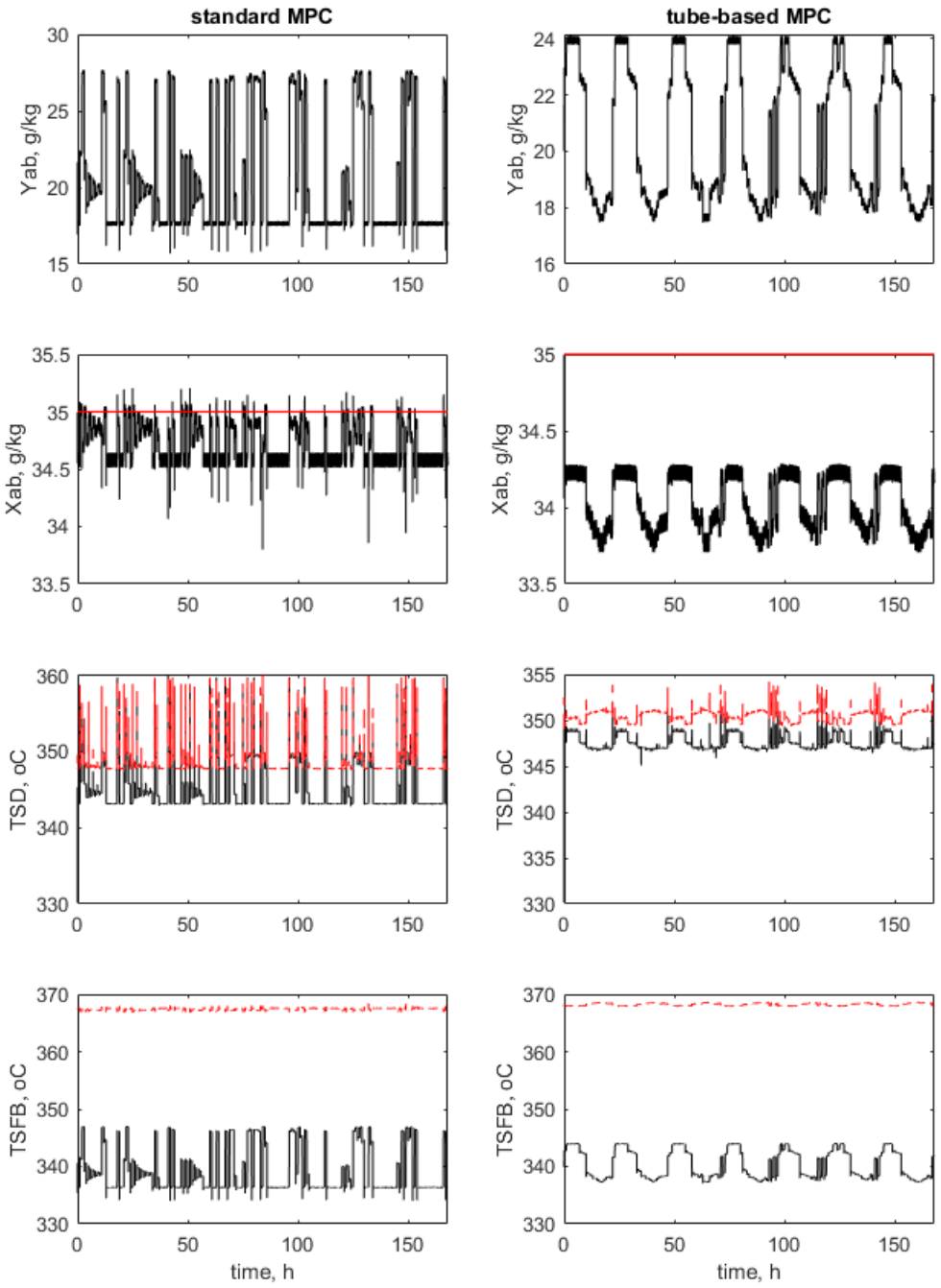

Figure 4.10 - The state behavior of the system in a 168-hour simulation

Two control strategies (standard MPC and tube-based MPC) are tracking setpoints defined by the scheduling optimization

4.6 Conclusions

In this work, a framework for the integration of scheduling and control problems was proposed. Uncertainties at the control level of the problem were considered, and two process control strategies were implemented: a standard Model Predictive Control strategy, and a tube-based Model Predictive Control strategy. Tube-based MPC implicitly takes into account the effect of disturbances and the feedback nature of the control when making control decisions. Therefore, tube-based MPC can cope with disturbances affecting the system, and presents a superior performance when compared to standard MPC regarding constraint violations. Both MPC strategies were used to generate a data set representative of historical operation of the systems in consideration. The data sets were then used to train neural networks to approximate functions $\phi^{\tilde{x}}, \phi^{\bar{x}}, \phi^{\overline{u}}$ and ξ, capturing the closed loop behavior of scheduling relevant variables, as well as the feasible space of operation of the control system. The neural network predictors were incorporated in the scheduling problem, and two optimization problems were solved for each case study: the first problem integrating scheduling and standard MPC, and the second problem integrating scheduling and robust MPC. The optimization solutions were then implemented in the control layer of the problem, and the system was simulated for the entire scheduling horizon. The simulation results clearly show the advantages of utilizing a robust approach in relation to the uncertainties at the control level, from both scheduling and control perspectives.

This work extends the concept of simulation-optimization for integration of decision-making processes, first introduced in Chapter 2. To address limitations of the previous approach, the simulation-optimization is solved using surrogate models. By obtaining an algebraic model for the input-output relationship for the control level

simulation, we expand the variety of problems that can be solved through the framework, opening the possibility of including discrete decisions in the integrated problem, and the possibility of using a variety of optimization algorithms that could not be employed in the previous framework

Future works should focus on handling dimensional problems, taking advantage of techniques that the machine learning field provides, specially related to rigorous approaches for feature selection in the surrogate training step of the framework. Future works should also address uncertainties in multiple levels of the integrated problem, as well as the uncertainties related to the surrogate models.

Nomenclature

Control Problem

α	Tightening state constraint set factor
β	Tightening input constraint set factor
Δ	Control sample time
k	Discretization steps of the control problem
N	Prediction horizon of the control problem
P_f	Terminal penalty matrix
$\mathbb{P}_N$	Tube-based control problem
$\overline{\mathbb{P}}_N$	Nominal control problem
Q	State penalty matrix
R	Input penalty matrix
$\mathbf{u}_k$	Control input (manipulated variables) vector
$\mathbf{u}^{sp}$	Input setpoints
$\mathbf{U}$	Input sequences for $N-1$ time steps
$\mathbb{U}$	Input constraint set
$\mathbf{v}_k$	Nominal input vector
$\mathbf{v}^{sp}$	Nominal input setpoint
$\mathbb{V}$	Nominal input constraint set
$\mathbf{w}$	Disturbances at control problem
$\mathbf{x}_{initial}$	Current state of the system
$\mathbf{x}_k$	State vector
$\mathbf{x}^{sp}$	State setpoints

$\mathbf{x}_N$	Terminal state of the system
$\mathbb{X}$	State constraint set
$\mathbf{z}_k$	Nominal state vector
$\mathbf{z}^{sp}$	Nominal state setpoint
$\mathbb{Z}$	Nominal state constraint set

Neural Networks

$\mathbf{b}_{input}$	Bias in the input layer
$\mathbf{b}_h$	Bias in the output layer
$Loss$	Loss function
O	Number of nodes in the hidden layer
R	Predictor
$\mathbf{w}_{input}$	Weights in the input layer
$\mathbf{w}_h$	Weights in the output layer

Integrated Problem

ξ	Classification model approximating feasible region of operation of control problem
τ	Duration of each scheduling slot
ϕ	Surrogate models
ψ	Control law
$Cost_t$	Operation cost at time t
k	Control discretization periods

K	Number of control steps k within a scheduling slot t
l	Scheduling constraints
t	Scheduling discretization periods
$\mathbf{x}_{kt}$	State vector at step k in scheduling slot t
$\mathbf{x}_t^{sp}$	State setpoints for scheduling slot t
$\tilde{\mathbf{x}}_t$	State vector estimation at scheduling slot t
$\bar{\mathbf{x}}_t$	Average state values at scheduling slot t
$\bar{\mathbf{x}}_t^s$	Average state values for a subset s of variables at scheduling slot t
$\mathbb{X}$	State constraint set
$\mathbf{u}_{kt}$	Input vector at step k in scheduling slot t
$\mathbf{u}_t^{sp}$	Input setpoints for scheduling slot t
$\tilde{\mathbf{u}}_t$	Input vector estimation at scheduling slot t
$\bar{\mathbf{u}}_t$	Average input values at scheduling slot t
$\bar{\mathbf{u}}_t^s$	Average input values for a subset s of variables at scheduling slot t
$\mathbb{U}$	Input constraint set

Chapter 5

Integration of Planning, Scheduling and Control

Abstract: In this chapter, we present a framework to achieve integration of planning, scheduling and control problems. The framework builds upon the concepts of data-driven feasibility analysis and surrogate models discussed in the previous chapter, as well as the concept of feature selection. We apply the proposed methodology to a complex case study, involving the optimization of an enterprise of air separation plants.

5.1 Introduction

The ultimate goal of enterprise-wide optimization is the integration of planning, scheduling and control decisions. Few works in the literature attempted to address this problem, which poses challenges related to the dimensionality and the complexity of the resulting model. In this chapter, we present a systematic framework to achieve the overall integration of decision-making processes. The framework consists of formulating the integrated problem as a grey-box optimization problem, and using data-driven feasibility analysis and surrogate models to approximate the unknown black-box constraints. We follow a systematic procedure to achieve this integration, consisting of two building blocks (Figure 5.1): first, we address the integration of scheduling and control. Next, we address the integration of planning and scheduling. To handle dimensionality issues, we introduce the concept of feature selection when building the surrogate models. The methodology is applied to the optimization of an enterprise of air separation plants.

This chapter is organized as follows. In section 5.2, a background in grey-box optimization and feature selection methodologies is provided, building the theoretical basis

of the proposed framework. In section 5.3, the problem in consideration is summarized, and section 5.4 presents the methodology proposed to integrate planning, scheduling and control problems. The performance of the proposed framework is demonstrated through case studies in section 5.5. Final conclusions and future work are discussed in section 5.6.

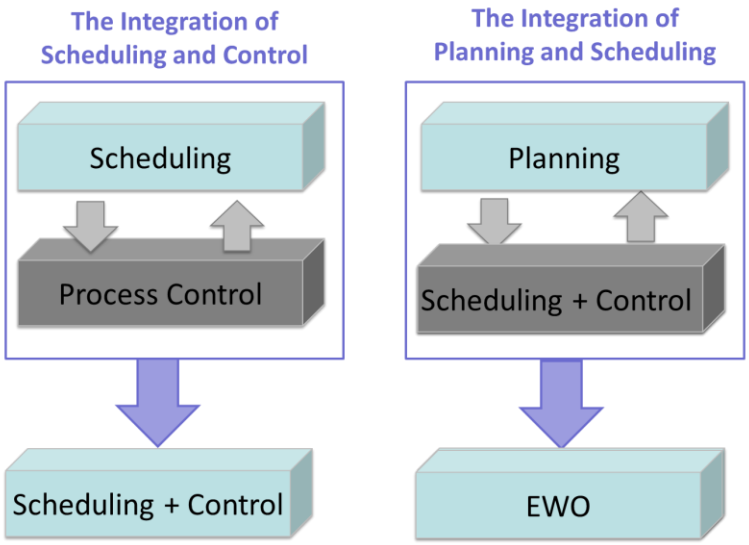

Figure 5.1 - Building blocks for enterprise-wide optimization

5.2 Background

5.2.1 Grey-box optimization

Grey-box problems are characterized by partial or total lack of closed-form equations describing the constraints and the objective of the problem. Such problems arise in a variety of fields including chemical engineering, geosciences, financial management, molecular engineering and aerospace engineering [114]. Grey-box problems primarily rely on expensive simulations, input/output data, or phenomena which have not yet been defined by physics-based mathematical equations.

A general constrained grey-box nonlinear optimization problem is defined as:

$$
\begin{aligned}
&\min_{x} f(\mathbf{x}) \\
&s.t.\, g_m(\mathbf{x}) \leq 0, \qquad \forall m \in \{1, \dots, M\} \\
&g_k(\mathbf{x}) \leq 0, \qquad \forall k \in \{1, \dots, K\} \\
&\mathbf{x} \in \mathbb{X}
\end{aligned}
\tag{1}
$$

where K represents the total number of constraints with known closed-form while M represents the total number of constraints with unknown closed-form. Since no explicit equations for $g_m(\cdot)$ are available, the direct use of deterministic optimization algorithms based on analytical C^2 functions is prohibitive.

The general formulation in Eq. (1) can be adopted in the integration of decision-making processes as follows: in the integration of scheduling and control, scheduling constraints take the place of $g_k(\cdot)$ while closed-loop control behavior of the dynamic system takes the place of $g_m(\cdot)$. In the integration of planning and scheduling, planning constraints take the form of $g_k(\cdot)$, while closed-loop scheduling-control behavior takes the place of $g_m(\cdot)$. In this work, surrogate models and data-driven feasibility methodologies will be used to approximate $g_m(\cdot)$, which provides an algebraic expression for such equations and allows the use of deterministic optimization algorithms to solve problem 1. In a sense, the grey-box structure has been implicitly utilized in the previous chapters. However, by explicitly defining the integration of decision-making processes as a grey-box optimization problem, we expand the class of strategies that can be used to solve the integrated problem. A variety of strategies have been proposed in the literature in the field of grey-box optimization [115-119], and the integration of decision-making can greatly benefit from the implementation of theses strategies.

5.2.2 Feature selection

Feature selection methodologies are the focus of much research in the area of machine learning applications. Feature selection can be viewed as the systematic selection of a subset of variables $\mathbf{z} \in \mathbb{Z} \subseteq \mathbb{X}$ that are most relevant to predicting y. The objectives of feature selection is three-fold: improving the prediction performance, providing faster and more cost-efficient predictors, and providing a better understanding of the underlying process that generated the data [110]. A wide variety of feature selection procedures has been proposed in the literature, and they are usually classified in filter methods, wrapper methods and embedded methods.

Filter methods apply statistical tests to the dataset in order to quantify the correlation between a variable (or attribute) and the output y in consideration [120]. They are a pre-processing step in the predictor training process, and they are independent of the choice of the predictor. Examples of statistical tests include the Pearson correlation, Fisher's criterion and mutual information estimation. Filter methods are usually univariate, i.e., they consider features independently, one at a time. After a statistical test has been applied to the dataset, a score is assigned to each feature, and a subset of features with high scores are selected to be further used in the training of the predictor (surrogate model). The ranking of features may not be optimal. However, filter methods are efficient (since they only require the computation of d scores, where d is the total number of features); and statistically scalable [121].

Wrapper methods utilize the learning machine (the predictor, or the surrogate) of interest to score subsets of variables according to their usefulness to a given predictor. They essentiality treat the machine as a black-box, and use a greedy search strategy to include or

remove variables into the subset of selected features according to the performance of the predictor. Performance is usually measure using a validation set or by cross-validation. Wrapper methods usually achieve better recognition rates when compared to filter methods, since they are tuned to the specific interactions between the predictors and the dataset. However, wrapper methods are computationally expensive and can become infeasible when applied to datasets with large number of features. In this work, a forward selection wrapper methodology is implemented for feature selection, and the methodology is further described in section 5.4.

Embedded methods perform variable selection in the process of training and are usually specific to given learning machines. An example of embedded method is the use of the sensitivity of the objective function used for machine training purposes to a determined feature x_i. The subset of features with highest impact in the objective function is selected as the final set to be used in predictions [110].

It is important to note the differences between feature selection and dimensionality reduction strategies, such as principal component analysis and singular value decomposition. Both methods seek to reduce the number of attributes in a dataset. However, dimensionality reduction techniques do so by creating new combinations of attributes, while feature selection techniques include and exclude attributes from a dataset without modifying them. Therefore, while dimensionality reduction techniques may preserve a higher level of information from the original dataset, feature selection techniques preserve the identity and the physical meaning of each attribute. Therefore, feature selection provides a useful way of reducing the dimensionality of a problem while also providing insights of the underlying physical process.

5.3 Problem definition

Consider an air separation unit that produces gas nitrogen (GN2), gas oxygen (GO2), and liquid nitrogen (LN2). The unit operates by first compressing ambient air in a large multistage compressor, followed by removal of water, carbon dioxide and hydrocarbons, and by cooling in a multi-stream heat exchanger. This air feed mixture of oxygen, nitrogen and argon is then split into two substreams. The first stream consists of pure air entering the bottom of the high-pressure column (MA) and the second stream consists of expanded air entering the 8^{th} tray of the low-pressure column (EA). The high-pressure column (HPC) contains 40 trays and operates at 6.2-6.4 bars, while the low-pressure column operates at 1.2-1.35 bars and also contains 40 trays. The reboiler of the low-pressure column is integrated with the condenser of the high-pressure column. The main products of the high-pressure column are pure nitrogen (GN2) (99.99%) and crude liquid oxygen (~50%). The crude oxygen stream is fed into the 20^{th} tray of the low-pressure column. A high purity separation is achieved in the low-pressure column, leading to nitrogen gas (GN2) with ~99.9% purity, liquid nitrogen (LN2) with ~99.9% and oxygen (GO2) with ~95% purity as products. A schematic of the air separation unit is given in Figure 5.2.

The air separation unit operates in a continuous mode and is capable of flexible operation, in the sense that the production rate (defined in terms of the flow rate of the gas nitrogen product) can assume any value within a given range. Furthermore, the operation of the air separation unit consumes significant amounts of electricity, specially related to the unit operations of compression. It is assumed the electricity needed for the plant operation is purchased in a day-ahead market, for which accurate price forecasts for an

entire week are available. The electricity prices in the day-ahead market fluctuate in an hourly basis.

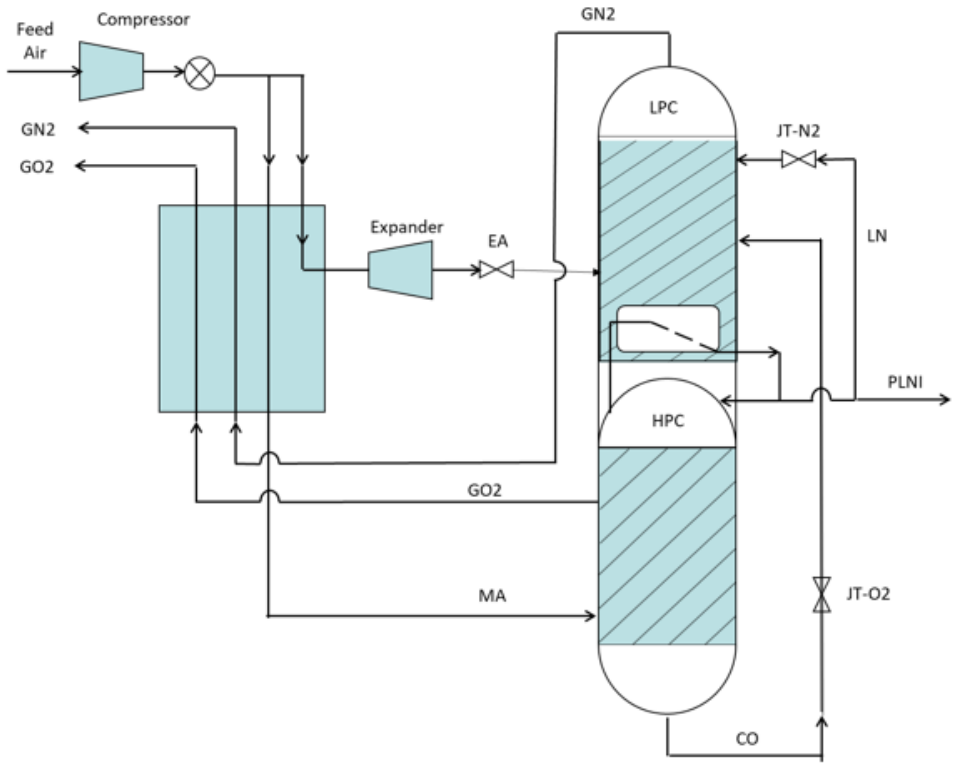

Figure 5.2 - Air separation unit schematic

We aim to solve a **scheduling problem** which takes into consideration the electricity prices fluctuations to optimize the operation of an air separation plant. We assume that a storage system is available for the finished liquid product; gas nitrogen can be liquefied and further vaporized to meet gas demand; and the production targets for each product (GN2, GO2, LN2) are given. The production targets for gas products are given as a steady flow rate that must be satisfied throughout the entire schedule, while the production targets for the liquid nitrogen should be met by the end of the scheduling horizon. A scheduling horizon of one week is considered, and it is discretized in hourly periods. Production rates can fluctuate (thereby changing the electricity demand of the

process) and the product demand can be met either directly (i.e., by the plant itself) or by a combination of plant production and depleting the inventory of stored product. Therefore, the goal of the scheduling problem is to define hourly production rates setpoints that minimize operation costs, which achieving the given production targets. A simplified network of the air separation process is given in

The scheduling problem is highly dependent on the control performance of the system: the energy consumption varies with the flows of main air MA and expanded air EA; and the production rates of GN2, LN2 and GO2 vary with the initial state of the system, control actions, and changes in the setpoints. Therefore, it is essential to consider the closed-loop control behavior, and solved an **integrated scheduling and control** problem in order to define the overall optimal production cost.

Once the scheduling and control problem is formulated, we aim to solve a **planning problem** for an enterprise of three air separation plants. Each plant produces high purity nitrogen and oxygen in cryogenic distillation columns, at different production rates and subject to different electricity contracts. Each plant is associated to one liquid nitrogen storage unit, $L1$, $L2$ and $L3$. The enterprise delivers gas and liquid products to six customers, and the possible routes between plant/storage facility and customers are shown in Figure 5.3.

We aim to solve a planning problem for a horizon of 52 weeks. Demand forecasts for each customer in the enterprise is available. It is assumed that the delivery of liquid products happens at the end of each week, and the demand of gas products from each customer remains constant within the week horizon. Furthermore, the average cost of electricity prices for each week and each plant is available. Note that, while the scheduling

problem considers electricity prices varying in an hourly basis, the planning problem aggregates this information in as a weekly energy cost. We aim to determine production targets and inventory levels for each air separation plant and storage facility, at each week of the planning horizon.

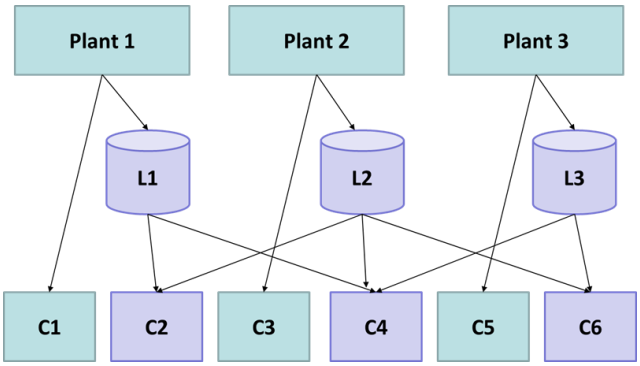

Figure 5.3 – Enterprise network

In order to solve the planning model, some essential information from the scheduling problem is needed: (i) the planning problems requires an estimate of the energy consumption in each production plant, as a function of the production targets; (ii) the planning problem also requires information about the feasible space of the scheduling problem as a function of production targets. The energy consumption and the energy price forecasts are used to estimate the costs of operating the enterprise, which is minimized at the planning level. The planning problem attempts to determine feasible production targets, given the scheduling information.

Since the planning problem depends on the scheduling behavior, which in its turn depends on the dynamic behavior of the system, we aim to formulate an integrated **planning, scheduling and control** problem to determine the optimal operation of the enterprise. In the next section, a systematic framework for the integration of planning, scheduling and control problems is proposed.

5.4 Framework for the integration of planning, scheduling and control

The systematic methodology for the integration of planning, scheduling and control includes two building blocks. First, the integration of scheduling and control is performed by treating the control problem as a black box. Input-output data is obtained by simulating the control problem. Algebraic expressions for unknown constraints are obtained using feasibility analysis and surrogate models. Therefore, an algebraic model for the scheduling-control problem is obtained, and can be solved to optimality. In the second building block, the scheduling-control problem is treated as a black box, as it is used to generate input-output data to be incorporated in the planning problem. Algebraic expressions to represent the scheduling-control behavior are obtained and incorporated in the planning problem, providing a model for enterprise-wide optimization.

The implementation of each building block follows five steps, shown in Figure 5.4. First, a higher-level optimization problem is formulated. In other words, the mathematical model for the scheduling problem is defined (in the problem of integrating scheduling and control), or the mathematical model for the planning problem is defined (in the overall integration). Once the problem is formulated, the unknown constraints and all the information needed from the black box problem is identified. The unknown constraints will include feasibility equations and equality constraints. In the third step, a subset of features is selected, followed by the identification of surrogate models. Step 3 and 4 may have an iterative aspect, depending on the feature selection procedure of choice. Finally, in the last step, the surrogate models are incorporated in the optimization problem, which is solved to (local) optimality.

In the next subsections, each step of the proposed framework is described in detail. We first describe the block of integrating scheduling and control, followed by the block of integration of the overall integration.

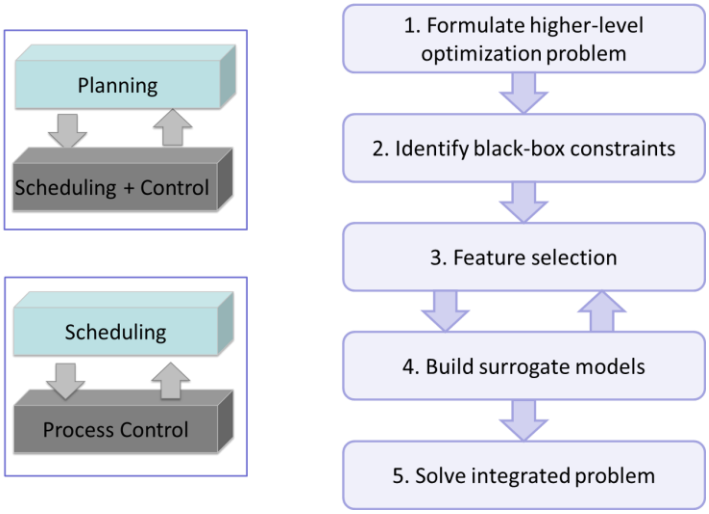

Figure 5.4 - Framework for the integration of planning, scheduling and control

5.4.1 Scheduling and control integration

5.4.1.1 Higher-level optimization problem and black-box constraint identification

A general formulation for the scheduling model was presented in section 4.3:

$$\min_{\mathbf{x}_t^{sp},\mathbf{u}_t^{sp}} \sum_t Cost_t(\bar{\mathbf{x}}_t^s, \bar{\mathbf{u}}_t^s) \tag{2a}$$

$$l(\bar{\mathbf{x}}_t^s, \bar{\mathbf{u}}_t^s) = 0 \tag{2b}$$

$$\tilde{\mathbf{x}}_{t+1}^0 = \tilde{\mathbf{x}}_t, , \tilde{\mathbf{u}}_{t+1}^0 = \tilde{\mathbf{u}}_t \tag{2c}$$

$$\tilde{\mathbf{x}}_t = \phi^{\tilde{x}}(\mathbf{x}_t^{sp}, \mathbf{u}_t^{sp}, \tilde{\mathbf{x}}_t^0, \tilde{\mathbf{u}}_t^0) \tag{2d}$$

$$\tilde{\mathbf{u}}_t = \phi^{\tilde{u}}(\mathbf{x}_t^{sp}, \mathbf{u}_t^{sp}, \tilde{\mathbf{x}}_t^0, \tilde{\mathbf{u}}_t^0) \tag{2e}$$

$$\bar{\mathbf{x}}_t^s = \phi^{\bar{x}}(\mathbf{x}_t^{sp}, \mathbf{u}_t^{sp}, \tilde{\mathbf{x}}_t^0, \tilde{\mathbf{u}}_t^0) \tag{2f}$$

$$\bar{\mathbf{u}}_t^s = \phi^{\bar{u}}(\mathbf{x}_t^{sp}, \mathbf{u}_t^{sp}, \tilde{\mathbf{x}}_t^0, \tilde{\mathbf{u}}_t^0) \tag{2g}$$

In this formulation, Eq. (2a) represents the scheduling cost, as a function of the state and input variables of the system. Eq. (2b) represents scheduling constraints, and ensures that demands are met at all times, and that the amount of product stored does not deplete/exceed the physical capacity of the storage system. $l(\cdot)$ may also involve discrete decisions such as the assignment of products to scheduling slots, or the assignment of equipment to tasks. Equation (2c) links the final state of the system at scheduling period t, to the initial state of the system at scheduling period $t+1$. Equations (2d) to (2g) are related to the prediction of state **x** and input **u** variables which are relevant to the scheduling problem.

We now derive the scheduling model for the specific problem of scheduling of air separation units, following the format of equation (2) and based on the following process network:

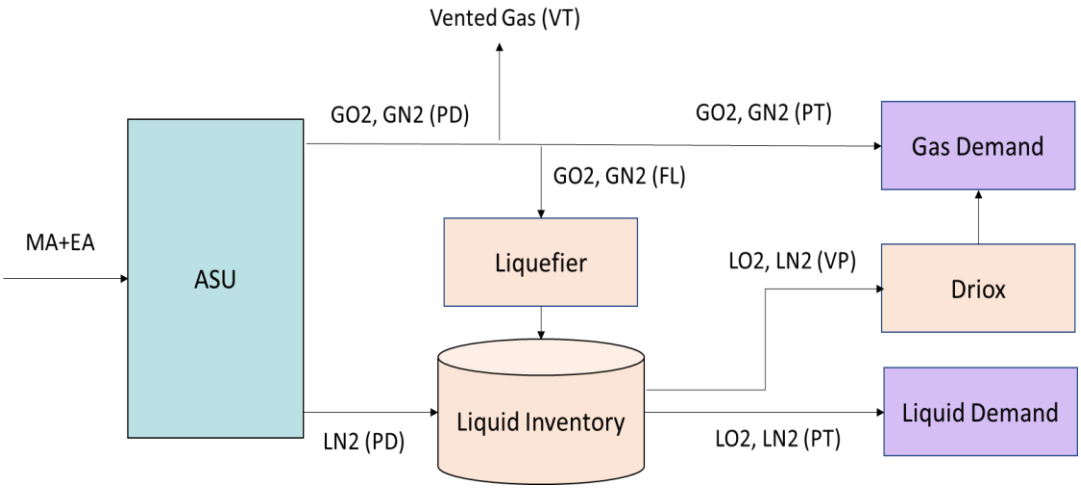

Figure 5.5 - ASU process network

In the case of operation of an air separation plant, the scheduling cost can be written as:

$$\min_{\mathbf{x}^{sp},y} SC = \sum_{t} e_t \tau(\gamma_c(\overline{MA}_t + \overline{EA}_t) - \gamma_t \overline{EA}_t + \gamma_L FL_{kt}) \tag{3}$$

in which $\overline{MA}_t$ is the average flow of main air into the distillation column and $\overline{EA}_t$ is the average flow of expanded air. γ_c, γ_t and γ_L are parameters (with units Watts/(mol/s)) which relate the flow rate in the compressor, turbine and liquefier, respectively, to the amount of electricity consumer (or generated) by the unit operation. e_t is the forecasted electricity price throughout the scheduling horizon, and τ is the scheduling discretization time.

Mass balances for the scheduling problem, derived according to the process network in Figure 5.5, are given by Eq. (4a) – (4b).

$$I_{kt} = I_{k,t-1} + \overline{PD}_{kt} + \rho_k^{liq} FL_{kt} - T_{kt} - VP_{kt}, \qquad \forall k \in K^L, t \in T \tag{4a}$$

$$\overline{PD}_{kt} + \rho_k^{dri} VP_{kt} - VT_{kt} - \rho_k^{liq} FL_{kt} - T_{kt} = 0, \qquad \forall k \in K^G, t \in T \tag{4b}$$

The ASU produces liquid and gaseous products, which are defined by the product sets K^L and K^G respectively, that is, $K^L = \{LN2\}$ and $K^G = \{GO2, GN2\}$. $\overline{PD}_{kt}$ is the average product flow from the air separation unit for product k at for the entire scheduling period t. FL_{kt} is the flow of products into and out of the liquefier. VP_{kt} is the flow products which are vaporized to satisfy the production target T_{kt}. VT_{kt} is the flow of gas nitrogen and oxygen, vented to the ambient. ρ_k^{dri} and ρ_k^{liq} take care of the conversions between liquid and gaseous flows. Finally, I_{kt} are the inventory levels of liquid products for products $k \in K^L$ at scheduling period t. The inventory levels are further constrained with equations (5a)-(5c). There is no storage for gaseous products $k \in K^G$.

$$I_{k0} = I_k^{ini}, \qquad \forall k \in K^L \tag{5a}$$

$$I_{k,tf} \geq I_k^f, \qquad \forall k \in K^L \tag{5b}$$

$$I_k^l \leq I_{kt} \leq I_k^u, \qquad \forall k \in K^L, t \in T \tag{5c}$$

Equation (5a) states the initial inventory level for products $k \in K^L$. Equation (5b) ensures that the final inventory level is above certain value I_k^f, avoiding the depletion of stored products at the end of the scheduling horizon. I_k^l and I_k^u define the upper and lower bounds for the product storage.

To allow different modes of operation in the air separation plant, binary variables y_{mt} are introduced. The modes include "on", "off", and "startup". y_{mt} takes the value of 1 is the plant is operating at mode m in scheduling period t. The associated constraints are given by Eq. (6a)-(6c), adopted from Zhang, Sundaramoorthy [68]

$$\sum_m y_{mt} = 1, \qquad \forall t \tag{6a}$$

$$\sum_{m' \in TR_m^f} z_{m',m,t-1} - \sum_{m' \in TR_m^i} z_{m,m',t-1} = y_{m,t} - y_{m,t-1}, \qquad \forall m, t \tag{6b}$$

$$y_{m\prime t} \geq \sum_{k=1}^{\theta_{mm'}} z_{m,m',t-k}\,, \qquad \forall i, (m,m') \in TR_i, t \tag{6c}$$

Equation (6a) states that the air separation plant can operate in one operation mode m at every scheduling period t. Equation (6b) introduces binary variables $z_{m',mt}$, which define the transitions between operating modes. A transition can only occur if $(m,m') \in TR$, with TR being the set of all possible transitions. Furthermore, $TR_m^f = \{m' : (m',m) \in TR\}$ and $TR_m^i = \{m' : (m,m') \in TR\}$. The restriction that a process has to remain in a certain mode for a minimum amount of time is expressed with equation (6c).

The ASU scheduling constraints (3) – (6) replace constraint (2b) from the general scheduling formulation. An analysis of these constraints shows a clear relationship with

state and manipulated variables of the system, $\overline{PD}_{kt}$, $\overline{MA}_{kt}$, $\overline{EA}_{kt}$. Suitable approximation of these variables should be obtained, and they can be initially expressed as:

$$\tilde{\mathbf{x}}_{t+1}^{0} = \tilde{\mathbf{x}}_{t}, , \tilde{\mathbf{u}}_{t+1}^{0} = \tilde{\mathbf{u}}_{t} \tag{7a}$$

$$\tilde{\mathbf{x}}_{t} = \phi^{\tilde{x}}(\mathbf{x}_{t}^{sp}, \mathbf{u}_{t}^{sp}, \tilde{\mathbf{x}}_{t}^{0}, \tilde{\mathbf{u}}_{t}^{0}) \tag{7b}$$

$$\tilde{\mathbf{u}}_{t} = \phi^{\tilde{u}}(\mathbf{x}_{t}^{sp}, \mathbf{u}_{t}^{sp}, \tilde{\mathbf{x}}_{t}^{0}, \tilde{\mathbf{u}}_{t}^{0}) \tag{7c}$$

$$\overline{PD}_{kt} = f(\mathbf{x}_{t}^{sp}, \mathbf{u}_{t}^{sp}, \tilde{\mathbf{x}}_{t}^{0}, \tilde{\mathbf{u}}_{t}^{0}, y_{mt}) \tag{7d}$$

$$MA_{t} = f\left(\mathbf{x}_{t}^{sp}, \mathbf{u}_{t}^{sp}, \tilde{\mathbf{x}}_{t}^{0}, \tilde{\mathbf{u}}_{t}^{0}, y_{mt}\right) \tag{7e}$$

$$EA_{t} = f\left(\mathbf{x}_{t}^{sp}, \mathbf{u}_{t}^{sp}, \tilde{\mathbf{x}}_{t}^{0}, \tilde{\mathbf{u}}_{t}^{0}, y_{mt}\right) \tag{7f}$$

In the next steps, a subset of variables $\mathbf{z} \subset \left\{\mathbf{x}_{t}^{sp}, \mathbf{u}_{t}^{sp}, \tilde{\mathbf{x}}_{t}^{0}, \tilde{\mathbf{u}}_{t}^{0}\right\}$ will be defined and the predictors for $\overline{PD}_{kt}$, $\overline{MA}_{t}$, $\overline{EA}_{t}$ and $\mathbf{z}$ will be obtained.

5.4.1.2 Feature selection and surrogate models

To identify a subset of features $\mathbf{z} \subset \left\{\mathbf{x}_{t}^{sp}, \mathbf{u}_{t}^{sp}, \tilde{\mathbf{x}}_{t}^{0}, \tilde{\mathbf{u}}_{t}^{0}\right\}$, we propose to use a sequential forward selection (SFS) method as the feature selection methodology, coupled with a simple linear regressor as the surrogate model. As it will be seen in section 5.5, the prediction accuracy of the linear regressor in the ASU problem was satisfactory, and therefore there is no need to employ more complex surrogate models that would later introduce nonlinearities to the integrated problem. The feature selection methodology will be employed using a set $\mathbf{S}$ of N data points, $\mathbf{S} = \{S_1, S_2, \dots, S_N\}$ where each $S_i = \left\{\mathbf{x}_{i}^{sp}, \mathbf{u}_{i}^{sp}, \tilde{\mathbf{x}}_{i}^{0}, \tilde{\mathbf{u}}_{i}^{0}\right\}, i = \{1,2 \dots., N\}$ is an input to a simulation of the control problem. The simulation of the control problem is performed for a horizon τ, where τ is equivalent to the discretization step in the scheduling problem. The simulation generates an output $Y_i = \{PD_{ki}, MA_i, EA_i, \tilde{\mathbf{x}}_{i+1}^{0}, \tilde{\mathbf{u}}_{i+1}^{0}\}$, and an output dataset $\mathbf{Y} = \{Y_1, Y_2, \dots Y_N\}$ is complied.

Following a similar approach, testing sets $\mathbf{S}^{test}$ and $\mathbf{Y}^{test}$ are generated to later evaluate the performance of the surrogate models.

The feature selection method is implemented next. SFS has two components:

1) An objective function, or criterion, which the methods seeks to minimize over all feasible feature subsets. The criterion employed by this work is mean squared error MSE.
2) A sequential search algorithm which adds features to a candidate subset until the addition of further features does not improves the criterion

SFS follows an iterative greedy search algorithm. Starting from an empty set $\boldsymbol{z} = \{\emptyset\}$, SFS adds the feature z^+ that results in the smallest criterion $J(\mathbf{z}, z^+)$ when combined with the features $\mathbf{z}$ that have already been selected. The $J(\mathbf{z}, z_i)$ is evaluated using k-fold cross validation and training the linear regression model k times for every candidate feature z_i. The procedure stops when the improvement in the criterion $J(\cdot)$ is smaller than a tolerance ϵ. This tolerance can be used as a tuning parameter in the integration of scheduling and control: small values of ϵ will result in better accuracy of the predictor, while selecting a larger number of features that will later increase the dimensionality of the integrated problem.

In this work, the feature selection and surrogate model training procedure was implemented in Matlab using the Statistics and Machine Learning Toolbox.

5.4.1.3 Integrated problem

Once the feature selection and surrogate model training procedure is finalized, the following model are formulated to take the place of Eq. (7):

$$\overline{PD}_{kt} \leq \mathbf{A}\mathbf{z}_t + \mathbf{b} + M(1 - y_{mt}), \qquad \forall k, t, m = \{\text{on}\} \tag{8a}$$

$$\overline{PD}_{kt} \geq \mathbf{A}\mathbf{z}_t + \mathbf{b} - M(1 - y_{mt}), \qquad \forall k, t, m = \{\text{on}\} \tag{8b}$$

$$\overline{MA}_t \leq \mathbf{C}\mathbf{z}_t + \mathbf{d} + M(1 - y_{mt}), \qquad \forall k, t, m = \{\text{on}, \text{startup}\} \tag{8c}$$

$$\overline{MA}_t \geq \mathbf{C}\mathbf{z}_t + \mathbf{d} - M(1 - y_{mt}), \qquad \forall k, t, m = \{\text{on}, \text{startup}\} \tag{8d}$$

$$\overline{EA}_t \leq \mathbf{E}\mathbf{z}_t + \mathbf{f} + M(1 - y_{mt}), \qquad \forall k, t, m = \{\text{on}, \text{startup}\} \tag{8e}$$

$$\overline{EA}_t \geq \mathbf{E}\mathbf{z}_t + \mathbf{f} - M(1 - y_{mt}), \qquad \forall k, t, m = \{\text{on}, \text{startup}\} \tag{8f}$$

$$\mathbf{z}_{t+1} \leq \mathbf{G}\mathbf{z}_t + \mathbf{h} + M(1 - y_{mt}), \qquad \forall k, t, m = \{\text{on}, \text{startup}\} \tag{8g}$$

$$\mathbf{z}_{t+1} \geq \mathbf{G}\mathbf{z}_t + \mathbf{h} - M(1 - y_{mt}), \qquad \forall k, t, m = \{\text{on}, \text{startup}\} \tag{8h}$$

$$\mathbf{z}_0 = \mathbf{z}^{initial} \tag{8i}$$

Note that binary variable y_{mt} is included in Equation (8) to account for the different modes of operation. M is a big value. Using similar constraints, The variables $\overline{PD}_{kt}, \overline{MA}_{kt}$ and $\overline{EA}_{kt}$ are set to zero if the operation mode at period t is "off". Furthermore, $\overline{PD}_{kt}$ is set to zero is the operation mode at period t is set to "startup".

The integrated scheduling and control problem can be summarized as:

Minimize Objective Function (2)

Subject to: Mass balances (Eq. 3)

Inventory constraints (Eq. 4) (P1)

Assignment constraints (Eq. 5)

Black-box constraints (Eq. 8)

The resulting integrated model is a mixed integer linear problem which is solved using GAMS/CPLEX. The integration of scheduling and control is achieved. Next, we discuss the second building block in enterprise optimization.

5.4.2 Overall integration of planning, scheduling and control problems

5.4.1.1 Higher-level optimization problem and black-box constraint identification

In the overall integration, the planning problem is the "higher-level" problem, while the integrated scheduling and control problem are treated as a black-box. A general formulation for planning problems was given in section 3.2.2. Here, the formulation is extended to account for multiple production facilities and multiple customers.

$$\min PC = \sum_{w} (Cp_w + Ch_w + Cu_w) \tag{9a}$$

$$F^P(\mathbf{q}_{wp}) \leq 0, \qquad \forall k, w, p \tag{9b}$$

$$Cp_w = \sum_{p} el_{pw} \times E^p(\mathbf{q}_{wp}), \qquad \forall k, w \tag{9c}$$

$$Ch_w = \sum_{k} \sum_{p} h_{kp} \times IT_{kwp}, \qquad \forall w \tag{9d}$$

$$Cu_t = \sum_{k} \sum_{c} u_{kc} \times U_{kwc}, \qquad \forall w \tag{9e}$$

$$IT_{kwp} = IT_{k,w-1,p} + PT_{kwp} - \sum_{c \in CP} Del_{kwpc}, \qquad \forall w, p, k \in K^L \tag{9f}$$

$$PT_{kwp} = \sum_{c \in CP} Del_{kwpc}, \qquad \forall w, p, k \in K^G \tag{9g}$$

$$U_{kwc} = Dem_{kwc} - \sum_{p \in PC} Del_{kwpc}, \qquad \forall k, w, c \tag{9h}$$

The feasibility of the scheduling-control problem for each week w and for each production facility p is modeled via eq. (9b). The production cost Cp_w in period w is expressed via eq. (9c) as a function of the weekly electricity prices forecasted for each plant, el_{pw}, and the electricity consumption, $E^p(\cdot)$. Both equations (9b) and (9c) are a

function of a subset of scheduling-control variables, $\mathbf{q}_{wp}$. Holding cost Ch_w and unmet demand penalty Cu_w are calculated in equations (9d) and (9e), respectively. Holding costs are dependent on the product $k \in \{GN2, GO2, LN2\}$ and the production facility p, while unmet demand penalties are related to the product k and customer c. Inventory targets IT_{kwp} for each production facility are calculated according to equation (9f). In equation (9f), Del_{kwpc} refers to the amount of product k delivered from production facility p to customer c at planning period w. The subset CP relates the customers which are connected to production facilities, according to the enterprise network in Figure 5.3. Equation (9g) is the mass balance for gaseous products. Unmet demand is calculated in equation (9h) as the difference between the forecasted demand Dem_{kwc} and the total deliveries to the customer c from the various plants in the enterprise.

In this planning model, the black-box constraints related to the scheduling-control problem are given by equations (9b) and (9c). With this identification, steps 1 and 2 of the proposed framework are concluded, and we move to steps 3 and 4 in the next subsection.

5.4.2.2 Feature selection and surrogate models

The selection of features $\mathbf{q}$ which will be used to estimate the feasibility of the scheduling-control problem and its associated cost will be performed based on ad-hoc knowledge. Ad-hoc feature selection methodologies are recommended whenever domain knowledge is existent [110]. We therefore select $\mathbf{q}$ as the production targets PT_{kwp}, where PT_{kwp} is related to scheduling target T_{kt} of plant p as $T_{kt} = PT_{kwp}/\mathrm{H}$, and H is the number of discretization periods in the scheduling problem. Then, the feasibility constraints $F^P(PT_{kwp}) \leq 0$ are defined according to the methodologies described in Chapter 3: for a

training set of production targets $\mathbf{S^p} = \{S_1, S_2, \dots S_N\},\ S_i = PT_{kwp}, i \in \{1,2, \dots N\}$, the feasibility of each sample is verified by solving the integrated scheduling and control problem. The results are compiled as $\mathbf{F^p} = \{F_1, F_2, \dots, F_N\}$, where $F_i = -1$ if sample i is feasible, and $+1$ otherwise. Neural networks are trained and used to predict the feasibility of any point in the design space. A similar procedure is followed to estimate the energy consumption $E^P(PT_{kwp})$. First, a training set of feasible production targets is generated, $\mathbf{S^p} = \{S_1, S_2, \dots S_N\},\ S_i = PT_{kwp}, i \in \{1,2, \dots N\}, F\left(PT_{kwp}\right) \leq 0$. The minimum energy consumed to achieve each production targets is calculated by solving the integrated scheduling and control problem, and the results are compiled in a vector **Y**. Then, neural networks are trained and used to predict the energy consumption of any point in the design space.

After the training of neural networks, equations (9b) and (9c) take the form of Eq. (10a) and (10b), respectively:

$$\mathbf{w}_h^p \times \left(\frac{1}{1 + \exp\left(\mathbf{w}_{input}^p \times \begin{bmatrix} PT_{GN2,w,p} \\ PT_{GO2,w,p} \\ PT_{LN2,w,p} \end{bmatrix} + \mathbf{b}_{input}^p \right)} \right) + b_h \leq 0 \tag{10a}$$

$$Cp_w = \sum_p el_{pw} \times \boldsymbol{m}_h^p \times \left(\frac{1}{1 + \exp\left(\boldsymbol{m}_{input}^p \times \begin{bmatrix} PT_{GN2,w,p} \\ PT_{GO2,w,p} \\ PT_{LN2,w,p} \end{bmatrix} + \boldsymbol{c}_{input}^p \right)} \right) + c_h \tag{10b}$$

5.4.1.3 Integrated problem

The planning, scheduling and control problem can be summarized as:

Minimize Objective Function (9a)

Subject to: Feasibility constraints (Eq. 10a)

Operational costs (Eq. 10b) (P2)

Holding and unmet demand costs (Eq. 9d-9e)

Material balances and demand considerations (Eq. 9f-9h)

Due to the linear nature of the planning constraints, and due to the nonlinearities introduced by the black-box constraints (Eq. 10), the resulting problem is a nonlinear optimization problem, which is solved in GAMS/CONOPT to determine a local optimal solution.

5.5 Case study

The framework for the integration of planning, scheduling and control was implemented in the enterprise of air separation plants described in section 5.3. In this section, we give a brief overview of the dynamic control simulation, followed by the results of integrating scheduling and control. Finally, we discuss the results of the overall planning, scheduling and control optimization.

5.5.1 Air Separation Unit Model

A first principle model for the dynamic behavior of an air separation unit was been derived. The models for the cryogenic distillation column, integrated reboiler/condenser, multi-stream heat exchanger, compressor and turbines were obtained, following previously

published works [64, 66]. A summary of the dynamic equations employed in the model is given in Appendix C.

Following this derivation, system identification procedures were used to identify a state-space model, which would be used in a Model Predictive Control implementation. The output variables of the model are the flows of gas nitrogen leaving the distillation column, $GN2$; the flow of gas oxygen leaving the distillation column, $GO2$; the reboiler holdup, M_{reb}; the temperature driving force in the integrated reboiler/condenser, ΔT; the impurity level of the gas oxygen stream, Imp_{GO2}; and the impurity level in the gas nitrogen stream, Imp_{GN2}. The control (manipulated) variables were selected as the main air flow rate, MA; the expanded air flow rate, EA; the reflux rate from the condenser, R_{con}; the reflux rate from the reboiler, R_{reb}; and the liquid nitrogen stream, LN_2.

A model predictive control was implemented to drive changes of the setpoints of the system. The MPC was implemented with sample step of 360s, and a prediction horizon of 1 hour. The MPC problem attempts to track setpoints for the output variables, while also imposing rate of change penalties for the input variables. Additionally, hard input constraints are imposed to the main air and expanded air flows (MA, EA), and soft control constraints are imposed to the impurity levels in the product streams, $Imp_{GO2} \geq 0.946$ and $Imp_{GN2} \geq 0.997$.

The control level simulation then consists of: (1) measuring the current state of the system, (2) solving the MPC problem, (3) implementing the control actions defined by MPC in the first-principle dynamic model, (4) returning to step (1) after the dynamic system has been simulated for 360s. The procedure is repeated until the horizon of the simulation (which is defined based on the scheduling discretization step) has been reached.

5.5.2 Integration of scheduling and control

We first consider the problem of integrating scheduling and control for a single air separation plant, which is capable of flexible operation, in the sense that the production rate (defined in terms of the flow rate of gas nitrogen product) can assume any value in the range of 40 mol/s to 60 mol/s. The nominal production rate is 50mol/s. The scheduling problem takes into consideration a forecast of energy prices varying at every hour, as well as a constant demand of gas nitrogen, set at 50mol/s. The scheduling problem then defines production flows and setpoints for every point of the scheduling horizon while minimizing operational costs. Since flows and operational costs depend the dynamic behavior of the system, the integration of scheduling and control is sought.

We follow the proposed methodology to achieve the integration of scheduling and control. First, the higher-level optimization model is formulated, as described in section 5.4.1.1. Then, the feature selection and surrogate model training is performed, in order to identify a subset of variables $\mathbf{z} \subset \{\mathbf{x}_t^{sp}, \mathbf{u}_t^{sp}, \tilde{\mathbf{x}}_t^0, \tilde{\mathbf{u}}_t^0\}$ which will be used to predict the flows PD_{kt}, MA_t, EA_t. The set of candidate features include $\mathbf{x}_t^0 = \{GN2^0, GO2^0, M_{reb}^0, \Delta T^0, Imp_{GO2}^0, Imp_{GN2}^0\}$, $\tilde{\mathbf{u}}_t^0 = \{MA, EA, R_{con}, R_{reb}, LN2\}$, and $\mathbf{x}_t^{sp} = \{GN2^{sp}, GO2^{sp}\}$. The remaining control input setpoints are not considered since they are kept at a constant value throughout out simulations. Furthermore, the input variable setpoints are not relevant to this problem, since the MPC implemented here only weights the rate of change values for control variables (instead of tracking a setpoint). The feature selection and training steps were performed in Matlab using the Statistical and Machine Learning toolbox. After the feature selection procedure was completed, the following subset of variables were obtained:

$$\overline{PD}_{GN2,t} = f(GN2^0, GN2^{sp})$$

$$\overline{PD}_{GO2,t} = f(GN2^0, GO2^0, GN2^{sp})$$

$$\overline{PD}_{LN2,t} = f(MA^0, LN2^0, GN2^{sp})$$

$$\overline{MA}_t = f(GN2^0, GO2^0, GN2^{sp})$$

$$\overline{EA} = f(EA^0, LN2^0, GN2^{sp})$$

$$GN2^+ = f(GN2^{sp})$$

$$GO2^+ = f(LN2^0, GN2^{sp})$$

$$LN2^+ = f(EA^0, LN2^0, GN2^{sp})$$

$$MA^+ = f(LN2^0, GN2^{sp})$$

$$EA^+ = f(LN2^0, GN2^{sp})$$

In the above equations, the superscript ^{0} is used to denote the state/input values of the system at the beginning of the simulation; the superscript $^+$ is used to denote the state/input values of the system at the end of the simulation; $\bar{x}$ denotes an average flow rate for the entire simulation. It is interesting to see how the selection of features provides insights of the underlying behavior of the system, such as a strong and mutual relationship between the values for expanded air EA and liquid nitrogen flows $LN2$.

The above equations are rewritten using big-M formulations to take the form of Eq. (8), and the integrated problem P1 is solved to global optimality. The results of this problem are shown in Table 5.1. The performance of the integrated scheduling solution is compared to the baseline case of keeping the production levels constant throughout the entire schedule, i.e., $PD_{GN2,t} = 50mol/s$, in order to satisfy the constant demand of 50mol/s of nitrogen gas. Finally, the scheduling solutions are implemented in the control problem simulation, which is now solved for the entire scheduling horizon. The simulation provides

the actual costs of operating the system, as well as the state and control variables behavior throughout the entire schedule. The results are shown in Figure 5.6 and Figure 5.7. The problems are implemented in a 64-bit Windows system with Intel Core i7-2600 CPU at 3.40 GHz and 16 GB RAM.

Table 5.1 – Optimal schedule results

Case	**Predicted Overall Cost**	**Actual Overall Cost**	**Difference from Baseline**	**Solution Time**
Baseline*	$ 3124.79	$ 3124.79	0%	-
Case 1 (MPC)	$ 3012.33	$ 3019.29	3.38%	67 s

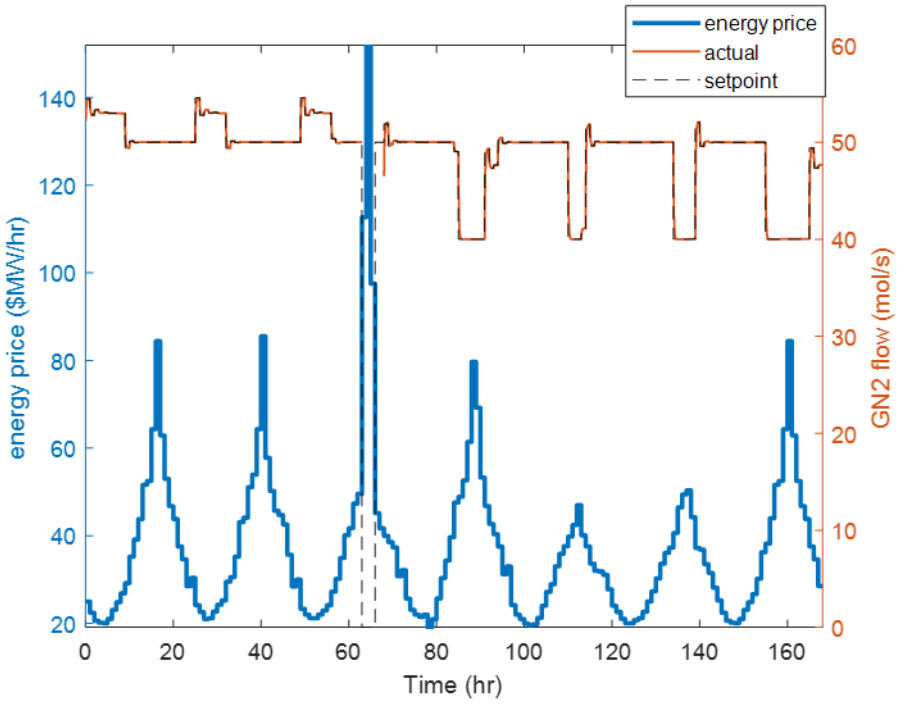

Figure 5.6 - Optimal production rates and electricity prices over time

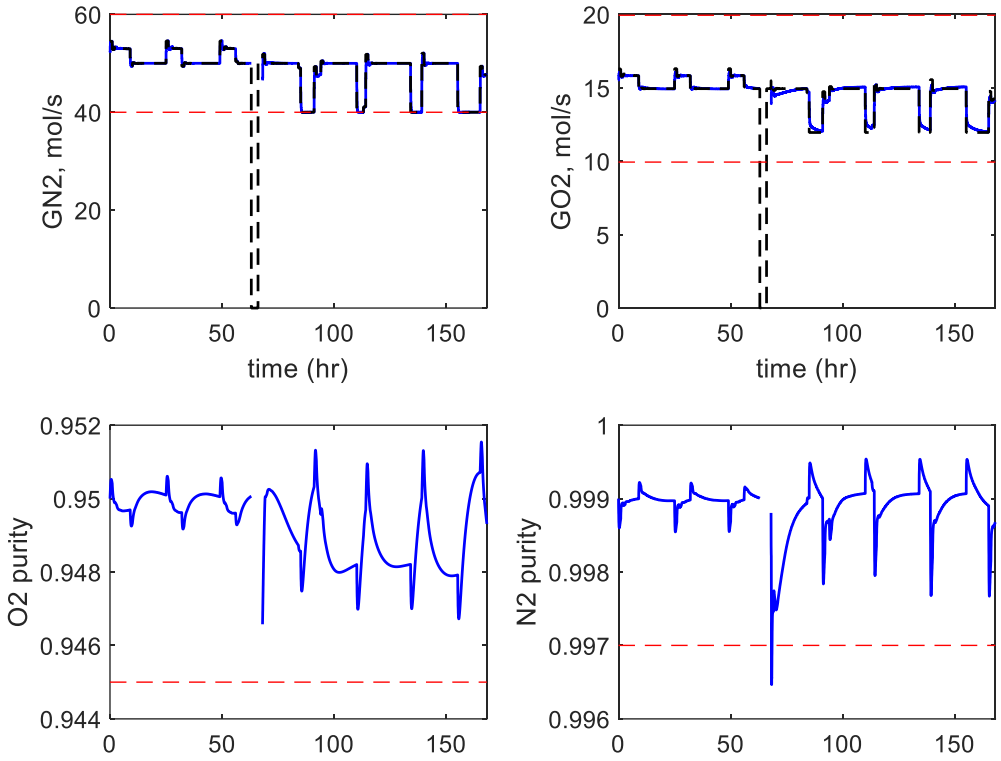

Figure 5.7 - Outputs of the control problem over time

The integrated scheduling and control problem results outperforms the baseline case, achieving a 3.38% reduction of operational costs. Furthermore, the predicted operational costs show a deviation of only 0.23% from the actual operational costs, which demonstrates the accuracy of the surrogate models and validates the proposed methodology for scheduling and control integration. By analyzing Figure 5.6, we observe that the behavior of the scheduling-control problem follows intuitive predictions: it sets the production rates to higher levels whenever the energy prices are low. The scheduling problem also makes the choice of shutting down the operation during a peak of energy price. Finally, the impurity levels of nitrogen and oxygen flow are maintained above the

quality specifications throughout the entire operation, and the ASU produces off-spec material only during the starting up of the plant.

In the next section, three different air separation plants will be considered in an enterprise. Therefore, the integrated scheduling and control model for each plant should be obtained. To obtain three different models, we first replicate the dynamic model for each ASU. By modifying constraints on production and air flows, as well as the efficiency parameters for different operation units, two new production facilities are modeled. Plant 1 is defined as the original problem. Plant 2 produces nitrogen in a smaller range, varying from 40mol/s to 50mol/s. Furthermore, the energy consumption for the operation of Plant 2 can be compared to the energy consumption in Plant 1 in a 1.1 to 1 ratio. Plant 3 also consumes higher levels of energy for the operation of the same production flows when compared to Plant 1, in a ratio of 1.05 to 1.

5.5.3 Integration of planning, scheduling and control

After the model for the scheduling and control integration has been obtained, it can be used as a black-box in an enterprise-wide optimization. In this section, we consider the problem of defining weekly production targets for an enterprise consisting of three air separation plants. The enterprise network is demonstrated in Figure 5.3. We assume that demand forecasts related to each customer is available. Furthermore, we assume that the different production facilities are subject to different electricity contracts, and a forecast for the average weekly electricity prices is available. A planning horizon of 52 weeks is considered. To define production targets for each facility, a planning problem with scheduling considerations should be solved. Since the scheduling of air separation units is

highly dependent on the dynamic behavior of the system, the integration of planning, scheduling and control is sought.

Following the proposed framework, the higher-level optimization problem is formulated, taking the form of equation 9. The features for the integrated problem are selected (based on ad-hoc knowledge) as the production targets PT_{kwp}, for each product $k = \{GN2, GO2, LN2\}$, each week of the planning horizon w, and each production facility p. Then, a training set is generated by solving the integrated scheduling and control problem for different production targets PT_{kwp}. The results are used to derive feasibility constraints and energy consumption predictors using neural networks, as described in section 5.4.2. The neural networks are incorporated in the planning problem, and the resulting integrated planning, scheduling and control problem is solved to local optimality.

The resulting problem is a nonlinear problem which is solved using GAMS/CONOPT in a 64-bit Windows system with Intel Core i7-2600 CPU at 3.40 GHz and 16 GB RAM. Results are obtained in 6.377 s. The overall cost of operating the enterprise for the entire planning horizon is defined as $ 319,263.69. The resulting target levels for each product and each production facility are shown in Figure 5.8. Furthermore, the predicted energy prices and predicted energy consumption for each facility are shown in Figure 5.9.

The production targets for Plant 1 show a higher degree of fluctuation, demonstrating the higher flexibility of operating this plant. Plant 2 operates near its lower bound for most of the planning horizon, a result of the higher energy consumption levels at this plant. A higher level of production targets for Plant 2 can be observed when the

energy prices of Plant 1 are high, around week 32. This demonstrates the capability of the integrated problem is defining cooperative decisions across time and space scales.

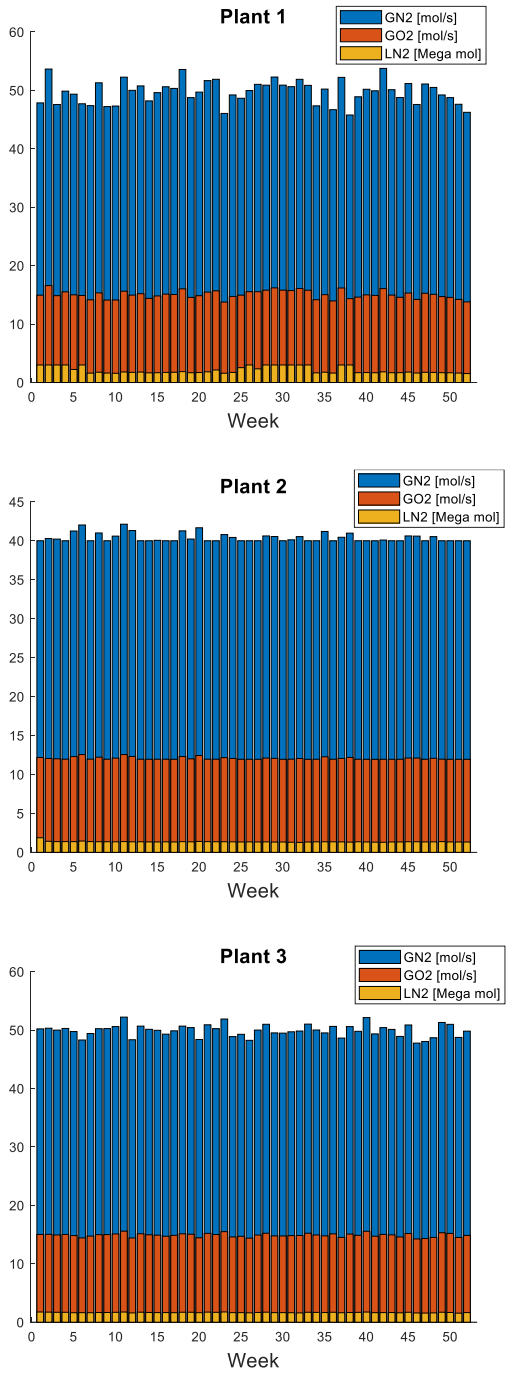

Figure 5.8 - Production targets for each Air Separation Plant

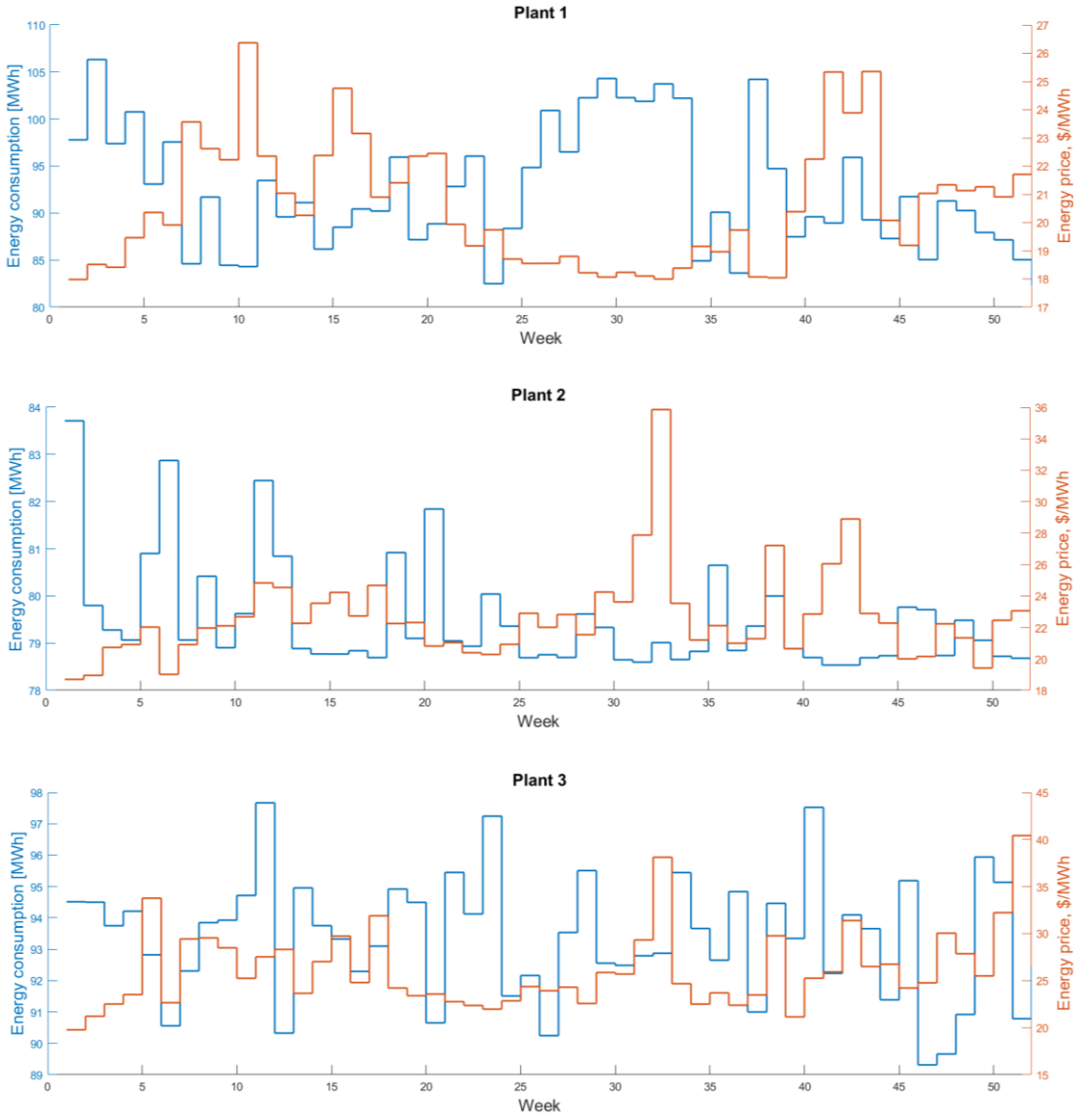

Figure 5.9 – Predicted energy consumption and average electricity prices

Next, the results for the integrated problem for Plant 1 are implemented at the scheduling level, for the entire planning horizon. In other words, 52 scheduling and control problems are solved, given the weekly production targets defined by the overall integrated problem, and given electricity prices varying on an hourly basis. Figure 5.10 shows the results of this implementation. The predicted energy consumption and operational costs are

very similar to the actual values obtained in the scheduling and control simulation. The black-box constraints approximation were successful, and the integrated planning, scheduling and control integration is validated.

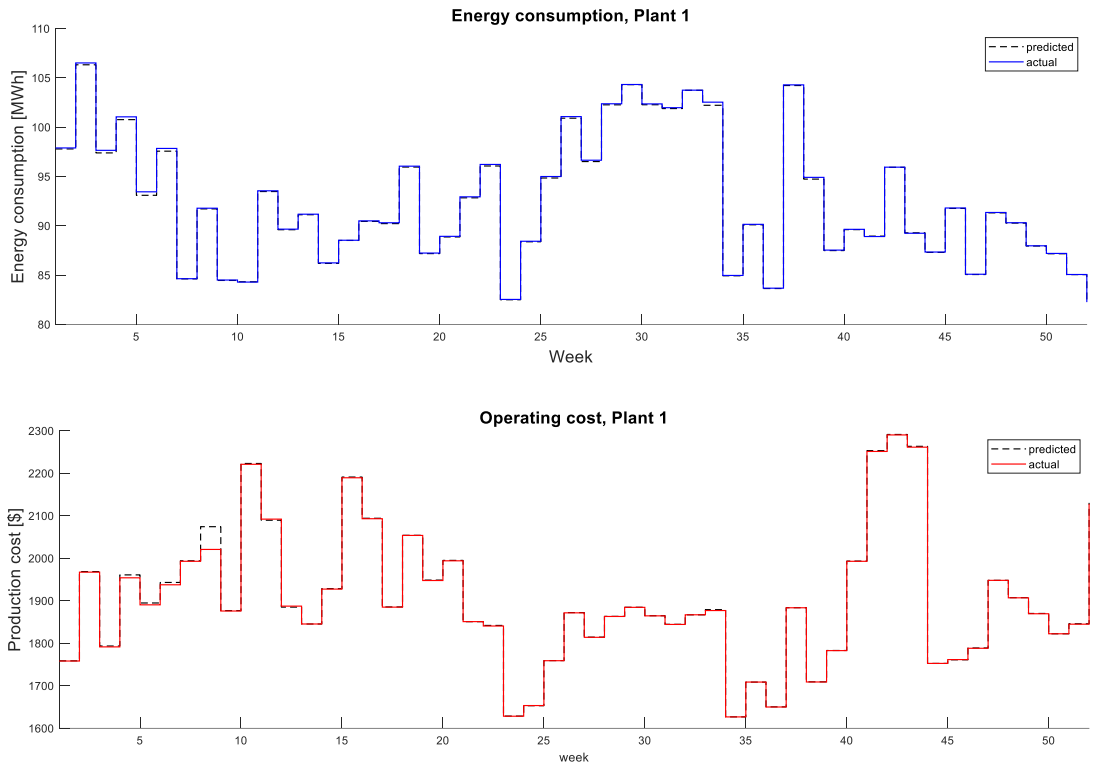

Figure 5.10 – Planning predictions and actual realizations of operational costs

5.6 Conclusions

In this work, a systematic methodology for the integration of planning, scheduling and control problems was proposed. The framework consists of two-building blocks, which are addressed following principles of grey-box optimization, feature selection, and surrogate modeling. The proposed framework was implemented in the problem of optimization an enterprise of air separation plants, taking into consideration varying electricity prices and

flexible operation of production facilities. Results of the integrated problem were compared to baseline solutions, and validated through comparisons of predictions versus actual operational costs.

One of the main challenges that has not been addressed in this work is related to uncertainties affecting all the levels of the decision-making processes. Throughout this entire chapter, we assumed the existence of accurate demand and energy forecasts, as well as deterministic operation of the dynamic system. Future work should focus in addressing the challenges of uncertainties that arise in all of the different levels of the decision-making hierarchy, as well as uncertainties introduced by inaccuracies in the surrogate models.

Chapter 6

Summary and Future Work

The integration of planning, scheduling and control problems is a challenging task, given the high dimensionality of industrial sized problems and the complexities of enterprises. Complex frameworks will not find its way to real applications, and major organizational challenges need to be addressed before the integration of decision-making processes can become a reality. Nevertheless, this work demonstrates the economic benefits that integrated processes can provide. The frameworks presented here may seem complex, as they carry in their core concepts of simulation-optimization, feasibility analysis, grey-box optimization, machine learning, model predictive control, and robust optimization. However, the methodologies proposed here can be summarized as a simple combination of data-driven methodologies and classical optimization techniques, which are gradually and reliably being adopted in industrial applications.

Throughout the chapters, several possibilities and future directions for enterprise-wide optimization have been discussed. We summarize here the main opportunities and venues for future research:

- Data-driven feasibility analysis can be implemented in a variety of problems within the process systems engineering field. We foresee interesting applications in the fields of modular design and design optimization
- Feature selection methodologies were not implemented in the data-driven feasibility analysis for the integrating planning and scheduling. Nevertheless, the case studies in chapter 3 demonstrate that the initial state of the system has an important impact in the feasibility of the schedule. The significance of this impact

will depend on the scheduling horizon, process network, and capacity utilization. Future work should investigate these relationships, and how feature selection can be employed in the selection of a subset of scheduling states for the optimal prediction of the feasible production region.

- The choice of kernels and activation functions for neural networks and support vector machines usually fall within the context of machine learning. Nevertheless, the use of the predictors in optimization problems defines a new criterion for model selection, since the complexity of the predictor affects the complexity of the final planning problem. Future work should investigate the structure of different predictors, how they cope with different solvers, and how the increase the complexity of optimization problems
- Black-box problems involving discrete decisions are likely to provide discontinuous outputs. To better implement surrogate models in such black-boxes, surrogate model techniques need to be coupled with methodologies for the identification of discontinuities

Acknowledgment of previous publications

Several sections of this dissertation have been published elsewhere or are being prepared for publication. Following are acknowledged.

- Chapter 1 includes extracts from the following published works:

Dias, L.S. and M.G. Ierapetritou, From process control to supply chain management: An overview of integrated decision making strategies. Computers & Chemical Engineering, 2017. 106(Supplement C): p. 826-835.

Dias, L.S. and M.G. Ierapetritou, Integration of scheduling and control under uncertainties: Review and challenges. Chemical Engineering Research and Design, 2016. 116: p. 98-113.

- Chapter 2 has been published in full under the citation:

Dias, L.S., et al., A simulation-based optimization framework for integrating scheduling and model predictive control, and its application to air separation units. Computers & Chemical Engineering, 2018. 113: p. 139-151

- Chapter 3 has been published in full under the citation:

Dias, L.S. and M. Ierapetritou. Data-driven feasibility analysis for the integration of planning and scheduling problems. Optimization and Engineering.

https://doi.org/10.1007/s11081-019-09459-w

- Chapter 4 has been submitted under the citation:

Dias, L.S. and M. Ierapetritou. Integration of scheduling and robust model predictive control. Industrial Engineering Chemistry Research.

- Chapter 5 has been submitted under the citation:

Dias, L.S. and M. Ierapetritou. Integration of planning, scheduling and control problems using feasibility analysis and surrogate models. Computers & Chemical Engineering..

Bibliography

1. *The Structural Cost of Manufacturing in the United States*. 2011 [cited 2016 10/2016].
2. Grossmann, I.E., *Enterprise-wide optimization: A new frontier in process systems engineering.* AIChE Journal, 2005. **51**(7): p. 1846-1857.
3. Dias, L.S. and M.G. Ierapetritou, *From process control to supply chain management: An overview of integrated decision making strategies.* Computers & Chemical Engineering, 2017. **106**(Supplement C): p. 826-835.
4. Dias, L.S. and M.G. Ierapetritou, *Integration of scheduling and control under uncertainties: Review and challenges.* Chemical Engineering Research and Design, 2016. **116**: p. 98-113.
5. Harjunkoski, I., et al., *Scope for industrial applications of production scheduling models and solution methods.* Computers & Chemical Engineering, 2014. **62**(Supplement C): p. 161-193.
6. Floudas, C.A. and X. Lin, *Continuous-time versus discrete-time approaches for scheduling of chemical processes: a review.* Computers & Chemical Engineering, 2004. **28**(11): p. 2109-2129.
7. Maravelias, C.T., *General framework and modeling approach classification for chemical production scheduling.* AIChE Journal, 2012. **58**(6): p. 1812-1828.
8. Qin, S.J. and T.A. Badgwell, *A survey of industrial model predictive control technology.* Control Engineering Practice, 2003. **11**(7): p. 733-764.
9. Mayne, D.Q., et al., *Constrained model predictive control: Stability and optimality.* Automatica, 2000. **36**(6): p. 789-814.
10. Engell, S. and I. Harjunkoski, *Optimal operation: Scheduling, advanced control and their integration.* Computers & Chemical Engineering, 2012. **47**(Supplement C): p. 121-133.
11. Engell, S. and I. Harjunkoski, *Optimal operation: Scheduling, advanced control and their integration.* Comput Chem Eng, 2012. **47**: p. 13.
12. Maravelias, C.T. and C. Sung, *Integration of production planning and scheduling: Overview, challenges and opportunities.* Computers & Chemical Engineering, 2009. **33**(12): p. 1919-1930.
13. Harjunkoski, I. and I.E. Grossmann, *Decomposition techniques for multistage scheduling problems using mixed-integer and constraint programming methods.* Computers & Chemical Engineering, 2002. **26**(11): p. 1533-1552.
14. Jain, V. and I.E. Grossmann, *Algorithms for Hybrid MILP/CP Models for a Class of Optimization Problems.* INFORMS Journal on Computing, 2001. **13**(4): p. 258-276.
15. Maravelias, C.T., *A decomposition framework for the scheduling of single- and multi-stage processes.* Computers & Chemical Engineering, 2006. **30**(3): p. 407-420.
16. Roe, B., L.G. Papageorgiou, and N. Shah, *A hybrid MILP/CLP algorithm for multipurpose batch process scheduling.* Computers & Chemical Engineering, 2005. **29**(6): p. 1277-1291.
17. Birewar, D.B. and I.E. Grossmann, *Simultaneous production planning and scheduling in multiproduct batch plants.* Industrial & Engineering Chemistry Research, 1990. **29**(4): p. 570-580.
18. Dogan, M.E. and I.E. Grossmann, *A Decomposition Method for the Simultaneous Planning and Scheduling of Single-Stage Continuous Multiproduct Plants.* Industrial & Engineering Chemistry Research, 2006. **45**(1): p. 299-315.

19. Wellons, M.C. and G.V. Reklaitis, *Scheduling of multipurpose batch chemical plants. 2. Multiple-product campaign formation and production planning.* Industrial & Engineering Chemistry Research, 1991. **30**(4): p. 688-705.
20. Wilkinson, S.J., N. Shah, and C.C. Pantelides, *Aggregate modelling of multipurpose plant operation.* Computers & Chemical Engineering, 1995. **19**(Supplement 1): p. 583-588.
21. Li, Z. and M.G. Ierapetritou, *Rolling horizon based planning and scheduling integration with production capacity consideration.* Chemical Engineering Science, 2010. **65**(22): p. 5887-5900.
22. Li, Z. and M.G. Ierapetritou, *Integrated production planning and scheduling using a decomposition framework.* Chemical Engineering Science, 2009. **64**(16): p. 3585-3597.
23. Sung, C. and C.T. Maravelias, *An attainable region approach for production planning of multiproduct processes.* AIChE Journal, 2007. **53**(5): p. 1298-1315.
24. Sung, C. and C.T. Maravelias, *A projection-based method for production planning of multiproduct facilities.* AIChE Journal, 2009. **55**(10): p. 2614-2630.
25. Wu, D. and M. Ierapetritou, *Hierarchical approach for production planning and scheduling under uncertainty.* Chemical Engineering and Processing: Process Intensification, 2007. **46**(11): p. 1129-1140.
26. Papageorgiou, L.G. and C.C. Pantelides, *Optimal Campaign Planning/Scheduling of Multipurpose Batch/Semicontinuous Plants. 2. A Mathematical Decomposition Approach.* Industrial & Engineering Chemistry Research, 1996. **35**(2): p. 510-529.
27. Li, Z. and M.G. Ierapetritou, *Production planning and scheduling integration through augmented Lagrangian optimization.* Computers & Chemical Engineering, 2010. **34**(6): p. 996-1006.
28. Baldea, M. and I. Harjunkoski, *Integrated production scheduling and process control: A systematic review.* Computers & Chemical Engineering, 2014. **71**(Supplement C): p. 377-390.
29. Flores-Tlacuahuac, A. and I.E. Grossmann, *Simultaneous Cyclic Scheduling and Control of a Multiproduct CSTR.* Industrial & Engineering Chemistry Research, 2006. **45**(20): p. 6698-6712.
30. Terrazas-Moreno, S., A. Flores-Tlacuahuac, and I.E. Grossmann, *Simultaneous cyclic scheduling and optimal control of polymerization reactors.* AIChE Journal, 2007. **53**(9): p. 2301-2315.
31. Zhuge, J. and M.G. Ierapetritou, *Integration of Scheduling and Control with Closed Loop Implementation.* Industrial & Engineering Chemistry Research, 2012. **51**(25): p. 8550-8565.
32. Nyström, R.H., et al., *Production campaign planning including grade transition sequencing and dynamic optimization.* Computers & Chemical Engineering, 2005. **29**(10): p. 2163-2179.
33. Nyström, R.H., I. Harjunkoski, and A. Kroll, *Production optimization for continuously operated processes with optimal operation and scheduling of multiple units.* Computers & Chemical Engineering, 2006. **30**(3): p. 392-406.
34. Chu, Y. and F. You, *Integration of production scheduling and dynamic optimization for multi-product CSTRs: Generalized Benders decomposition coupled with global mixed-integer fractional programming.* Computers & Chemical Engineering, 2013. **58**(Supplement C): p. 315-333.
35. Chu, Y. and F. You, *Integrated Scheduling and Dynamic Optimization of Complex Batch Processes with General Network Structure Using a Generalized Benders Decomposition Approach.* Industrial & Engineering Chemistry Research, 2013. **52**(23): p. 7867-7885.

36. Zhuge, J. and M.G. Ierapetritou, *Integration of scheduling and control for batch processes using multi-parametric model predictive control.* AIChE Journal, 2014. **60**(9): p. 3169-3183.
37. Zhuge, J. and M.G. Ierapetritou, *An integrated framework for scheduling and control using fast model predictive control.* AIChE Journal, 2015. **61**(10): p. 3304-3319.
38. Simkoff, J.M. and M. Baldea, *Production scheduling and linear MPC: Complete integration via complementarity conditions.* Computers & Chemical Engineering, 2019. **125**: p. 287-305.
39. Du, J., et al., *A time scale-bridging approach for integrating production scheduling and process control.* Computers & Chemical Engineering, 2015. **79**(Supplement C): p. 59-69.
40. Baldea, M., et al., *Integrated production scheduling and model predictive control of continuous processes.* AIChE Journal, 2015. **61**(12): p. 4179-4190.
41. Pattison, R.C., et al., *Optimal Process Operations in Fast-Changing Electricity Markets: Framework for Scheduling with Low-Order Dynamic Models and an Air Separation Application.* Industrial & Engineering Chemistry Research, 2016. **55**(16): p. 4562-4584.
42. Pattison, R.C., et al., *Moving horizon closed-loop production scheduling using dynamic process models.* AIChE Journal, 2017. **63**(2): p. 639-651.
43. Gutiérrez-Limón, M.A., A. Flores-Tlacuahuac, and I.E. Grossmann, *MINLP Formulation for Simultaneous Planning, Scheduling, and Control of Short-Period Single-Unit Processing Systems.* Industrial & Engineering Chemistry Research, 2014. **53**(38): p. 14679-14694.
44. Shi, H., Y. Chu, and F. You, *Novel Optimization Model and Efficient Solution Method for Integrating Dynamic Optimization with Process Operations of Continuous Manufacturing Processes.* Industrial & Engineering Chemistry Research, 2015. **54**(7): p. 2167-2187.
45. Charitopoulos, V.M., L.G. Papageorgiou, and V. Dua, *Closed-loop integration of planning, scheduling and multi-parametric nonlinear control.* Computers & Chemical Engineering, 2018.
46. Ławryńczuk, M., *Accuracy and computational efficiency of suboptimal nonlinear predictive control based on neural models.* Applied Soft Computing, 2011. **11**(2): p. 2202-2215.
47. Lazar, M. and O. Pastravanu, *A neural predictive controller for non-linear systems.* Mathematics and Computers in Simulation, 2002. **60**(3): p. 315-324.
48. Chen, S., et al. *Approximating Explicit Model Predictive Control Using Constrained Neural Networks*. in *2018 Annual American Control Conference (ACC)*. 2018.
49. Parisini, T. and R. Zoppoli, *A receding-horizon regulator for nonlinear systems and a neural approximation.* Automatica, 1995. **31**(10): p. 1443-1451.
50. Ning, C. and F. You, *Data-driven decision making under uncertainty integrating robust optimization with principal component analysis and kernel smoothing methods.* Computers & Chemical Engineering, 2018. **112**: p. 190-210.
51. Fu, M.C., *Feature Article: Optimization for simulation: Theory vs. Practice.* INFORMS Journal on Computing, 2002. **14**(3): p. 192-215.
52. Fu, M.C., F.W. Glover, and J. April. *Simulation optimization: a review, new developments, and applications*. in *Proceedings of the Winter Simulation Conference, 2005.* 2005.
53. Grossmann, I.E., B.A. Calfa, and P. Garcia-Herreros, *Evolution of concepts and models for quantifying resiliency and flexibility of chemical processes.* Computers & Chemical Engineering, 2014. **70**(Supplement C): p. 22-34.
54. Bhosekar, A. and M. Ierapetritou, *Advances in surrogate based modeling, feasibility analysis, and optimization: A review.* Computers & Chemical Engineering, 2018. **108**: p. 250-267.

55. Sahay, N. and M. Ierapetritou, *Supply chain management using an optimization driven simulation approach.* AIChE Journal, 2013. **59**(12): p. 4612-4626.
56. Wang, L. and D.-Z. Zheng, *An effective hybrid optimization strategy for job-shop scheduling problems.* Computers & Operations Research, 2001. **28**(6): p. 585-596.
57. Xia, W. and Z. Wu, *An effective hybrid optimization approach for multi-objective flexible job-shop scheduling problems.* Computers & Industrial Engineering, 2005. **48**(2): p. 409-425.
58. Ierapetritou, M.G., et al., *Cost Minimization in an Energy-Intensive Plant Using Mathematical Programming Approaches.* Industrial & Engineering Chemistry Research, 2002. **41**(21): p. 5262-5277.
59. Zhang, Q., et al., *Air separation with cryogenic energy storage: Optimal scheduling considering electric energy and reserve markets.* AIChE Journal, 2015. **61**(5): p. 1547-1558.
60. Karwan, M.H. and M.F. Keblis, *Operations planning with real time pricing of a primary input.* Computers & Operations Research, 2007. **34**(3): p. 848-867.
61. Miller, J., W.L. Luyben, and S. Blouin, *Economic Incentive for Intermittent Operation of Air Separation Plants with Variable Power Costs.* Industrial & Engineering Chemistry Research, 2008. **47**(4): p. 1132-1139.
62. Zhu, Y., S. Legg, and C.D. Laird, *Optimal design of cryogenic air separation columns under uncertainty.* Computers & Chemical Engineering, 2010. **34**(9): p. 1377-1384.
63. Cao, Y., et al., *Optimization-based assessment of design limitations to air separation plant agility in demand response scenarios.* Journal of Process Control, 2015. **33**(Supplement C): p. 37-48.
64. Johansson, T., *Integrated scheduling and control of an air separation unit subject to time-varying electricity prices*. 2015, KTH/ University of Texas at Austin.
65. Cao, Y., *Design for dynamic performance: application to an air separation unit.*, in *Chemical Engineering & Management*. 2011, McMaster University, Hamilton, Ontario, Canada.
66. Huang, R., V.M. Zavala, and L.T. Biegler, *Advanced step nonlinear model predictive control for air separation units.* Journal of Process Control, 2009. **19**(4): p. 678-685.
67. Green, D. and R. Perry, *Perry's Chemical Engineers' Handbook. 8th ed. Chemical Engineers Handbook.* 2007, McGraw-Hill.
68. Zhang, Q., et al., *A discrete-time scheduling model for continuous power-intensive process networks with various power contracts.* Computers & Chemical Engineering, 2016. **84**(Supplement C): p. 382-393.
69. Zhuge, J. and M.G. Ierapetritou, *Integration of scheduling and control for batch processes using multi-parametric model predictive control.* AIChE Journal, 2014. **60**(9): p. 3169-3183.
70. Kleijnen, J.P.C., *Response surface methodology for constrained simulation optimization: An overview.* Simulation Modelling Practice and Theory, 2008. **16**(1): p. 50-64.
71. Kleijnen, J.P.C., *Kriging metamodeling in simulation: A review.* European Journal of Operational Research, 2009. **192**(3): p. 707-716.
72. Chau, M., et al. *Simulation optimization: A tutorial overview and recent developments in gradient-based methods*. in *Proceedings of the Winter Simulation Conference 2014*. 2014.
73. Ferris, M.C., T.S. Munson, and K. Sinapiromsaran. *A practical approach to sample-path simulation optimization*. in *2000 Winter Simulation Conference Proceedings (Cat. No.00CH37165)*. 2000.
74. Gurkan, G., A.Y. Ozge, and T.M. Robinson. *Sample-path optimization in simulation*. in *Proceedings of Winter Simulation Conference*. 1994.

75. Subramanian, K., C.T. Maravelias, and J.B. Rawlings, *A state-space model for chemical production scheduling.* Computers & Chemical Engineering, 2012. **47**(Supplement C): p. 97-110.
76. Gupta, D. and C.T. Maravelias, *On deterministic online scheduling: Major considerations, paradoxes and remedies.* Computers & Chemical Engineering, 2016. **94**(Supplement C): p. 312-330.
77. Ellis, M., H. Durand, and P.D. Christofides, *A tutorial review of economic model predictive control methods.* Journal of Process Control, 2014. **24**(8): p. 1156-1178.
78. Rawlings, J.B. and M.J. Risbeck, *Model predictive control with discrete actuators: Theory and application.* Automatica, 2017. **78**(Supplement C): p. 258-265.
79. Castro, P.M., I.E. Grossmann, and Q. Zhang, *Expanding scope and computational challenges in process scheduling.* Computers & Chemical Engineering, 2018. **114**: p. 14-42.
80. Kondili, E., C.C. Pantelides, and R.W.H. Sargent, *A general algorithm for short-term scheduling of batch operations—I. MILP formulation.* Computers & Chemical Engineering, 1993. **17**(2): p. 211-227.
81. Grossmann, I.E., K.P. Halemane, and R.E. Swaney, *Optimization strategies for flexible chemical processes.* Computers & Chemical Engineering, 1983. **7**(4): p. 439-462.
82. Grossmann, I.E. and D.A. Straub. *Recent Developments in the Evaluation and Optimization of Flexible Chemical Processes*. 1996. Berlin, Heidelberg: Springer Berlin Heidelberg.
83. Halemane, K.P. and I.E. Grossmann, *Optimal process design under uncertainty.* 1983. **29**(3): p. 425-433.
84. Grossmann, I.E. and C.A. Floudas, *Active constraint strategy for flexibility analysis in chemical processes.* Computers & Chemical Engineering, 1987. **11**(6): p. 675-693.
85. Boukouvala, F. and M.G. Ierapetritou, *Feasibility analysis of black-box processes using an adaptive sampling Kriging-based method.* Computers & Chemical Engineering, 2012. **36**(Supplement C): p. 358-368.
86. Wang, Z. and M. Ierapetritou, *A novel feasibility analysis method for black-box processes using a radial basis function adaptive sampling approach.* AIChE Journal, 2017. **63**(2): p. 532-550.
87. Kotsiantis, S.B., I. Zaharakis, and P.J.E.a.i.a.i.c.e. Pintelas, *Supervised machine learning: A review of classification techniques.* 2007. **160**: p. 3-24.
88. Li, Z. and M.G. Ierapetritou, *Reactive scheduling using parametric programming.* 2008. **54**(10): p. 2610-2623.
89. Llanas, B., S. Lantarón, and F.J. Sáinz, *Constructive Approximation of Discontinuous Functions by Neural Networks.* Neural Processing Letters, 2008. **27**(3): p. 209-226.
90. Boursier Niutta, C., et al., *Surrogate modeling in design optimization of structures with discontinuous responses.* Structural and Multidisciplinary Optimization, 2018. **57**(5): p. 1857-1869.
91. Gorodetsky, A. and Y. Marzouk, *Efficient localization of discontinuities in complex computational simulations.* 2014. **36**(6): p. A2584-A2610.
92. Stephenson, J., K. Gallagher, and C.C. Holmes, *Beyond kriging: dealing with discontinuous spatial data fields using adaptive prior information and Bayesian partition modelling.* Geological Society, London, Special Publications, 2004. **239**(1): p. 195-209.
93. Pedregosa, F., et al., *Scikit-learn: Machine Learning in Python.* Journal of Machine Learning Research, 2011. **12**: p. 2825-2830.
94. Breiman, L., *Classification and regression trees*. 2017: Routledge.

95. Grossmann, I.E. and F. Trespalacios, *Systematic modeling of discrete-continuous optimization models through generalized disjunctive programming.* 2013. **59**(9): p. 3276-3295.
96. Cortes, C. and V. Vapnik, *Support-vector networks.* Machine Learning, 1995. **20**(3): p. 273-297.
97. Mitra, K., et al., *Resiliency Issues in Integration of Scheduling and Control.* Industrial & Engineering Chemistry Research, 2010. **49**(1): p. 222-235.
98. Patil, B.P., E. Maia, and L.A. Ricardez-Sandoval, *Integration of scheduling, design, and control of multiproduct chemical processes under uncertainty.* 2015. **61**(8): p. 2456-2470.
99. Kiureghian, A.D. and O. Ditlevsen, *Aleatory or epistemic? Does it matter?* Structural Safety, 2009. **31**(2): p. 105-112.
100. Pistikopoulos, E.N., *Uncertainty in process design and operations.* Computers & Chemical Engineering, 1995. **19**: p. 553-563.
101. Li, Z. and M. Ierapetritou, *Process scheduling under uncertainty: Review and challenges.* Computers & Chemical Engineering, 2008. **32**(4): p. 715-727.
102. Campo, P.J. and M. Morari. *Robust Model Predictive Control*. in *1987 American Control Conference*. 1987.
103. Allwright, J.C. and G.C. Papavasiliou, *On linear programming and robust model-predictive control using impulse-responses.* Systems & Control Letters, 1992. **18**(2): p. 159-164.
104. Zheng, Z.Q. and M. Morari. *Robust Stability of Constrained Model Predictive Control*. in *1993 American Control Conference*. 1993.
105. Kothare, M.V., V. Balakrishnan, and M. Morari, *Robust constrained model predictive control using linear matrix inequalities.* Automatica, 1996. **32**(10): p. 1361-1379.
106. Lee, J.H. and Z. Yu, *Worst-case formulations of model predictive control for systems with bounded parameters.* Automatica, 1997. **33**(5): p. 763-781.
107. Rawlings, J.B. and D.Q. Mayne, *Model predictive control: Theory and design*. 2009: Nob Hill Pub. Madison, Wisconsin.
108. Mayne, D.Q., M.M. Seron, and S.V. Raković, *Robust model predictive control of constrained linear systems with bounded disturbances.* Automatica, 2005. **41**(2): p. 219-224.
109. Mayne, D.Q., et al., *Tube-based robust nonlinear model predictive control.* 2011. **21**(11): p. 1341-1353.
110. Guyon, I., et al., *An introduction to variable and feature selection.* J. Mach. Learn. Res., 2003. **3**: p. 1157-1182.
111. Risbeck, M.J. and J.B. Rawlings, *Nonlinear model predictive control tools for CasADi (Octave interface).* 2016: https://bitbucket.org/rawlings-group/octave-mpctools.
112. Andersson, J.A., et al., *CasADi: a software framework for nonlinear optimization and optimal control.* 2019. **11**(1): p. 1-36.
113. Petersen, L.N., et al. *Economic optimization of spray dryer operation using Nonlinear Model Predictive Control*. in *53rd IEEE Conference on Decision and Control*. 2014.
114. Boukouvala, F., M.M. Hasan, and C.A. Floudas, *Global optimization of general constrained grey-box models: new method and its application to constrained PDEs for pressure swing adsorption %J J. of Global Optimization.* 2017. **67**(1-2): p. 3-42.
115. Boukouvala, F. and C.A.J.O.L. Floudas, *ARGONAUT: AlgoRithms for Global Optimization of coNstrAined grey-box compUTational problems.* 2017. **11**(5): p. 895-913.
116. Le Digabel, S., *Algorithm 909: NOMAD: Nonlinear Optimization with the MADS Algorithm %J ACM Trans. Math. Softw.* 2011. **37**(4): p. 1-15.

117. Runarsson, T.P. and Y. Xin, *Search biases in constrained evolutionary optimization.* IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 2005. **35**(2): p. 233-243.
118. Powell, M.J.D., *A Direct Search Optimization Method That Models the Objective and Constraint Functions by Linear Interpolation*, in *Advances in Optimization and Numerical Analysis*, S. Gomez and J.-P. Hennart, Editors. 1994, Springer Netherlands: Dordrecht. p. 51-67.
119. Wilson, Z.T. and N.V. Sahinidis, *The ALAMO approach to machine learning.* Computers & Chemical Engineering, 2017. **106**: p. 785-795.
120. Kohavi, R. and G.H. John, *Wrappers for feature subset selection.* Artificial Intelligence, 1997. **97**(1): p. 273-324.
121. Hastie, T., R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning.* Springer series in statistics, ed. Springer. 2001, New York.

Appendix A

Data for the case studies in Chapter 3.

Figure A1. Demand for two-dimensional problem

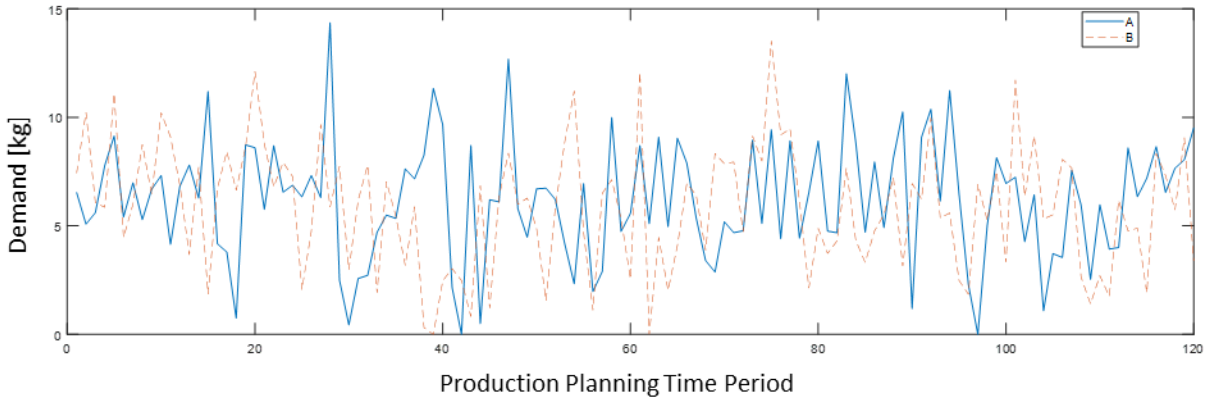

Table A1. Data for two-dimensional problem

		RMA	RMB	A	B
	$W_s^{initial}$ (kg)	∞	∞	0	0
	C_s^{max} (kg)	∞	∞	∞	∞
	h_s ($/kg)			20	20
	u_s($/kg)			400	400
ρ_{is}	T1	-1		1	
	T2		-1		1
	T3	-1		1	
	T4		-1		1

Table A2. Data for two-dimensional problem

	$V_{ij}^{min}/V_{ij}^{max}$ (kg/h)	
	U1	U2
T1	0/3	
T2	0/3	
T3		0/1
T4		0/1

Figure A2. Demand for three-dimensional problem

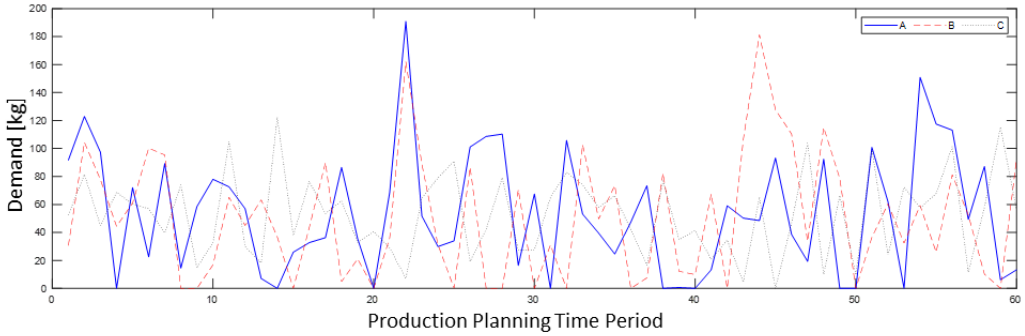

Table A3. Data for three-dimensional problem

		F1	F2	S1	S2	S3	S4	S5	S6	INT1	INT2	A	B	WS	C
$W_s^{initial}$ (kg)		∞	∞	0	0	0	50	0	0	0	0	0	0	0	0
C_s^{max} (kg)		∞	∞	100	100	100	∞	100	100	100	100	∞	∞	∞	∞
h_s ($/kg)												20	20		20
u_s($/kg)												400	400		400
ρ_{is}	T1	-1		1											
	T2			-1	1										
	T3				-1					1					
	T4					1				-1					
	T5					-1						1			
	T6					-1							0.98	0.02	
	T7		-0.95				-0.05	1							
	T8						0.1	-1			0.9				
	T9								1	-0.5	-0.5				
	T10								-1						1

Table A4. Data for three-dimensional problem

	Time (h)	$V_{ij}^{min}/V_{ij}^{max}$ (kg/h)					
	τ_i	U1	U2	U3	U4	U5	U6
T1	2	0/50					
T2	1		0/80				
T3	1			0/60			
T4	2	0/50					
T5	2				0/80		
T6	2				0/80		
T7	4					0/30	
T8	2						0/40
T9	2					0/30	
T10	3						0/40

Figure A3. Demand for the seven-dimensional problem

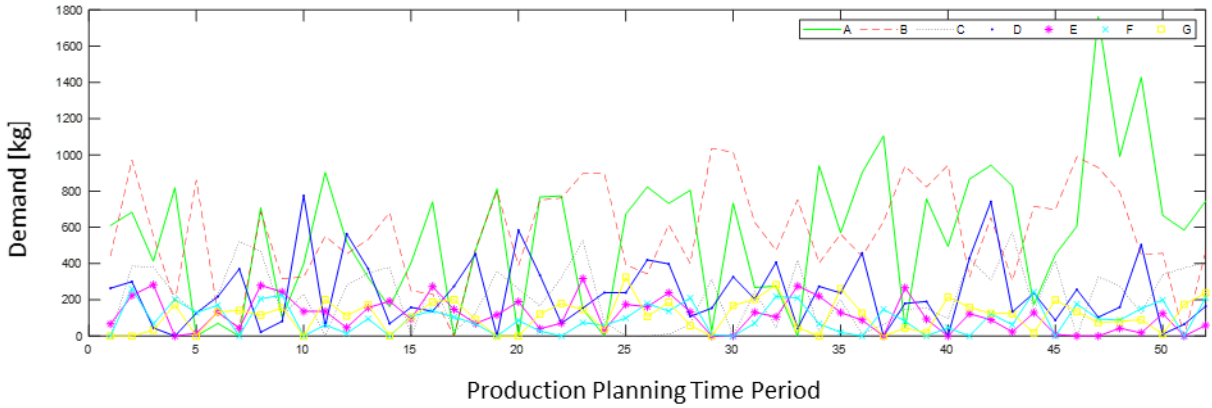

Table A5. Data for seven-dimensional problem

		F1	F2	S1	S2	S3	S4	A	B	C	D	E	F	G
	$W_s^{initial}$ (kg)	∞	∞	0	0	0	0	0	0	0	0	0	0	0
	C_s^{max} (kg)	∞	∞	30	30	30	30	∞	∞	∞	∞	∞	∞	∞
	h_s ($/kg)							20	20	10	10	5	30	40
	$u_s(\$/kg)$							400	500	400	300	200	400	500
ρ_{is}	T1	-1		1										
	T2		-1		1									
	T3	-1		1										
	T4		-1		1									
	T5			-1		1								
	T6				-1		1							
	T7			-1				1						
	T8			-1				1						
	T9					-1			1					
	T10						-1			1				
	T11					-1					1			
	T12						-1					1		
	T13						-1						1	
	T14				-1									1

Table A6. Data for seven-dimensional problem

	Time (h)	$V_{ij}^{min}/V_{ij}^{max}$ (kg/h)					
	τ_i	U1	U2	U3	U4	U5	U6
T1	2	0/10					
T2	2	0/10					
T3	2		0/15				
T4	2		0/15				
T5	1			0/10			
T6	1			0/10			
T7	2				0/15		
T8	1					0/10	
T9	1					0/10	
T10	1					0/10	
T11	1					0/10	
T12	1						0/10
T13	1						0/10
T14	1						0/10

Raw material costs for all raw materials and all the problems is $5/kg.

Appendix B

Data for the classifiers from the case studies in Chapter 3.

Table B1. Parameters for the support vector predictor in case study 1.

Vector	$\alpha_v \times y_v$	$\mathbf{x}_v$	
1	-2.79	5.35381	9.851428
2	-88.11	5.957423	13.24697
3	-0.54	0.367904	2.452296
4	-7.12	10.00091	5.295165
5	-45.01	12.12963	5.242975
6	31.71	12.25763	6.558401
7	5.31	3.043823	19.40842
8	74.91	7.234538	10.66534
9	25.94	6.157956	17.35169
10	5.69	19.02077	4.623234

$\gamma = 0.5, b = 0.9741$

Table B2. Parameter for the neural network predictor in case study 1.

$\mathbf{w}_{input}^T$		$\mathbf{b}_{input}^T$	$\mathbf{w}_h$	b_h
-5.36648	-3.75728	-4.03665	-9.63628	0.7934
10.05838	-11.7339	-11.7753	-16.4666	
-4.28878	-1.67988	-1.62722	-5.88031	
0.838191	1.240411	-1.07853	1.636828	
8.869156	-8.20985	7.649089	16.24723	
7.543368	12.09633	-1.48232	12.58419	
-4.2947	-8.91592	-6.6844	-10.649	
16.44189	7.568852	-5.25047	14.86137	
0.526034	-1.12809	-3.37523	-0.97602	
8.532768	8.379034	1.542781	7.329913	

Appendix C

A detailed air separation model is derived under the following simplifying assumptions:

- Negligible vapor holdups on each tray
- Ideal vapor phases
- Well-mixed streams entering each tray
- Constant pressure drop on each tray
- Vapor liquid equilibrium is established on each tray

Distillation column

Overall Mass Balance

$$\frac{dM_i}{dt} = L_{i-1} + V_{i+1} - L_i - V_i + F_i \tag{1}$$

where i is the index of each tray, starting from the top of the column. M_i is the liquid mole holdup ([mol]) on tray i, L_i and V_i are liquid and vapor molar flow rates, and F_i is the molar feed ([mol/min]). The only nonzero values of F_i are those corresponding to expanded air (EA, u_1), pure air (MA, u_2), and crude oxygen stream (CO).

In the first tray of the high pressure column:

$$\frac{dM_1^h}{dt} = L_{cond} * R_{cond} + V_2^h - L_1^h - V_1^h$$

$$L_{cond} = V_1^h$$

The total liquid leaving the condenser, L_{cond} is equal to the vapor sent to the condenser, V_1. R_{cond} is the reflux ratio in the condenser. The liquid nitrogen and the crude oil sent to the LPC are given by:

$$\text{LN} = (1 - R_{cond}) * L_{cond} - \text{PLNI}$$

$$\text{CO} = L_{40}^{h}$$

In the last tray of the low-pressure column:

$$\frac{dM_{40}^{l}}{dt} = L_{39}^{l} + V_{reb} * R_{reb} - L_{40}^{l} - V_{40}^{l}$$

$$V_{reb} = L_{40}\ (if\ in\ steady\ state)$$

$$\text{POX} = V_{reb}(1 - R_{reb})$$

Finally, all the vapor leaving the first tray of the LPC is pure nitrogen:

$$\text{PNI} = V_1^l$$

Component Balances

$$\frac{d(M_i x_{ij})}{dt} = L_{i-1}x_{i-1,j} + V_{i+1}y_{i+1,j} - L_i x_{i,j} - V_i y_{i,j} + F_i x_{i,j}^{f} \quad (2)$$

where $j \in COMP$ is the index of each component, x_{ij} and y_{ij} are component mole fractions in the liquid and vapor phases, x_{ij}^{f} are the mole fractions of the feed. We can rewrite this equation as:

$$M_i \frac{dx_{ij}}{dt} = L_{i-1}(x_{i-1,j} - x_{i,j}) + V_{i+1}(y_{i+1,j} - x_{i,j}) - V_i(y_{i,j} - x_{i,j}) + F_i(x_{i,j}^{f} - x_{i,j}) \quad (3)$$

Note that, similar to the mass balance, the equations for first, last, and feed trays are modified. For example:

$$M_1^h \frac{dx_{1j}^h}{dt} = L_{cond}R_{cond}(x_{cond,j} - x_{1,j}^h) + V_2^h(y_{2,j}^h - x_{1,j}^h) - V_1^h(y_{1,j}^h - x_{1,j}^h)$$

$$M_{40}^h \frac{dx_{40j}^h}{dt} = L_{39}^h\left(x_{39,j}^h - x_{40,j}^h\right) - V_{40}^h\left(y_{40,j}^h - x_{40,j}^h\right) + \mathrm{MA}_{40}\left(x_{MA,j}^f - x_{40,j}^h\right)$$

Energy balance

$$\frac{d(M_i h_i^L)}{dt} = L_{i-1}h_{i-1}^L + V_{i+1}h_{i+1}^V - L_i h_i^L - V_i h_i^V + F_i h_i^f \tag{4}$$

where $h_i^L = f^{hl}(T_i, P_i)$and $h_i^V = f^{hv}(T_i, P_i)$ are liquid and vapor enthalpies in $\left[\frac{kJ}{mol}\right]$, and h_i^f is the feed enthalpy.

$$h_i^L = \sum_j \left(a_j^L T_i + b_j^L\right)x_{ij}$$

$$h_i^V = \sum_j \left(a_j^V T_i + b_j^V\right)y_{ij}$$

The energy balance can be rewritten as:

$$M_i \frac{dh_i^L}{dt} = L_{i-1}(h_{i-1}^L - h_i^L) + V_{i+1}(h_{i+1}^V - h_i^L) - V_i(h_i^V - h_i^L) + F_i\left(h_i^f - h_i^L\right) \tag{5}$$

Note that, similar to the mass balance, the equations for first, last, and feed trays are modified.

Summation equation

$$\sum_j y_{i,j} = 1 \tag{6}$$

Hydraulic Equation

$$L_i = k_d M_i \tag{7}$$

where $k_d = 0.5\ min^{-1}$ is a tuning constant determined from empirical data.

Vapor-Liquid Equilibrium

$$y_{ij}p_i = \gamma_{ij}x_{ij}p_{ij}^{sat} \tag{8}$$

where p_i is the total pressure on tray i, and $p_{ij}^{sat} = f(T_i)$ is the saturation pressure of pure component j on tray i. Symbol γ_{ij} denotes the liquid activity coefficient describing the non-ideal vapor liquid equilibrium calculated from:

$$\gamma_{i,N2} = \exp\left[\frac{A_{N_2O_2}x_{i,O_2}^2 + A_{N_2Ar}x_{i,Ar}^2 + \left(A_{N_2O_2} + A_{N_2Ar} - A_{O_2Ar}\right)x_{i,O_2}x_{i,Ar}}{RT_i}\right] \tag{9a}$$

$$\gamma_{i,O2} = \exp\left[\frac{A_{N_2O_2}x_{i,N_2}^2 + A_{O_2Ar}x_{i,Ar}^2 + \left(A_{N_2O_2} + A_{O_2Ar} - A_{N_2Ar}\right)x_{i,N_2}x_{i,Ar}}{RT_i}\right] \tag{9b}$$

$$\gamma_{i,Ar} = \exp\left[\frac{A_{N_2Ar}x_{i,N_2}^2 + A_{O_2Ar}x_{i,O_2}^2 + \left(A_{N_2Ar} + A_{O_2Ar} - A_{N_2O_2}\right)x_{i,N_2}x_{i,O_2}}{RT_i}\right] \tag{9c}$$

where R is the ideal gas constant and the coefficient A_{jk} account for the liquid phase interactions between components j and k. These can be calculated using Margules equations.

Condenser

$$L_{cond} = V_1^h \tag{10}$$

L_{cond} is the liquid molar flow leaving the condenser, which is equal to V_1, the top stage vapor rate.

$$Q_{cond} = L_{cond} \cdot h_1^V - L_{cond}h_{condenser}^L = L_{cond}(h_1^V - h_{cond}^L) \tag{11}$$

where h_1^V is the top tray molar vapor enthalpy, h_{cond}^L is the liquid molar enthalpy in the condenser and Q_{cond} is the heat extracted from the condensation of vapor.

Because all of the inlet vapor is condenser, the composition of the outlet liquid is the same as the inlet vapor

$$y_{1,j} = x_{condenser,j} \tag{12}$$

where y_{1j} is the top stage composition fo component j and $x_{condenser,j}$ is the liquid molar composition in the condenser of component j.

Furthermore, because the liquid is assumed to be saturated, the condenser pressure can be calculated using the saturation pressure of each component and the liquid composition:

$$p_{cond} = \sum_j x_{condenser,j} p_j^{sat} \tag{13}$$

Note that p_{cond} corresponds to the bubble point pressure under the assumption that Raoult's law applies. In this study, the condenser pressure is treated as a modeling specification.

Reboiler

Mass balance

$$\frac{dM_{reboiler}}{dt} = L_{40} - V_{reboiler} - L_{drain} \tag{14}$$

where L_{40} is the liquid flow from the bottom tray in the column, L_{drain} is the drain flow from the reboiler, and $V_{reboiler}$ is the waste vapor flow from the reboiler.

Component balance

$$\frac{dx_{reboiler}}{dt} = \frac{L_{40}\left(x_{40,j} - x_{reboiler,j}\right) - V_{reboiler}\left(y_{reboiler,j} - x_{reboiler,j}\right)}{M_{reboiler}} \tag{15}$$

Energy balance*:*

As there is no material or energy accumulation in the expansion valve, the expansion valve and the reboiler models are lumped together for simplicity.

$$\left(\frac{\partial h_{reb}^{L}}{\partial T_{reb}}\bar{T}_{reb} + \sum \frac{\partial h_{reb}^{L}}{\partial x_{reb,j}}\bar{x}_{reb,j}\right) = \frac{L_{40}(h_{40}^{L} - h_{reb}^{L}) - V_{reb}(h_{reb}^{V} - h_{reb}^{L}) + Q_{reb}}{M_{reb}}$$

where $Q_{reb} = Q_{condenser}$

The pressure at the reboiler was held constant at 1.3bar.

$$\bar{x}_{rebj} = \frac{dx_{reb,j}}{dt} = \frac{L_{30}\left(x_{30,j} - x_{reboiler,j}\right) - V_{reboiler}\left(y_{reboiler,j} - x_{reboiler,j}\right)}{M_{reboiler}}$$

$$\bar{T}_{reb} = \frac{dT_{reb}}{dt} = -\frac{\sum_j \left[x_{rebj}\sum_k \left(\frac{\partial K_{reb,j}}{\partial x_{reb,k}}\bar{x}_{reb,k}\right) + K_{reb,j}\bar{x}_{reb,j}\right]}{\sum_j x_{reb,j}\frac{\partial K_{reb,j}}{\partial T_{reb}}}$$