Computers and Chemical Engineering 184 (2024) 108630

Contents lists available at ScienceDirect

Computers and Chemical Engineering

journal homepage: www.elsevier.com/locate/compchemeng

Algebraic surrogate-based flexibility analysis of process units with complicating process constraints

Tim Forster [a], Daniel Vázquez [a,b], Isabela Fons Moreno-Palancas [a], Gonzalo Guillén-Gosálbez [a,*]

[a] Department of Chemistry and Applied Biosciences, Institute for Chemical and Bioengineering, ETH Zurich, Vladimir-Prelog-Weg 1, Zurich 8093, Switzerland
[b] IQS School of Engineering, Universitat Ramon Llull, Via Augusta 390, Barcelona 08017, Spain

ARTICLE INFO

Keywords:
Flexibility analysis
Surrogate model
Symbolic regression
Bioprocess

ABSTRACT

Flexibility analyses are widespread in chemical engineering to quantify allowed deviations from nominal conditions. Standard approaches to perform flexibility analysis can be hard to apply if process constraints are difficult to handle, as it happens in bioprocesses with dynamic constraints. Here, focusing on the computation of the traditional flexibility index in problems with complicating constraints, we apply symbolic regression to build algebraic expressions of the said complicating constraints, simplifying the flexibility analysis of complex process models by enabling the application of state-of-the-art deterministic solvers. Our approach is applied to ethanol production in fed-batch operation mode and a chromatographic process. The performance is assessed in terms of model building time, predictive accuracy of the model, and the time required to solve the flexibility formulations. Overall, our approach, which focuses on computing the original flexibility index proposed in the literature, provides an alternative way to analyse the flexibility of processes entailing complicating constraints.

1. Introduction

Uncertainty is always present in science and engineering. This uncertainty can reveal itself in, for example, product demands (Petkov and Maranas, 1997), supply chain and scheduling activities (Ehrenstein et al., 2019), and even in process design and operation (Pistikopoulos, 1995). A broad overview of various aspects of uncertainty, specifically in the Process Systems Engineering (PSE) field, is given by the works by Sahinidis (2004), Li and Ierapetritou (2008a), and Grossmann et al. (2014). When uncertainty is not taken into account, designing and optimizing process units assuming deterministic values for the uncertain parameters can lead to suboptimal solutions or, in the worst case, to infeasibilities during operation (Ben-Tal and Nemirovski, 2002; Grossmann et al., 1983; Li et al., 2011). Thus, it is common to embed uncertainty in the specifications of the problem, e.g., in the field of pharmaceutical development, the guidelines of the International Council for Harmonisation of Technical Requirements for Pharmaceuticals for Human Use (ICH) define that critical quality attributes (CQAs) are valid within a given acceptable range (US Food and Drug Administration (FDA), 2010), considering variations due to uncertain input conditions.

Accounting for uncertainties in optimization problems during the early design, and operation phases is especially important for chemical processes. This is because optimal solutions tend to meet process constraints and quality requirements as deterministic inequalities or equalities, so any perturbation over the nominal conditions, often occurring in such processes, may have strong implications on their feasibility. There are two main mathematical methods in operations research to account for uncertainties in optimization problems, namely stochastic programming (Birge and Louveaux, 2011; Ierapetritou and Pistikopoulos, 1994; Li and Ierapetritou, 2012; Marti and Kall, 1995; Prékopa, 2011; Shapiro et al., 2021) and robust optimization (Ben-Tal et al., 2009; Ben-Tal and Nemirovski, 2002; Li and Ierapetritou, 2008b; Lin et al., 2004). Li and Grossmann (2021) considered chance-constrained programming as another approach for optimization problems under uncertainty, yet (arguably) it could also be regarded as a generalization of robust optimization, in which distributions are specified for the uncertainties and a level of probability is defined to satisfy constraints (Grossmann et al., 2016).

The flexibility index is an alternative approach for accounting for uncertainties that has been used mainly in process design (Pistikopoulos, 1995). Developed by the PSE community back in the 1980s (Grossmann et al., 1983; Halemane and Grossmann, 1983; Swaney and Grossmann, 1985a, 1985b), its primary goal is to assess the ability of a design to remain feasible against variations in the parameter values during the plant operation (Boukouvala et al., 2010; Grossmann et al., 1983). In

* Corresponding author.
E-mail address: gonzalo.guillen.gosalbez@chem.ethz.ch (G. Guillén-Gosálbez).

https://doi.org/10.1016/j.compchemeng.2024.108630
Received 20 July 2023; Received in revised form 24 January 2024; Accepted 15 February 2024
Available online 16 February 2024

Nomenclature

Abbreviations

ANN	Artificial neural network
ALAMO	Automated learning of algebraic models for optimization
BMS	Bayesian machine scientist
CPU	Central processing unit
CS	Case study
CQA	Critical quality attribute
FDA	Food and drug administration
ICH	International Council for Harmonisation of Technical Requirements for Pharmaceuticals for Human Use
KKT	Karush–Kuhn–Tucker
LHS	Latin hypercube sampling
MCMC	Markov-chain Monte Carlo
ODE	Ordinary differential equation
PDE	Partial differential equation
SR	Symbolic regression

Sets

E	$\{e \mid e$ is a symbolic mathematical expression$\}$
G	$\{g \mid g$ is a non-complicating constraint$\}$
H	$\{h \mid h$ is a complicating constraint$\}$
I	$\{i \mid i$ is a sample$\}$
J	$\{j \mid j$ is a constraint$\}$
K	$\{k \mid k$ is an uncertain parameter$\}$
N	$\{n \mid n$ is a control variable$\}$
T	set of uncertain parameters that maintain the process feasible

Parameters

d	design parameters of the process under consideration
M	big-M reformulation parameter
$\Delta\theta_k^{min}$ and $\Delta\theta_k^{max}$	maximum upper and lower deviation from a nominal point of the uncertain parameter k
$\underline{\theta_k}$ and $\overline{\theta_k}$	lower and upper bounds of the uncertain parameter k

Variables

f_j	process constraint
$\widehat{f}_g$	non-complicating process constraint
$\widetilde{f}_h$	complicating process constraint
s_j, s_g, and s_h	slack variables of constraints f_j, $\widehat{f}_g$, and $\widetilde{f}_h$
t	Time
u	upper bound for the constraint f_j
y_j, y_g, and y_h	binary variable of constraints f_j, $\widehat{f}_g$, an d $\widetilde{f}_h$
z_n	control variable n
δ	scaled deviation from nominal point
θ_k	uncertain parameter k
θ_k^c	critical value of the uncertain parameter k
θ_k^N	nominal operating point of the uncertain parameter k
γ_e	symbolic expression e
λ_j, λ_g, and λ_g	Lagrange multiplier of constraints f_j, $\widehat{f}_g$, and $\widetilde{f}_h$
ω_i	feature vector of sample i used for the model training
$\mathscr{L}$	Lagrange polynomial
$\mathscr{DL}$	description length of Bayesian machine scientist

essence, this is done by quantifying the feasibility level of a given design, which is related to whether a process remains feasible or otherwise becomes infeasible within a given range. Mathematically, the *feasibility function* can be calculated by solving a min-max-optimization problem, which will be discussed in detail later in this work. Grossmann et al. (1983) geometrically interpret this feasibility function as the "depth" of the feasible region since it quantifies a deviation from the nominal constraints. Based on this concept, the authors describe the flexibility index (Grossmann et al., 1983; Swaney and Grossmann, 1985a, 1985b), which characterizes the size of the region of feasible operation (T) in the space of uncertain parameters. This region T should be a subset of the entire feasible region (Zhang et al., 2016). In other words, the flexibility index describes the maximum range over which the involved uncertain parameters can vary (independently) such that the process remains feasible (Grossmann et al., 1983; Pulsipher et al., 2019). Alternatively, other metrics to quantify process flexibility were put forward. Those methods include for example the resilience index (Morari et al., 1985), and stochastic measures such as the design reliability (Kubic and Stein, 1988) and the stochastic flexibility index (Pistikopoulos and Mazzuchi, 1990; Straub and Grossmann, 1993, 1990). Specifically, the stochastic flexibility index was developed to tackle the limitation of the flexibility index to address discrete and continuous uncertainties at the same time (Straub and Grossmann, 1990), and to enable the use of arbitrary probability distributions of the uncertain parameters in the analysis (Rogers and Ierapetritou, 2015a).

The flexibility index can be computed using deterministic mathematical models (Pistikopoulos, 1995; Pulsipher et al., 2019) as long as process constraints are described in a closed-form algebraic manner (Floudas et al., 2001; Ierapetritou, 2001; Pistikopoulos and Ierapetritou, 1995; Straub and Grossmann, 1993). Specifically, the main methods to quantify the flexibility index include vertex searches (Grossmann et al., 1983; Swaney and Grossmann, 1985b, 1985a), active set strategies with KKT reformulations (Grossmann and Floudas, 1987), or branch-and-bound approaches (Ostrovsky et al., 1994) based on the evaluation of the lower and upper bounds of the feasibility function. Since global optimality cannot be guaranteed using local solvers for such bounding methods (Migdalas et al., 1998), a global optimization approach was developed using reformulation and relaxation approaches for the feasible region (Floudas et al., 2001). We note that some of these approaches rely on specific convexity assumptions (Goyal and Ierapetritou, 2003, 2002; Grossmann and Floudas, 1987).

When some constraints are not available in algebraic closed form, analysing process flexibility becomes much more complex and a straightforward computation of the flexibility index with state-of-the-art deterministic solvers is not possible anymore. This might happen, for instance, if the only knowledge about the system consists of observations of input and output data due to limited process understanding (Boukouvala and Ierapetritou, 2012). Additionally, very complex underlying process dynamics (ordinary or partial differential equations) can be another reason why constraints are difficult to derive in a closed-form manner (Ding and Ierapetritou, 2021). Even if some knowledge about the process dynamics can be described by differential equations that could be discretized (i.e., orthogonal collocation on finite elements Carey and Finlayson 1975), finding a solution might still be challenging due to the size of the reformulated optimization problem.

Several works applied adaptive sampling techniques with Kriging interpolation (Krige, 1951), also known as Gaussian process regression (Rasmussen and Williams, 2006), to perform flexibility analyses when dealing with situations where closed-form models for process constraints are inexistent or hard to build (Boukouvala et al., 2011; Boukouvala and Ierapetritou, 2012; Ding and Ierapetritou, 2021; Rogers and Ierapetritou, 2015b, 2015a; Wang and Ierapetritou, 2017). Broadly speaking, these methods are used to approximate the feasibility function, namely the function that evaluates the feasibility of the model for given values of the decision variables and the parameters. Such data-driven strategies can handle process models with non-convex

feasible regions (Rogers and Ierapetritou, 2015b, 2015a). Similarly, other works substitute the Gaussian process models with neural networks (Metta et al., 2021). In a very recent work by Sachio et al. (2023), the authors developed a highly flexible framework that performs a design space identification followed by a design space analysis. The researchers used a Sobol sampling approach with a subsequent approximation of the design space by alpha shapes, where the usage of alpha shapes was also successfully described in earlier works for feasibility analysis (Banerjee and Ierapetritou, 2005). All the methods mentioned in this paragraph approximate the feasibility function with a surrogate and they do not rely on the original deterministic flexibility index, but rather they use alternative flexibility metrics.

Here, we shall develop an alternative approach for flexibility problems, focusing on the computation of the flexibility index where challenging process dynamics or hard-to-model process constraints are encountered. While more refined flexibility metrics have been proposed (Pistikopoulos and Mazzuchi, 1990; Straub and Grossmann, 1990), we focus on the original flexibility index metric due to the already existing methods for its computation applicable to analytical closed-form models, into which we reformulate process models with complicating constraints as explained later in the article. In the following, we use the term "complicating constraints" to describe hardly accessible or completely inaccessible constraints, that is, constraints that are either hard to model in algebraic form and/or hard to handle in an optimization model. In essence, here we shall replace those constraints with algebraic surrogates built with a symbolic regression algorithm (SR). These algebraic surrogates are hence subsequently incorporated into the original flexibility analysis formulation, thereby simplifying the flexibility analysis. SR algorithms aim to find the model structure and associated parameters that fit some data. Compared to algorithms like ALAMO or ALVEN that restrict the search to a specific set of functions, general SR approaches make use of symbolic expression trees that can represent a very large number of plausible algebraic surrogate models (Cozad and Sahinidis, 2018). Here, the best model in the symbolic tree can be identified following different approaches and applying some fitting criteria. These include the formulation and solution of an MINLP problem (Cozad and Sahinidis, 2018), where binary variables encode the model structure, and continuous ones its parameters, or the application of stochastic search approaches (Cranmer et al., 2020; Diveev and Shmalko, 2021; Guimerà et al., 2020). For example, Cranmer et al. (2020) created the open-source algorithm PySR, a multi-population evolutionary algorithm, which is freely available in Python (Cranmer, 2023, 2020). There are also algorithms that are available as proprietary software, such as Eureqa (Schmidt and Lipson, 2009) or TuringBot (2023). To build the surrogate models in this work, however, we use an SR method developed by Guimerà et al. (2020), based on a Markov-Chain Monte Carlo approach to identify the most suitable closed-form expression to represent given data. One of the advantages of SR is that it does not assume a predetermined model structure or a reduced set of alternative model structures (e.g., like in the automated learning of algebraic models for optimization (ALAMO) approach (Wilson and Sahinidis, 2017), or the above mentioned HDMR approach). The user only defines some allowable mathematical operations (i.e., addition, multiplication, subtraction, etc.) that are used in a symbolic tree to build plausible expressions to explain the data at hand. This symbolic tree can be seen as a superstructure of mathematical expressions from which the most suitable one and its associated parameters must be identified using specific algorithms. SR was successfully applied in many different fields, such as distillation (Ferreira et al., 2019b, 2019a; McKay et al., 1997), food extrusion process (McKay et al., 1999), process control (Keane et al., 1993), or the discovery of physical laws (Cranmer et al., 2020; Schmidt and Lipson, 2009). Moreover, the BMS was also previously applied by some of us to approximate process simulations of carbon capture plants (Negri et al., 2022), to model the link between energy-related impacts and socioeconomic drivers in macro-economic studies (Vázquez et al., 2022), and for surrogate-based global optimization of process units and flowsheets by coupling SR with deterministic global optimization (Forster et al., 2023).

Our proposed approach represents an alternative way to handle complicating constraints in flexibility problems that does not rely on any discretization technique, like those applied to differential equations, thereby avoiding adding auxiliary variables that increase the dimensionality of the optimization problem. Additionally, no pre-defined model structure is assumed for the surrogate model replacing the complicating constraints. Instead, an SR algorithm, the BMS, creates an algebraic model from a set of samples of the functions describing the complicating process constraints. We show the advantages of this approach in two case studies covering a chromatographic column of an antibody production process and bioethanol production in fed-batch operation mode. To the best of our knowledge, this is the first work that combines SR with the initially defined flexibility index problem, giving rise to a hybrid optimization problem where some constraints are replaced with algebraic surrogates. In the end, the most appropriate approach to quantify flexibility performance in the presence of complicating constraints will depend on the problem at hand and the goal and scope of the analysis, including the selection of the flexibility metric to be evaluated.

The remainder of the article is organized as follows: First, the problem statement is described, followed by the methodology. Afterward, two case studies are introduced, and the results are subsequently discussed. Finally, the conclusions of the work are drawn.

2. Problem statement

Here, without loss of generality, we shall consider an existing process or process unit, where a known and fixed process design (i.e., equipment dimensions) is given by variable d. Additionally, there are K uncertain parameters $\theta_k, k \in K$, which have a given nominal value of θ_k^N. Last, there are N control variables, with a value z_n, $n \in N$, that can be adjusted during the operation to regain feasibility.

Within this process, a set of J process constraints f_j, $\forall j \in J$ (i.e., material balances, process or product specifications or restrictions, etc.) need to be considered, as stated in Eq. (1):

$$f_j(d, z, \theta) \leq 0, \; j \in J \tag{1}$$

Considering the above, we want to assess how far the uncertain parameters θ can deviate from the nominal operating point θ^N, such that the process remains feasible, i.e., we are interested in the flexibility index problem as described later in the next section. To quantify the flexibility of a process, the feasibility function given in Eq. (2) must be assessed. To do so, the min-max-optimization problem shown in Eq. (2) must be solved:

$$\psi(d, \theta) = \min_{z} \max_{j \in J} \left\{ f_j(d, z, \theta) \right\} \tag{2}$$

In this expression, $\psi(d, \theta)$ represents the feasibility function for a given design d and a realization of the uncertain parameters θ. However, some of the process constraints $f_j, j \in J$, might be very challenging to be evaluated, or might not even be directly accessible as closed-form algebraic equations. As a consequence, they cannot be directly included in the formulation given in Eq. (2). Complicating constraints might be encountered in complex systems (i.e., involving complex process dynamics, with complex unit operations hard to model mechanistically).

Hence, we divide the set of constraints J into two proper subsets $G \subset J$ and $H \subset J$, as shown in Eq. (3). Set G contains process constraints $\widehat{f}_g, g \in G$ that are non-complicating, i.e., clearly defined by an algebraic equation that can be easily incorporated into the model described in Eq. (2) and handled numerically in an efficient manner. Set $H \subset J$, on the other hand, contains complicating constraints, denoted by $\widetilde{f}_h, h \in H$, which cannot be incorporated directly into the model in a straightforward manner. Note

that whether one constraint should be considered complicating or not might depend on the specific case and the numerical performance of the standard approach.

$$\begin{aligned} &\widehat{f}_g(d,z,\theta) \le 0, \ \ g \in G \\ &\widetilde{f}_h(d,z,\theta) \le 0, \ \ h \in H \end{aligned} \tag{3}$$

The idea here is to replace the complicating constraints in Eq. (2) with algebraic surrogate models that are constructed by solving an SR problem. Herein, we shall identify such a surrogate model without assuming a pre-defined model structure, as discussed next.

3. Methodology

For the sake of completeness, we will first present the flexibility index formulation developed by Grossmann et al. (1983), Halemane and Grossmann (1983), and Swaney and Grossmann (1985a, 1985b), which is taken as a basis to derive our approach. The reader is referred to these works for more details and further mathematical insights. For simplicity, during the subsequently shown derivation, we use the set J to describe all the constraints, where we split this set into the two subsets G and H – as shown in Section 2 – in the very end of the derivation. After that, we describe how the surrogate models can be incorporated in the flexibility formulation. Last, we discuss how to build these surrogate models and assess their performance.

3.1. Fundamentals of feasibility and flexibility

Consider the formulation in Eq. (4) that aims to calculate the feasibility function $\psi(d,\theta)$ of a given design d and a specific realization of θ_k, $k \in K$, where some control variables $z_n, n \in N$ are present (Grossmann et al., 1983; Halemane and Grossmann, 1983; Swaney and Grossmann, 1985a, 1985b):

$$\psi(d,\theta) = \min_{z} \max_{j \in J} \{f_j(d,z,\theta)\} \tag{4}$$

Using an upper bound u for the constraints $f_j, j \in J$, we can reformulate the min-max formulation into the following single-level problem:

$$\begin{aligned} &\psi(d,\theta) = \min_{z,u} u \\ &s.t. \ \ f_j(d,z,\theta) \le u, \quad \forall j \in J \end{aligned} \tag{5}$$

Formulation (5) seeks the smallest u such that each constraint f_j results in a value less or equal to u. Overall, a value of $\psi(d,\theta) \le 0$ means the process is feasible for a given realization of d and θ. On the other hand, $\psi(d,\theta) > 0$ implies that the process is infeasible for these specific values of d and θ.

The feasibility formulation seeks the worst value of $\psi(d,\theta)$ over the entire uncertain parameters space $\theta \in T$. This problem can be formulated as the following tri-level optimization model, which provides the *feasibility test* function $\chi(d)$ given in Eq. (6).

$$\begin{aligned} \chi(d) &= \max_{\theta \in T} \psi(d,\theta) \\ &= \max_{\theta \in T} \min_{z} \max_{j \in J} \{f_j(d,z,\theta)\} \end{aligned} \tag{6}$$

In formulation (6), if $\chi(d) \le 0$, the process is feasible for the entire space of the uncertain parameters Θ. Using formulation (5) given above for the feasibility function $\psi(d,\theta)$, the feasibility test problem in Eq. (6) can be reformulated as a bilevel optimization problem shown in Eq. (7)

$$\begin{aligned} &\chi(d) \ \ = \max_{\theta} \psi(d,\theta) \\ &s.t. \quad \psi(d,\theta) = \min_{z,u} u \\ &\qquad s.t. \ f_j(d,z,\theta) \le u, \quad \forall j \in J \\ &\qquad \theta \in T \end{aligned} \tag{7}$$

Grossmann et al. (1983) proposed an approach to quantify and identify the largest possible uncertainty set $\theta \in T$, such that the process is still feasible over the entire range of θ. The authors described this as the *flexibility index* problem, which is given in Eq. (8)

$$\begin{aligned} &FI \ \ = \max_{\delta \in \mathbb{R}_{0\le}} \delta \\ &s.t. \quad \chi(d) = \max_{\theta} \psi(d, \ \theta) \le 0 \end{aligned} \tag{8}$$

where, FI represents the flexibility index, and δ should be a nonnegative real number ($\mathbb{R}_{\ge 0}$). The newly introduced variable δ scales the uncertainty set T, which is therefore subsequently denoted by $T(\delta)$. In other words, δ can be regarded as a scaled deviation from a nominal point θ^N, such that the realization of θ results in a feasible solution. The goal is to maximize the mentioned set $T(\delta)$, under which there exists the possibility of recovering feasibility through the control variable z. In their original work, Swaney and Grossmann (1985a, 1985b) showed that the bilevel problem given in Eq. (8) can be reformulated. Instead of searching for the largest possible set $T(\delta)$ by maximizing δ, the authors showed that it is equivalent to looking for the minimum δ such that the solution is located precisely on the boundary ($\psi(d,\theta) = 0$). In other words, one is looking for the constraint that is closest to the nominal operating point. This reformulation can therefore be expressed as shown in Eq. (9)

$$\begin{aligned} &FI \ \ = \min_{\delta \in \mathbb{R}_{0\le}} \delta \\ &s.t. \quad \chi(d) = \max_{\theta} \psi(d, \ \theta) = 0 \end{aligned} \tag{9}$$

The flexibility index problem shown in Eq. (9) ensures that the feasibility function is precisely zero. Using the definition of the feasibility test problem given in Eq. (7), the flexibility index problem can be reformulated as follows:

$$\begin{aligned} &FI \ \ = \min_{\delta \in \mathbb{R}_{0\le}} \delta \\ &s.t. \quad \chi(d) = \max_{\theta} \psi(d,\theta) = 0 \\ &\qquad s.t. \ \ \psi(d,\theta) = \min_{z} u \\ &\qquad f_j(d,z,\theta) - u + s_j = 0, \qquad \forall j \in J \\ &\qquad \theta \in T(\delta) \end{aligned} \tag{10}$$

Where the inequality constraints of problem (7) are expressed as equality constraints using nonnegative slack variables, s_j. The resulting flexibility index problem is challenging due to the non-differentiability of max-min-max (or min-max-min) functions. To tackle this challenge, we can substitute the innermost optimization problem with its Karush–Kuhn–Tucker (KKT) conditions (Grossmann et al., 2014). The Lagrange function $\mathscr{L}(\mathrm{d},\theta)$ of this innermost problem can be formulated as follows:

$$\mathscr{L}(d,\theta) = u + \sum_j \left(\lambda_j \left(f_j(d,z,\theta) - u + s_j\right)\right) \tag{11}$$

Where λ_j represents the Lagrange multipliers for constraint f_j. Subsequently, the corresponding stationary (12) and complementarity (13) conditions for problem (10) therefore read as follows:

$$\begin{aligned} &\frac{\partial \mathscr{L}(d,\theta)}{\partial u} = 0 = 1 - \sum_j \lambda_j \\ &\frac{\partial \mathscr{L}(d,\theta)}{\partial z_n} = 0 = \sum_j \lambda_j \frac{\partial f_j(d,z,\theta)}{\partial z_n}, \qquad \forall n \in N \\ &\frac{\partial \mathscr{L}(d,\theta)}{\partial \lambda_j} = 0 = f_j(d,z,\theta) - u + s_j, \qquad \forall j \in J \end{aligned} \tag{12}$$

$$\begin{aligned} &\lambda_j s_j = 0, \quad \forall j \in J \\ &\lambda_j, s_j \ge 0, \quad \forall j \in J \end{aligned} \tag{13}$$

In 1987, Grossmann and Floudas (1987) described how problem (10) can be reformulated into a mixed-integer nonlinear program (MINLP) by applying an active set strategy where some constraints might be inactive in the optimal solution. The usage of active set methods requires making discrete choices on the complementarity conditions $\lambda_j s_j$. Therefore, it is

necessary to introduce binary variables $y_j \in \{0,1\}$ that establish whether a constraint is active ($y_j = 1$) or not ($y_j = 0$). Furthermore, the KKT complementarity conditions are formulated using the following two inequalities in Eq. (14).

$$\begin{aligned} &s_j \le M\big(1-y_j\big), \quad \forall j \in J \\ &\lambda_j \le y_j, \quad \forall j \in J \end{aligned} \tag{14}$$

where, M represents a large enough parameter that acts as the upper bound for the slack variables s_j. Properly selecting M is one of the main drawbacks of this method since it is hard to define tight bounds for the Lagrange multipliers. If M is too small, the solution obtained with the reformulation in Eq. (14) will not coincide with the optimum of the original problem, since this value would act as an active constraint. On the other hand, an excessively large M often causes numerical instabilities (Cococcioni and Fiaschi, 2021). Consequently, its value must be selected in accordance with the problem, which might not be easy. In addition to the transformations mentioned above, another constraint could be added that enforces the number of potential sets of active constraints to be lower or equal to $|N|+1$, where $|N|$ stands for the number of control variables z (Grossmann and Floudas, 1987). For specific mathematical details, the reader is referred to the original work of Grossmann and Floudas (1987), and the more recent works by Ochoa and Grossmann (2020) and Pulsipher et al. (2019).

Although there are several options to describe the set $T(\delta)$, in this work, we restrict our approach and the discussed case studies to a rectangular form of $T(\delta)$. Therefore, the constraint $\theta \in T(\delta)$ given in Eq. (10) can be expressed by the two inequality constraints shown in Eq. (15). The reader is referred to the work of Pulsipher et al. (2019), which addresses the case of an ellipsoidal form of $T(\delta)$.

$$\begin{aligned} &\theta_k^N - \delta\Delta\theta_k^{min} \le \theta_k \\ &\theta_k \le \theta_k^N + \delta\Delta\theta_k^{max} \end{aligned} \tag{15}$$

Using the above-shown reformulation techniques and assumptions, the reformulated flexibility index problem can be expressed as shown in Eq. (16)

$$\begin{aligned} FI \quad &= \min_{\delta} \delta \\ s.t. \quad & f_j(d,z,\theta) - u + s_j = 0, \quad \forall j \in J \\ & \sum_j \lambda_j = 1 \\ & \sum_j \lambda_j \frac{\partial f_j(d,z,\theta)}{\partial z_n} = 0, \quad \forall n \in N \\ & s_j \le M\big(1-y_j\big), \quad \forall j \in J \\ & \lambda_j \le y_j, \quad \forall j \in J \\ & \sum_j y_j \le |N|+1 \\ & \theta \in T(\delta) \\ & \lambda_j \ge 0, \quad \forall j \in J \\ & s_j \ge 0, \quad \forall j \in J \\ & \delta \ge 0 \end{aligned} \tag{16}$$

3.2. Flexibility index formulation with complicating constraints

As already said, here we define as complicating constraints those that are either hard to model explicitly or lead to complex expressions hard to handle numerically. Such a situation might arise, for example, in dynamic systems with constraints on temporal profiles, or in process models with complex unit operations whose behaviour is hard to model mechanistically. In the former case, discretization methods such as orthogonal collocation (Carey and Finlayson, 1975) might be applied, but this will likely result in complex models posing numerical challenges (i.e., convergence problems, entrapment in low-quality local optima, etc.). On the contrary, by non-complicating constraints, we mean constraints that are directly accessible as standard algebraic expressions. To be able to use the flexibility index formulation in Eq. (16), we will follow the approach depicted in Fig. 1.

Therefore, we introduce the two proper subsets $G \subset J$ and $H \subset J$ for the non-complicating and complicating constraints, respectively. With this, the original flexibility index problem given in Eq. (16) is reformulated as given by Eq. (17), while inheriting the assumptions of Eq. (16):

$$\begin{aligned} FI \quad &= \min_{\delta} \delta \\ s.t. \quad & \widehat{f}_g(d,z,\theta) - u + s_g = 0, \quad \forall g \in G \\ & \widetilde{f}_h(d,z,\theta) - u + s_h = 0, \quad \forall h \in H \\ & \sum_g \lambda_g + \sum_h \lambda_h = 1 \\ & \sum_g \lambda_g \frac{\partial \widehat{f}_g(d,z,\theta)}{\partial z_n} + \sum_h \lambda_h \frac{\partial \widetilde{f}_h(d,z,\theta)}{\partial z_n} = 0, \quad \forall n \in N \\ & s_g \le M\big(1-y_g\big), \quad \forall g \in G \\ & s_h \le M(1-y_h), \quad \forall h \in H \\ & \lambda_g \le y_g, \quad \forall g \in G \\ & \lambda_h \le y_h, \quad \forall h \in H \\ & \sum_g y_g + \sum_h y_h \le |N|+1 \\ & \theta \in T(\delta) \\ & \lambda_g \ge 0, \quad \lambda_h \ge 0, \quad \forall g \in G,\ h \in H \\ & s_g \ge 0, \quad s_h \ge 0, \quad \forall g \in G, h \in H \\ & \delta \ge 0 \end{aligned} \tag{17}$$

As stated in Section 2, $\widehat{f}_g, g \in G$ represent the non-complicating constraints, whereas the complicating constraints are denoted by $\widetilde{f}_h$, $h \in H$. Due to the introduction of the two subsets G and H, also the slack variables s_j, the Lagrange multipliers λ_j, and the binary variables y_j must be split into the two respective subsets. This requires adjusting the indices in formulations (16) and (17). It is worth mentioning that this split of J into G and H, does not alter the total number of constraints involved in the problem.

As discussed in the introduction, a situation with complicating or unknown constraints was also addressed in the works by Rogers and Ierapetritou (2015b, 2015a), where the authors modelled the feasible region boundaries using surrogate models. These trained surrogates could then be used to approximate the stochastic flexibility index (Straub and Grossmann, 1993), which can consider probabilistic information. The authors overcome the challenge of not having available closed-form expressions for process constraints by using a Kriging binary classification method, which allows to iteratively approximate the feasible region. With the trained classification models, the authors evaluated a range of uncertain parameter combinations and assessed if these realizations were either feasible or infeasible. However, our approach differs from these works in several ways. First and foremost, Rogers and Ierapetritou (2015a, 2015b) used their surrogate model to evaluate the stochastic flexibility index (Straub and Grossmann, 1990), which measures the probability of feasible operation, while we use the

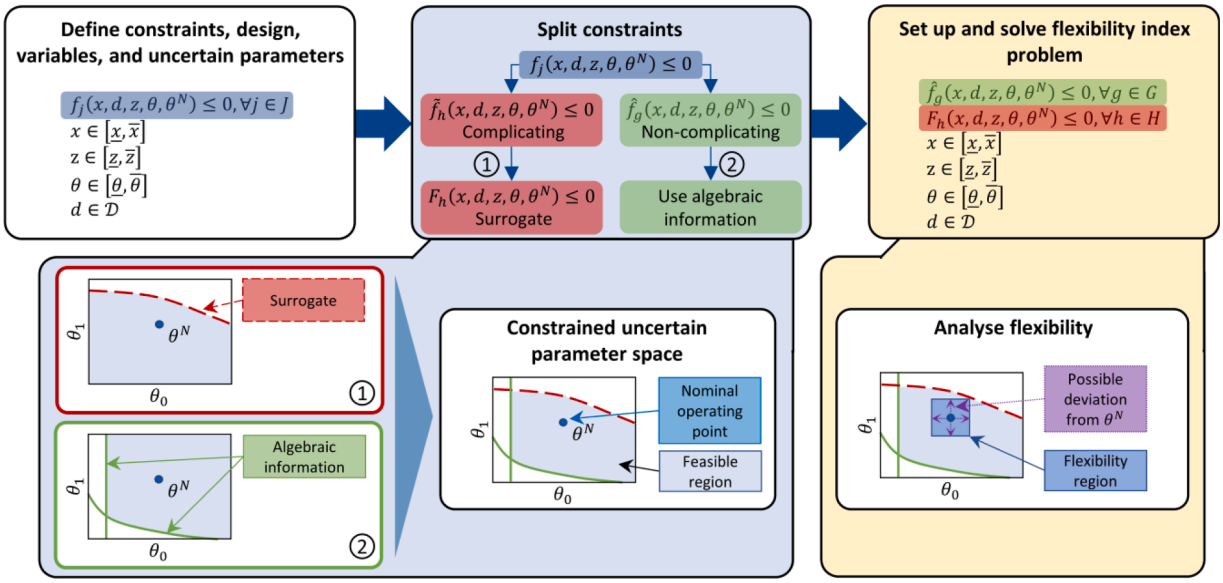

Fig. 1. Overview of the discussed procedure in Sections 3.2 and 3.3. After the constraints f_j are defined (top left white box), which are subsequently split into complicating (red, $\tilde{f}_h$) and non-complicating (green, $\hat{f}_g$) constraints (top central box with blue background). In step 1, the complicating constraints are approximated by using surrogate models. In step 2, the available algebraic information about the non-complicating constraints are used (lower boxes in blue background). Last, the information is combined to solve the flexibility index problem (right boxes with yellow background) For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article..

surrogate to incorporate it into the originally proposed deterministic flexibility index formulation (Grossmann et al., 1983; Halemane and Grossmann, 1983; Swaney and Grossmann, 1985a, 1985b). Hence, we quantify the original flexibility index, which measures the maximum allowable perturbation of parameters within which the process remains feasible, so probabilistic information is not considered in the calculations. Second, we do not use any classification approach, but rather a regression approach. The output of the surrogate model in our work is a continuous variable that determines the value of the constraint for given values of the decision variables and parameters. Third, instead of approximating the entire feasible region with the surrogate, we only approximate individual complicating constraints, while keeping the non-complicating constraints in the formulation. To solve formulation (17), the complicating constraints $\tilde{f}_h$ and their respective derivatives will be replaced by algebraic surrogate models, as discussed next. In this manner, the structure of the original flexibility index problem is kept.

3.3. Incorporation of algebraic surrogate models for the complicating constraints

We include algebraic models substituting the complicating constraints to solve the flexibility formulation shown in Eqs. (16) or (17) using state-of-the-art deterministic solvers. To be able to use off-the-shelf optimization solvers, we follow the procedure described in Section 2 and graphically summarized in Fig. 1. The first step is the identification of complicating constraints. These constraints, described by $\tilde{f}_h$, $h \in H$, are then separated from the other non-complicating constraints as shown in Fig. 1. Once separated, we use a surrogate model approximation, F_h, as a simplification for the complicating constraints $\tilde{f}_h$. The original flexibility index problem in Eq. (17) is therefore reformulated into the hybrid expression (18) that combines the main backbone of the flexibility index problem with a data-driven surrogate model defined for the complicating constraints, as shown below.

$$
\begin{aligned}
FI \quad &= \min_{\delta} \delta \\
s.t. \quad & \hat{f}_g(d,z,\theta) - u + s_g = 0, && \forall g \in G \\
& F_h(d,z,\theta) - u + s_h = 0, && \forall h \in H \\
& \sum_g \lambda_g + \sum_h \lambda_h = 1 \\
& \sum_g \lambda_g \frac{\partial \hat{f}_g(d,z,\theta)}{\partial z_n} + \sum_h \lambda_h \frac{\partial F_h(d,z,\theta)}{\partial z_n} = 0, && \forall n \in N \\
& s_g \le M(1-y_g), && \forall g \in G \\
& s_h \le M(1-y_h), && \forall h \in H \\
& \lambda_g \le y_g, && \forall g \in G \\
& \lambda_h \le y_h, && \forall h \in H \\
& \sum_g y_g + \sum_h y_h \le |N| + 1 \\
& \theta \in T(\delta) \\
& \lambda_g \ge 0, \quad \lambda_h \ge 0, && \forall g \in G,\ h \in H \\
& s_g \ge 0, \quad s_h \ge 0, && \forall g \in G, h \in H \\
& \delta \ge 0
\end{aligned}
\tag{18}
$$

As visible in Eq. (18), the complicating constraint $\tilde{f}_h(d,z,\theta)$ was replaced by an adequate surrogate model $F_h(d,z,\theta)$. Other than that, expression (18) does not differ from expression (17).

3.4. Surrogate model building

This subsection explains the individual steps involved in the surrogate model generation in detail. The model building follows a similar procedure as described in a previous work by the authors (Forster et al., 2023), where we assume that a mapping of the uncertain parameters to the process response is possible. First, $\tilde{f}_h$ is evaluated at different points. Second, SR tools are applied to define a constraint F_h using a closed-form

algebraic surrogate model that fits the generated data points precisely (i.e., F_h approximates the given process constraint $\widetilde{f}_h$ accurately). Last, the performance of the obtained surrogate model is assessed by suitable metrics.

3.4.1. Step 1: data generation

A schematic overview of the data generation and model-building process is given in Fig. 2. We simulate the desired case study in Python by changing some independent variables (degrees of freedom) and observing the response of the dependent variables. To map these independent variables (also called the features of the model) to the observed response (also called the target of the model), we describe the feature vector $w_i = [z,\ \theta]$, where $i \in I$ refers to the set of samples. The feature vector ω_i consists of the control variables $z_n, n \in N$ and the uncertain parameters $\theta_k, k \in K$. The target vector is denoted as $\widetilde{f}_h(w_i)$, or $\widetilde{f}_{h,i}$ in short. Therefore, the sampling matrix is generated with the desired number of samples $|I|$ using, without generality loss, the Latin hypercube sampling (LHS).

The resulting dataset I is split into two proper subsets as shown in Eq. (19):

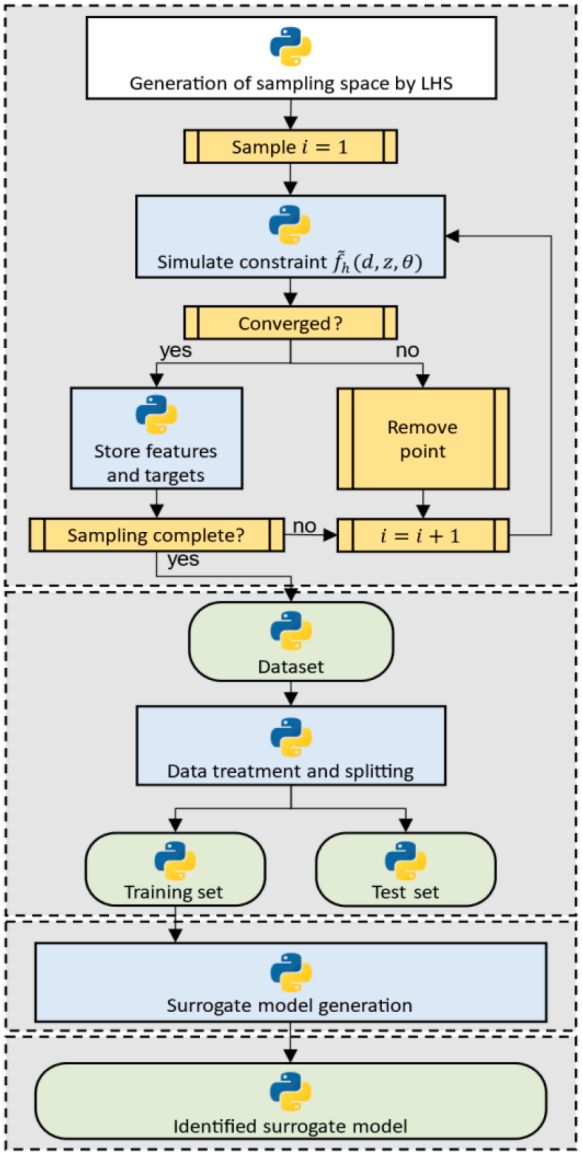

Fig. 2. Schematic representation of the data generation and surrogate model building procedure.

$$I := I^{TR} \cup I^{TE}$$
$$I^{TR} \cap I^{TE} = \varnothing \tag{19}$$

where, I^{TR} and I^{TE} represent the training and test subsets, respectively. The training subset is later used for model training, whereas the test subset is used for model testing.

3.4.2. Step 2: surrogate model building

After preprocessing the data, we proceed to find an expression in the form of a surrogate model $F_h(d, z, \theta)$ that accurately maps the above-described feature vector ω_i to the corresponding targets $\widetilde{f}_{h,i}$. Herein, since we apply an SR algorithm, we do not rely on any aprioristic assumption on the structure for $F_h(d,z,\theta)$. As mentioned, SR aims to find a suitable mathematical expression for the observed data by representing the appropriate expressions in a symbolic tree. An example of such a search is schematically shown in Fig. 3.

Fig. 3 (a) visualizes the space of all possible mathematical expressions γ, which is described by E. Starting from one symbolic tree representation $\gamma_e, e \in E$, we perform changes in the tree that lead to different mathematical expressions. One example of such a tree evolution is shown in Fig. 3 (node replacement). Another adaptation would be the elementary tree replacement (i.e., exchanging the complete sub-tree $(x_3 + x_4)$ by another sub-tree). Based on this tree evolution, a defined performance metric can be calculated for each resulting expression. This metric aims to quantify how well the expression fits the observed data. The SR algorithm then proceeds to search the space of expressions, seeking the expression with the best goodness of fit. This search is stochastic, as in other evolutionary algorithms (Costa and Oliveira, 2001; Guimerà et al., 2020).

As mentioned in the introduction, several SR algorithms are available to identify algebraic surrogates. Without loss of generality, we use the approach developed by Guimerà et al. (2020), the BMS, to simplify the complicating constraints $\widetilde{f}_h(d, z, \theta)$. The BMS uses statistical prior information about the mathematical operations in the equations, and it is straightforward to implement, working out-of-the-box and allowing interconnection with the Python environment without need of extensive coding. This easy implementation facilitates its application in different fields and case studies. Moreover, we note that the BMS was already successfully applied to build process models (Forster et al., 2023; Jog et al., 2023; Negri et al., 2022). The BMS can provide closed-form algebraic expressions from data based on a set of user-defined mathematical operations (i.e., addition, subtraction, multiplication, etc.). We next provide a high-level overview of how the BMS works. For further information, the reader is referred to the original paper (Guimerà et al., 2020).

A conditional probability $p(\gamma_e|D)$ is assigned to each expression γ_e that is used to fit some data D. This probability is calculated according to Bayes Theorem (Bishop, 2006; Murphy, 2013):

$$p(\gamma_e|D) = \frac{p(D|\gamma_e)\ p(\gamma_e)}{p(D)} \tag{20}$$

where, $p(D)$ represents the marginal likelihood of some data D. $p(D)$ is independent of γ_e and therefore only acts as a normalization constant. Marginalizing over the parameters ϕ_e associated with expression γ_e (Murphy, 2013), the numerator in Eq. (20) can be expressed as an integral over the space of all possible parameter values Φ_e (Guimerà et al., 2020). This marginalization is then described by the description length $\mathscr{DL}(\gamma_e)$ (Guimerà et al., 2020; Hansen and Yu, 2001; Murphy, 2013):

$$\begin{aligned} \mathscr{DL}(\gamma_e) &= -\log[p(D|\gamma_e)\ p(\gamma_e)] \\ &= -\log\left[\int_{\Phi_e} p(D|\gamma_e,\ \phi_e)\ p(\phi_e|\gamma_e)\ p(\gamma) d\phi\right] \end{aligned} \tag{21}$$

The computation of this integral is challenging (Guimerà et al., 2020;

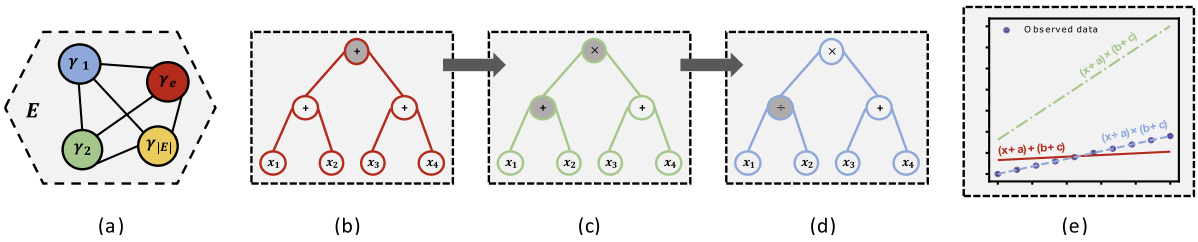

Fig. 3. (a) The space E of all possible expressions γ_e is schematically shown as a dashed polygon. (b) A representation of an initial mathematical expression $\gamma(x) = (x_1 + x_2) + (x_3 + x_4)$ as a symbolic tree (red). (c) A root node replacement is performed (grey node) to reach the green symbolic expression in (c). Performing another node replacement (grey node), the blue symbolic tree is reached (d), representing $\gamma(x) = (x_1 / x_2) \times (x_3 + x_4)$. The fitting visualization of the three expressions is shown in (e), together with the observed data as circles (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.).

Murphy, 2013). It was shown (Grünwald, 2007; Murphy, 2013) that under certain assumptions, the description length can be approximated through the Bayesian information criterion (BIC) and the prior of the corresponding symbolic expression γ_e:

$$\mathscr{DL}(\gamma_e) \approx \frac{BIC(\gamma_e)}{2} - \log(p(\gamma_e)) \tag{22}$$

The description length, and, therefore, this final equation can be interpreted as the plausibility of observing an expression γ_e, conditioned on some data D. According to Grünwald (2007), $\mathscr{DL}(\gamma_e)$ can also be understood as an encoded length of the expression γ_e (number of natural units).

In the applied SR approach (Guimerà et al., 2020), a Markov chain Monte Carlo (MCMC) (Hastings, 1970) algorithm is used to explore the space E of expressions, where the number of MCMC iterations is defined by the user. After evaluating the description length of each expression $\mathscr{DL}(\gamma_e)$, the BMS keeps the most plausible one, representing the expression with the shortest description length (the best goodness-of-fit).

3.5. Surrogate model performance

The performance of the surrogate model is assessed by calculating several metrics for both the training and test data sets, S^{TR} and S^{TE}. Here, to this end, the root mean squared error (*RMSE*), mean absolute error (*MAE*), and the coefficient of determination (R^2) were used:

$$\begin{aligned} RMSE &= \sqrt{\frac{1}{n}\sum_{a\in A}\left(\widetilde{f}_h(w_i) - F_h(w_i)\right)^2} \\ MAE &= \frac{1}{n}\sum_{a\in A}\left|\widetilde{f}_h(w_i) - F_h(w_i)\right| \\ R^2 &= 1 - \frac{SSR}{SST} = 1 - \frac{\sum_{a\in A}\left(F(w_i) - \widetilde{f}_h(w_i)\right)^2}{\sum_{a\in A}\left(\widetilde{f}_h(w_i) - \mu_{\widetilde{f}_h}\right)^2} \end{aligned} \tag{23}$$

In Eq. (23), the predictions by the model are described by $F_h(\boldsymbol{w}_i)$ using the given input vector $\boldsymbol{w}_i$ of one sample i. The observed response $\widetilde{f}_h$ and the mean of the observed process responses are described by $\widetilde{f}_h(\boldsymbol{w}_i)$ and $\mu_{\widetilde{f}_h}$, respectively. As already mentioned, both the model predictions $F_h(\boldsymbol{w}_i)$ and the observed response $\widetilde{f}_h(\boldsymbol{w}_i)$ are calculated by using input data from the training or test set. Variables *SSR* and *SST* denote the sum of squares of residuals and the total sum of squares (proportional to the variance of the data), respectively. In addition to these performance metrics, the time required for both the model training and for solving the flexibility index problem is reported as a central processing unit (CPU) time. Lastly, both the solver and model status are reported.

3.6. Software implementation

All calculations were carried out on an Intel®Core™ i7-8700 CPU and 16 GB of RAM. We used Python v3.10 with NumPy v1.23.5, SciPy v1.9.3, and pyDOE v0.3.8 to construct the sampling dataset. The algorithm provided by Guimerà et al. (2020) was used to train the BMS. The symbolic equation generated by the BMS was incorporated into the flexibility index problem, which was solved using Pyomo (Bynum et al., 2021; Hart et al., 2011) v6.4.4 interfacing with the solver BARON (Sahinidis, 1996) v22.7.23.

4. Case studies

We apply the hybrid flexibility approach discussed above to two case studies (CS). The first covers a protein-A chromatographic process; the second is a bioprocess in fed-batch operation mode. The CS and corresponding data generation processes are described in the following.

4.1. Fed-batch bioreactor for ethanol production (CS-I)

We consider a bioreactor in a fed-batch operation mode. The model was taken from the dynamic optimization examples demonstrated on APmonitor.com (Hedengren et al., 2014). A schematic representation of the reactor is given in Fig. 4. The reactor is equipped with a liquid feed, an air supply (with a submerged aerator), a heating/cooling jacket, and a temperature probe inside the reactor.

In the reactor, microorganisms grow and produce ethanol by consuming oxygen and glucose. To describe the dynamic evolution of the species, the system of ODEs given in Table 1 is used together with the corresponding parameters indicated in Table 2.

One major goal is to that the final ethanol concentration reaches at least a user-defined lower bound $\underline{E}$. The control variable here is the temperature of the cooling agent, that is, $z = T_c$. Furthermore, the uncertain parameters θ are the glucose concentration in the feed (S_{in}) and the temperature within the reactor (T). The constraints of this problem can therefore be formulated as given in Eq. (41).

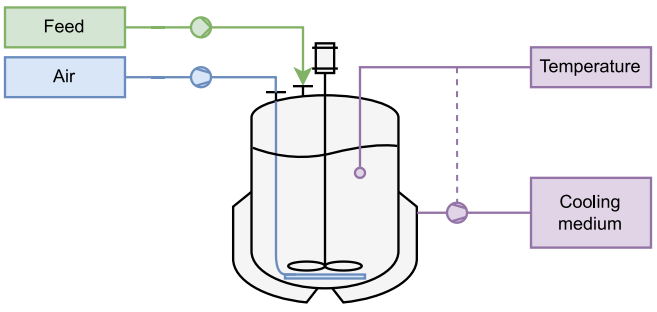

Fig. 4. Schematic representation of a bioreactor used in case study I.

Table 1
System of ordinary differential equations used for simulating the bioreactor discussed in the case study I. The corresponding parameters are shown in Table 2.

Physical meaning	Equation	
Specific growth rate	$\mu = \mu_{max}\frac{S}{K_{SX}+S}\frac{O_{liq}}{K_{OX}+O_{liq}}\left(1-\frac{E}{Emax}\right)\frac{1}{1+\exp(-(100-S))}$ $\mu_{max} = [(a_1(T-k_1))(1-\exp(b_1(T-k_2)))]^2$ $E_{max} = E_{maxb}+\frac{E_{maxT}}{1-\exp(-b_2(T-k_3))}$ $q_E = a_E\mu + b_E$	(24)
Non-growth ethanol products	$b_E = c_1\exp\left(-\frac{A_{P1}}{T}\right) - c_2\exp\left(-\frac{A_{P2}}{T}\right)$	(25)
Ethanol consumption	$q_S = \frac{\mu}{Y_{XS}} + \frac{q_E}{Y_{ES}}$	(26)
Oxygen consumption	$q_O = \frac{q_{O,max}}{Y_{XO}}\frac{O_{liq}}{K_{OX}+O_{liq}}$	(27)
Biomass deactivation	$K_d = K_{db}+\frac{K_{dT}}{1+\exp(-b_3(T-k_4))}$	(28)
Oxygen saturation	$O_{sat} = Z\frac{O_{gas}RT}{K_H}$	(29)
Oxygen mass transfer	$k_la = (k_la)_0(1.2)^{T-20}$	(30)
Total volumes	$V = V_l + V_g$	(31)
Liquid volume	$\frac{dV_l}{dt} = Q$	(32)
Total biomass	$\frac{dX_t}{dt} = \mu X_v + \frac{Q}{V_l}(X_{t,in}-X_t)$	(33)
Viable biomass	$\frac{dX_v}{dt} = (\mu - K_d)X_v + \frac{Q}{V_l}(X_{v,in}-X_v)$	(34)
Glucose	$\frac{dS}{dt} = \frac{Q}{V_l}(S_{in}-S) - q_SX_v$	(35)
Ethanol	$\frac{dE}{dt} = \frac{Q}{V_l}(E_{in}-E) + q_EX_v$	(36)
Liquid oxygen	$\frac{dO_{liq}}{dt} = \frac{Q}{V_l}(O_{sat}-O_{liq}) + k_la(O_{sat}-O_{liq}) - q_OX_v$	(37)
Gas oxygen	$\frac{dO_{gas}}{dt} = \frac{F_{air}}{V_g}(O_{gas,in}-O_{gas}) - \frac{V_lk_la}{V_g}(O_{sat}-O_{liq}) + \frac{O_{gas}Q}{V_g}$	(38)
Temperature	$\frac{dT}{dt} = \frac{Q}{V_l(T_{in}-T)} - \frac{T_{ref}}{V_l}Q + q_OX_v\frac{\Delta H}{MW_O\rho C_{p,br}} - \frac{K_TA_T(T-T_c)}{V_l\rho C_{p,br}}$	(39)
Cooling agent	$\frac{dT_c}{dt} = \frac{F_c}{V_{cj}}(T_{c,in}-T_c) + \frac{K_TA_T(T-T_c)}{V_{cj}\rho_cC_{p,c}}$	(40)

$$
\begin{aligned}
&f_1 : \underline{E} - E \le 0\\
&f_2 : \ \underline{T_c} - T_c \le 0\\
&f_3 : \ T_c - \overline{T_c} \le 0\\
&f_4 : \underline{S_{in}} - S_{in} \le 0\\
&f_5 : S_{in} - \overline{S_{in}} \le 0\\
&f_6 : \underline{T} - T \le 0\\
&f_7 : T - \overline{T} \le 0\\
&J := \{1,2,3,4,5,6,7\}
\end{aligned}
\tag{41}
$$

The ethanol concentration needs to be assessed, which is not straightforward. We add the first constraint to the set of complicating constraints $H = \{1\}$, namely, we define $\widetilde{f}_1 = f_1 = \underline{E} - E$. The other constraints are added to the set of non-complicating constraints $G = \{2, 3, 4, 5, 6, 7\}$. As mentioned above, this could for example be done by discretizing the differential equations appropriately (i.e., by applying orthogonal collocation on finite elements). However, one disadvantage is that the dimensionality of the resulting optimization problem would be very large due to the addition of many auxiliary variables (Carey and Finlayson, 1975; Guillén-Gosálbez et al., 2013). To circumvent such possible limitations, the ethanol concentration at the reactor outlet shall be modeled with the BMS. Therefore, $\mathscr{T}(S_{in}, T, T_c)$ represents a trained BMS model that maps the features S_{in}, T, and T_c to the final ethanol concentration E in the reactor. Hence, the constraint $\widetilde{f}_1 = \underline{E} - E$ is reformulated by using a closed-form algebraic expression, leading to $F_1 = \underline{E} - \mathscr{T}(S_{in}, T, T_c)$. The formulation in Eq. (42) then provides the entire reformulated problem. It is worth mentioning again that $\widetilde{f}_1$ describes the original complicating constraint, whereas F_1 describes the reformulated complicating constraint where a surrogate equation is included to facilitate the calculations.

Table 2
Parameters used in the ordinary differential equation system given in Table 1.

Parameter	Physical meaning	Value	Unit
a_1	Ratkowsky parameter	0.05	$^\circ C^{-1}\ h^{-0.5}$
a_E	Growth-associated parameter for ethanol production	4.5	–
A_{P1}	Activation energy parameter for ethanol production 1	6	°C
A_{P2}	Activation energy parameter for ethanol production 2	20.3	°C
b_1	Exponential scaling parameter for the maximum specific growth rate	0.035	$^\circ C^{-1}$
b_2	Exponential scaling parameter for the growth inhibitory ethanol concentration	0.15	$^\circ C^{-1}$
b_3	Exponential scaling parameter for the specific death rate	0.4	$^\circ C^{-1}$
c_1	Constant decoupling factor for ethanol production	0.38	$g_E\ g_X^{-1}\ h^{-1}$
c_2	Constant decoupling factor for ethanol production	0.29	$g_E\ g_X^{-1}\ h^{-1}$
k_1	Parameter in the maximum specific growth rate	3	°C
k_2	Parameter in the maximum specific growth rate	55	°C
k_3	Parameter in the growth-inhibitory ethanol concentration expression	60	°C
k_4	Temperature at the inflection point of the specific death rate sigmoid curve	50	°C
E_{maxb}	Temperature-independent product inhibition constant	90	$g\ L^{-1}$
E_{maxT}	Maximum value of product inhibition constant due to temperature	90	$g\ L^{-1}$
K_{db}	Basal specific cellular biomass death rate	0.025	h^{-1}
K_{dT}	Maximum value of specific cellular biomass death rate due to temperature	30	h^{-1}
K_{SX}	Glucose saturation constant for the specific growth rate	5	$g\ L^{-1}$
K_{OX}	Oxygen saturation constant for the specific growth rate	0.0005	$g\ L^{-1}$
$q_{O,max}$	Maximum specific oxygen consumption rate	0.05	h^{-1}
Y_{ES}	Theoretical yield of ethanol on glucose	0.51	$g_E\ g_S^{-1}$
Y_{XO}	Theoretical yield of biomass on oxygen	0.97	$g_X\ g_O^{-1}$
Y_{XS}	Theoretical yield of biomass on glucose	0.53	$g_X\ g_S^{-1}$
$C_{p,br}$	Heat capacity of the mass of reaction	4.18	$J\ g^{-1}{}^\circ C^{-1}$
$C_{p,c}$	Heat capacity of the cooling agent	4.18	$J\ g^{-1}{}^\circ C^{-1}$
ΔH	Heat of reaction of fermentation	518,000	$J\ mol_O^{-1}$
T_{ref}	Reference temperature	20	°C
K_H	Henry's constant for oxygen in the fermentation broth	200	$atm\ L\ mol^{-1}$
Z	Oxygen compressibility factor	0.792	-
R	Ideas gas constant	0.082	$L\ atm\ mol^{-1}{}^\circ C^{-1}$
$(k_la)_0$	Temperature-independent volumetric oxygen transfer coefficient	100	h^{-1}
K_T	Heat transfer coefficient	360,000	$J\ h^{-1}\ m^{-2}{}^\circ C^{-1}$
ρ	Density of the fermentation broth	1080	$g\ L^{-1}$
ρ_c	Density of the cooling agent	1000	$g\ L^{-1}$
MW	Molecular weight of oxygen	32	$g\ mol^{-1}$

$$
\begin{aligned}
&F_1 : \underline{E} - \mathscr{T}(S_{in}, T, T_c) \le 0\\
&f_2 : \ \underline{T_c} - T_c \le 0\\
&f_3 : \ T_c - \overline{T_c} \le 0\\
&f_4 : \underline{S_{in}} - S_{in} \le 0\\
&f_5 : S_{in} - \overline{S_{in}} \le 0\\
&f_6 : \underline{T} - T \le 0\\
&f_7 : T - \overline{T} \le 0\\
&H := \{1\},\ G := \{2,3,4,5,6,7\}
\end{aligned}
\tag{42}
$$

The goal of the flexibility analysis is to quantify and identify the largest possible uncertainty set $\theta \in T(\delta)$, such that the process is still feasible over the entire range of θ. In other words, one should assess how far the glucose inlet concentration and the reactor's temperature can deviate from the nominal operating point such that the process is still feasible (all the constraints still hold).

To find $\mathscr{T}$, the ODE system given in Table 1 was solved for different feature vectors $\omega_i = [S_{in}, T, T_c], i \in I$ with $|I| = 250$ samples using the explicit Runge–Kutta method of order 5 (Dormand and Prince, 1980). After simulating for each ω_i, the final ethanol concentration was obtained. The sampling procedure discussed above (Fig. 2) was applied, where the upper and lower bounds selected for the LHS are displayed in Table 3. The resulting dataset A was randomly split to $|I^{TR}| = 200$ training (80 %) and $|I^{TE}| = 50$ testing (20 %) samples.

To train the BMS, several unary ($\exp(x), \log(x), x^2, x^3, \sqrt{x}$) and binary ($+, -, \div, \times, x^y$) operators were allowed to be selected. In addition, the number of MCMC steps was fixed to 15×10^3. The model was allowed to contain up to eight parameters.

4.2. Protein-A affinity chromatography (CS-II)

This case study consists of a loading process of antibodies onto a protein-A affinity chromatographic column. A schematic representation of the different steps in chromatography is given in Fig. 5. First, the column is packed with the desired material (resin). Second, an equilibration is performed, which makes the column ready to be deployed. During the loading phase, the antibody mixture is added to the top of the column. Depending on the loading time (t_{load}), the antibody concentration in the feed (c_{in}), and the flowrate (Q), some of the product might be lost. Subsequently, the washing step is used to collect the desired product. The elution step terminates the entire operation.

We focus exclusively on the loading phase of the entire procedure. The loss ratio (LR) is the relationship between the mass of the leaked product relative to the total amount of protein fed. With this, the deterministic constraints of the problem can be formulated as given in Eqs. (43)–(53), which was adapted from Ref. Baur et al. (2016), where the corresponding parameters were taken from the same work (Baur et al., 2016; Ding and Ierapetritou, 2021).

$$\frac{\partial c}{\partial t} = -\frac{Q}{A_{col}\epsilon}\frac{\partial c}{\partial x} + D^{app}\left(\frac{\partial^2 c}{\partial x^2}\right) - \zeta\frac{\partial q}{\partial t} \tag{43}$$

$$D^{app} = \widehat{V}\left(\frac{d_p}{2}\right)\frac{Q}{A_{col}\epsilon} \tag{44}$$

$$\frac{\partial q}{\partial t} = k_m(q^* - q) \tag{45}$$

$$q^* = \frac{Hc}{1 + \frac{Hc}{q_{sat}}} \tag{46}$$

$$k_m = k_{max}\left(C_1 + (1 - C_1)\left(1 - \frac{q}{q_{sat}}\right)_2^C\right) \tag{47}$$

$$\left[\frac{\partial c}{\partial x}\right]_{x=L} = 0 \text{ and } \left[\frac{\partial q}{\partial x}\right]_{x=L} = 0 \tag{48}$$

Table 3
Upper and lower bounds for the features S_{in}, T, and T_c. The bounds were used to create the samples for case study I by applying a Latin hypercube sampling structure.

Feature	Lower bound	Upper bound	Unit
S_{in}	0	20	g L^{-1}
T	15	35	°C
T_c	20	40	°C

$$c(t=0) = c_0 \text{ and } q(t=0) = q_0 \tag{49}$$

$$LR = \frac{\int_0^{t_{load}} c(x,t)dt}{\int_0^{t_{load}} c_{in}dt} \leq \overline{LR} \tag{50}$$

$$\underline{Q} \leq Q \leq \overline{Q} \tag{51}$$

$$\underline{c_{in}} \leq c_{in} \leq \overline{c_{in}} \tag{52}$$

$$\underline{t_{load}} \leq t_{load} \leq \overline{t_{load}} \tag{53}$$

e:inline-figure))"^?^(1)[?tal=1]> The system given in Eqs. (43)–(49) describes the partial differential equations (PDE) for the dynamic evolution of the concentration profiles, which can be expressed in terms of concentration in the liquid phase (c) and in the adsorbed phase (q). The parameters of the PDE system are given in Table 4.

Q, c_{in}, and t_{load} are the adjustable flow rate, the inlet antibody concentration, and the loading time, respectively. Their lower and upper bounds are indicated by $\underline{Q}$, $\underline{c_{in}}$, $\underline{t_{load}}$, $\overline{Q}$, $\overline{c_{in}}$, and $\overline{t_{load}}$, respectively, which are represented in Eqs. (51)–(53). LR represents the loss rate, which is the relationship between the mass of leaked product relative to the total amount of product fed during the loading phase. $\overline{LR}$ is a user-defined upper bound for the loss rate. The entire system can be rewritten more compactly, as shown in Eq. (54).

$$\begin{aligned} &f_1: LR - \overline{LR} \leq 0 \\ &f_2: \underline{Q} - Q \leq 0 \\ &f_3: Q - \overline{Q} \leq 0 \\ &f_4: \underline{c_{in}} - c_{in} \leq 0 \\ &f_5: c_{in} - \overline{c_{in}} \leq 0 \\ &f_6: \underline{t_{load}} - t_{load} \leq 0 \\ &f_7: t_{load} - \overline{t_{load}} \leq 0 \\ &J := \{1,2,3,4,5,6,7\} \end{aligned} \tag{54}$$

This entire differential system in Eqs. (43)–(49) and the integrals in Eq. (50) are not trivial – and computationally expensive – to be incorporated into the optimization problem. Again, one option would be to apply an appropriate discretization method to the differential equations, which would increase the problem dimensionality and potentially lead to convergence issues, as discussed in CS-I. We, therefore, add the first constraint to the set of complicating constraints $H = \{1\}$, describing it by $\widetilde{f}_1 = f_1 = LR - \overline{LR}$. The other constraints are added to the set of non-complicating constraints $G = \{2,3,4,5,6,7\}$. A BMS model is used that maps the features c_{in}, t_{load}, and Q to LR. Hence, the constraint $\widetilde{f}_1 = LR - \overline{LR}$ is reformulated by using a closed-form algebraic expression, leading to $F_1 = \mathscr{T}(c_{in}, t_{load}, Q) - \overline{LR}$. The entire reformulated constraints are then given by the formulations shown in Eq. (55). Again, it is worth mentioning again that $\widetilde{f}_1$ describes the original complicating constraint, whereas F_1 describes the reformulated complicating constraint including the algebraic surrogate equation.

$$\begin{aligned} &F_1: \mathscr{T}(c_{in}, t_{load}, Q) - \overline{LR} \leq 0 \\ &f_2: \underline{Q} - Q \leq 0 \\ &f_3: Q - \overline{Q} \leq 0 \\ &f_4: \underline{c_{in}} - c_{in} \leq 0 \\ &f_5: c_{in} - \overline{c_{in}} \leq 0 \\ &f_6: \underline{t_{load}} - t_{load} \leq 0 \\ &f_7: t_{load} - \overline{t_{load}} \leq 0 \\ &H := \{1\},\ G := \{2,3,4,5,6,7\} \end{aligned} \tag{55}$$

Here, the flexibility analysis aims to assess how far the inlet concentration of the antibody and the loading time of the column can deviate from the nominal operating point such that the process is still feasible (all the constraints still hold).

To find a suitable model for $\mathscr{T}$, the PDE system given in Eqs. (43)–(49) was solved for several samples ($|I| = 250$ samples) of the feature vector $\omega_i = [c_{in,a}, t_{load,a}, Q_a], i \in I$. For each run, a spatial discretization

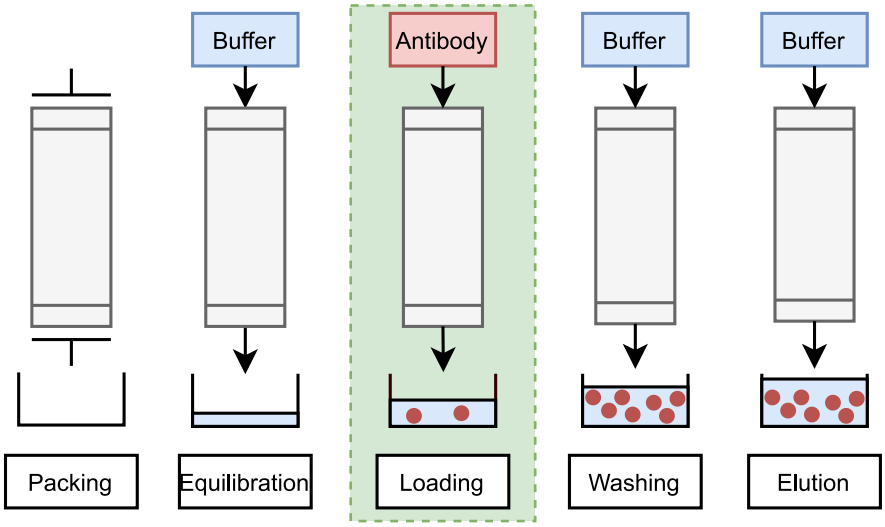

Fig. 5. Schematic representation of the five different steps in a chromatographic procedure. The loading phase (marked by the dashed green area) is the step of interest for this case study (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.).

Table 4
Parameters used for the chromatography model discussed in case study II. The corresponding model equations shown in Eqs. (43)-(49) were adapted from Baur et al. (2016). The parameters were taken from Baur et al. (2016) and Ding and Ierapetritou (2021).

Parameter	Physical meaning	Value	Unit
L_{col}	Column length	10	cm
A_{col}	Crossectional area of the column	0.2	cm^2
d_p	Average particle diameter	0.0044	cm
ϵ	Void fraction	0.368	–
$\widehat{V}$	Intercept of reduced Van-Deemter equation	35.13	–
H	Partition coefficient	246.8	–
q_{sat}	Saturation concentration in the adsorbed phase	94.72	$mg\ mL^{-1}$
k_{max}	Maximum mass transfer rate	0.18	min^{-1}
C_1	Pore blockage coefficient 1	0.6245	–
C_2	Pore blockage coefficient 2	2.071	–
c_0 and q_0	Initial values of the liquid and adsorbed phases	0	$mg\ mL^{-1}$

along the column length with 100 grid points was performed using a first-order central finite differences method. Subsequently, the resulting system of ordinary differential equations (ODE) was solved at each spatially discretized point using the explicit Runge–Kutta method of order 5 (Dormand and Prince, 1980). After simulating for each ω_i, the concentration profile was obtained integrating the expression in Eq. (50), and therefore a value for LR, could be numerically calculated. The number of spatial discretization points was fixed at 100. The sampling procedure discussed above (Fig. 2) was applied, where upper and lower bounds for the LHS are displayed in Table 5. The resulting dataset A was randomly split to $|I^{TR}| = 200$ training (80 %) and $|I^{TE}| = 50$ testing (20 %) samples.

To train the BMS, several unary $(\exp(x), \log(x),\ x^2, x^3, \sqrt{x})$ and binary $(+,\ -,\ \div,\ \times, x^y)$ operators were allowed to be selected. In addition, the number of MCMC steps was fixed to 20×10^3. The model was allowed to contain up to three parameters.

Table 5
Upper and lower bounds for the features c_{in}, t_{load}, and Q. The bounds were used to create the samples for case study II by applying a Latin hypercube sampling structure.

Feature	Lower bound	Upper bound	Unit
c_{in}	0.5	2.2	mg/mL
t_{load}	1/60	20	min
Q	0.001	20	mL/min

5. Results and discussion

5.1. Surrogate model generation

The results of the surrogate model training and testing for CS-I and CS-II are given in Table 6. In addition, visualizations of the model performances are shown in Fig. 6, where predicted values are plotted against observed ones. The corresponding closed-form expressions with the highest plausibility (lowest description length), and their estimated parameters are shown in Tables 7 and 8.

In general, both trained models can explain the variance in the data sufficiently well when considering R^2 values greater than 0.9 as acceptance criterion based on earlier works (Forster et al., 2023). The BMS was run using the maximum number of MCMC iterations as the stopping criterion, as indicated in Section 4. This led to CPU times of 0.8 h for CS-I and 2.7 h for CS-II. The low discrepancy between the R^2 values of the training and testing results indicates that the BMS is well-regularized and, therefore, less prone to overfitting, which is in line with the authors' expectations (Guimerà et al., 2020).

Table 6
The training performance criteria are summarized for the Bayesian machine scientist (BMS). Each row represents one case study (CS). The CPU time (in hours) needed for the model training is shown in the left part of the table. The error metrics (root mean squared error, mean absolute error, coefficient of determination) are shown for the training and testing data (format: training/testing). The error units are given in squared brackets. The identified algebraic expressions are indicated in Table 7, whereas the corresponding model parameters are reported in Table 8.

CS	CPU training	RMSE	MAE	R2
I	0.8 h	0.467 / 1.811 [g/L]	0.383 / 0.656 [g/L]	0.996 / 0.913 [-]
II	2.7 h	0.014 / 0.012 [-]	0.009 / 0.008 [-]	0.998 / 0.998 [-]

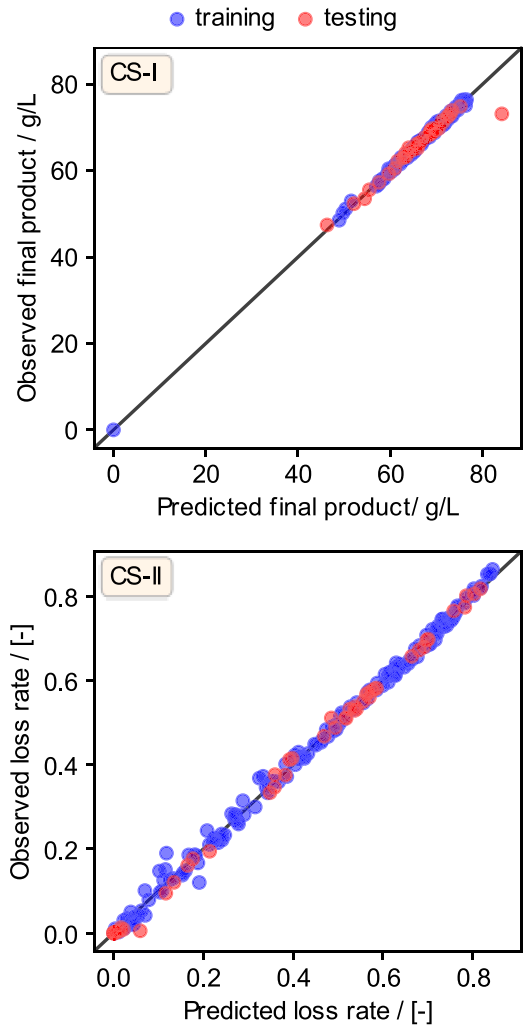

Fig. 6. Observed vs. predicted (OVP) values for the two different case studies are shown. Blue points represent the training data, whereas red points correspond to the test data. The black line represents the values where the observed value corresponds to the model predictions (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.).

In addition to the previous performance criterion ($R^2 > 0.9$), Fig. 6 shows that the surrogate models perform satisfactorily both in the training and test sets, where the model responses are very close to the outcome of the theoretical models. However, what can be observed for CS-II is that the risk of over or underprediction increases for low values of LR (higher spread of the training and testing points for values below around $LR = 0.5$). For CS-I, one can find most data points between 45 g/L and 80 g/L, where only one training sample was at 0 g/L. This point resulted from the LHS sampling and was not removed for training the BMS.

Regarding the surrogate models in Table 7, the BMS identified nonlinear expressions with all variables included as features. We recall that the model training considers the control variables and the uncertain parameters as features (inputs for the surrogates). This is required to adjust the control variables depending on the realization of the uncertain parameters, as done in the flexibility index problem.

The identified surrogate expressions were then incorporated into the hybrid formulation given in Eq. (18), as already discussed.

Table 7
The most plausible closed-form expressions for each case study (CS) identified by the Bayesian machine scientist (BMS) are shown. The corresponding estimated parameter values are reported in Table 8. The variable descriptions for each case study are given in Section 4.

CS	Prediction target	Identified expression
I	$E = E(T_c, S_{in}, T)$ $z = [T_c]$ $\theta = [S_{in}, T]$	$a_1 + \frac{a_4 + T_c}{T_c + a_7}\frac{a_3}{T}(S_{in} + a_1) + a_7 T_c^2 \frac{a_0}{(-S_{in} + a_4 a_5)a_1}$
II	$LR = LR(c_{in}, Q, t_{load})$ $z = [Q]$ $\theta = [c_{in}, t_{load}]$	$a_0 \left(\frac{t_{load}}{\frac{t_{load}/c_{in}}{\left(a_0 \frac{}{a_1}\right)} t_{load} + \frac{a_1}{\exp(c_{in}^{a_0})}} \right)^{-\left(\frac{a_2 (a_0 Q)^{a_2}}{a_0}\right)\left(\frac{Q}{c_{in}} + a_1\right)}$

Table 8
Parameter values of the most plausible surrogate model identified by the Bayesian machine scientist (BMS) for each case study (CS). The corresponding model equations are given in Table 7.

Parameter	CS	
	I	II
a_0	1.411	0.894
a_1	66.686	24.458
a_2	1.000	-1.844
a_3	4.123	–
a_4	-17.508	–
a_5	-1.922	–
a_6	1.000	–
a_7	-17.503	–

5.2. Incorporation of surrogate models in the flexibility index problem

The results of the case studies CS-I and CS-II are summarized in Table 9. Schematic representations of these solutions are given in Fig. 8.

Table 9 shows that in both case studies, the optimal control variable z^* will be chosen at one of the bounds ($z^* = 23.0$ °C for CS-I and $z^* = 4.0$ mL/min for CS-II). Additionally, the first constraint F_1 was active in both cases. These are the constraints that were modeled using the BMS surrogates. Active surrogate constraints were expected, since the control variable influences those constraints. In other words, the optimizer tries to maximize the distance from the nominal operating point to a constraint. The F_1 constraints (surrogates) are influenced by the control variable. The optimizer adapts the control variable to shift the surrogate constraint away from the nominal operating point. This is done until the control variable cannot be adjusted anymore when it reaches its bound. In the chosen scenarios, the control variable impacts only the surrogate constraints with the relationship $F_{1,CS-I} \propto \mathscr{T}(S_{in}, T, T_c)$ in CS-I and $F_{1,CS-II} \propto \mathscr{T}(c_{in}, t_{load}, Q)$ in CS-II. A visualization of how the control variable influences the surrogate constraints is schematically given in Fig. 7.

The resulting flexibility index δ^* can for example be used to compare two process designs in order to elucidate which one is more flexible. For example, a comparison of two different process designs for CS-II is

Table 9
Results summary of the case studies CS-I and CS-II.

	CS-I	**CS-II**
θ^N	[10.0 g/L, 30.0 °C]	[1.5 g/L, 8.0 min]
M	20	500
$\underline{z}$ and $\overline{z}$	23.0 °C and 28 °C	4.0 mL/min and 12.0 mL/min
$\Delta\theta_k^{min},\ k \in K$	[1 g/L, 1 °C]	$\theta_k^N - \underline{\theta_k}$
$\Delta\theta_k^{max},\ k \in K$	[1 g/L, 1 °C]	$\overline{\theta_k} - \theta_k^N$
δ^*	3.228	0.811
θ^*	[13.23 g/L, 33.23 °C]	[2.07 g/L, 13.67 min]
z^*	23.0 °C	4.0 mL/min
Active constraints	F_1, f_2	$F_1,\ f_4$
CPU	0.9 s	1.5 s

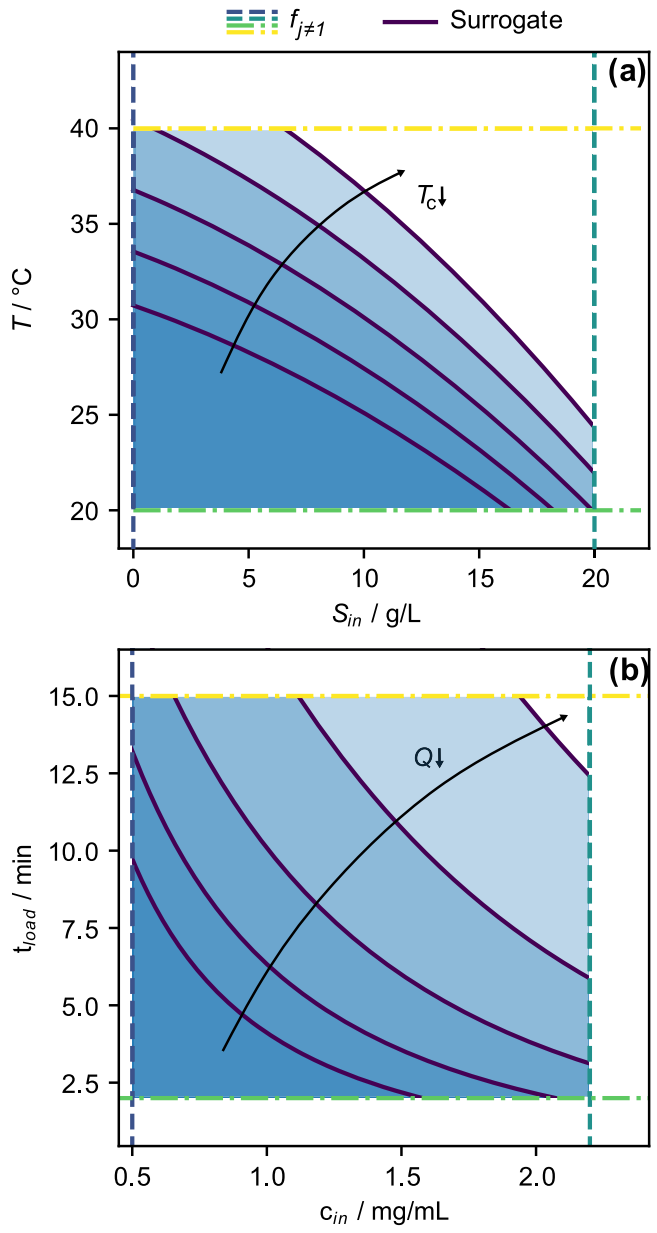

Fig. 7. Projection of the constraints onto the uncertain parameter plane for case studies CS-I (a) and CS-II (b). The feasible region is shown in shaded light blue color. The constraints in dashed lines represent the bounds of the un certain parameters. The solid lines represent the surrogate constraint which can be influenced by the control variable z. Decreasing the value of z increases the size of the feasible region (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.).

shown in the supplementary information Section S1. By using a longer and narrower column (design d_2) compared to the one given in Section 4.2 (design d_1), the flexibility is reduced ($\delta^*_{d_1} = 0.811$ vs. $\delta^*_{d_2} = 0.389$). The result is visualized in Fig. S1. Although such visualizations as in Fig. 7 cannot be done for higher dimensional case studies, the entire procedure can be applied in the same manner.

For both case studies, decreasing the control variable – the cooling temperature in Fig. 7 (a) and the flow rate in Fig. 7(b) – will increase the size of the feasible region. Considering for example CS-II, increasing the flow rate would decrease the time the antibodies would require to reach the column outlet. Therefore, a larger amount of product will be lost, which increases the loss rate during the loading phase. Keeping this fact in mind, one can observe that for lower flow rates, a higher loading time and higher antibody concentration would be possible, meaning these uncertain parameters (t_{load} and c_{in}) can deviate more from a nominal operating point, making the process feasible. This manifests in the larger feasible region given in Fig. 7(b). Similar behavior can be observed for CS-I. The surrogate model predicts a higher ethanol production with a decreased jacket temperature T_c. Therefore, the deviation on the reactor temperature and the feed concentration can be larger such that the process remains feasible, which again manifests in the higher feasible region visible in Fig. 7(a).

Fig. 8 visualizes the results given in Table 9, where the surrogate and

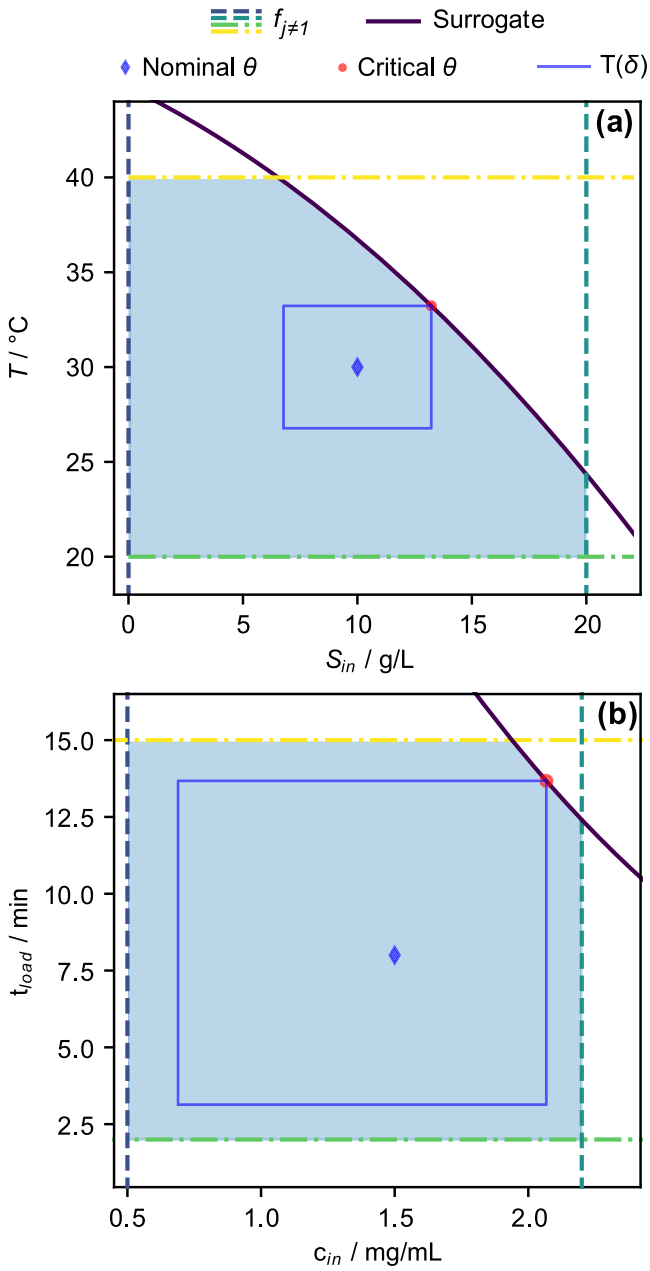

Fig. 8. Graphical representation of the solution for the flexibility index problem for CS-I (a) and CS-II (b). The feasible region is shown in shaded light blue color. The constraints in dashed lines represent the bounds of the uncertain parameters. The solid lines represent the surrogate constraint which can be influenced by the control variable z. The chosen nominal operating point $\boldsymbol{\theta}^N$ (blue diamond) lies within the set $\boldsymbol{T}(\boldsymbol{\delta})$ (blue box). As shown in Table 9, the surrogate constraints $\boldsymbol{F}_1$ are the active constraints, which is why $\boldsymbol{T}(\boldsymbol{\delta})$ touches $\boldsymbol{F}_1$ constraint (red circle) (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.).

control constraints are active. Having chosen a nominal operating point θ^N, the optimal value of theta in the optimum is called critical theta θ^c (red circles in Fig. 8), which indicates the scaled distance at which the process will hit the first bound. In other words, going beyond the set of parameters values θ^c ([13.23 g/L, 33.23 °C] for CS-I and [2.07 g/L, 13.67 min] for CS-II), will lead to the violation of the surrogate constraint, resulting in an infeasible process.

The flexibility problems were solved quickly, namely, in 0.9 s and 1.5 s, for CS-I and CS-II, respectively. Another advantage of having the algebraic surrogate becomes evident when the entire problem must be modified for any reason. For example, when the nominal operating point has to be changed, no re-training of the surrogate model is required since the training of the surrogates of the complicating constraints is decoupled from the flexibility index problem. This makes the adjustment of nominal operating points or bounds very simple because the solution time of the optimization problem is within seconds.

6. Conclusions

This work introduced a new approach to compute the flexibility index in problems with complicating constraints. Our approach combines the originally described deterministic formulation of the flexibility index problem with a symbolically regressed surrogate model that simplifies the modelling of the complicating constraints. The symbolic regression algorithm, the BMS, assumes no aprioristic model structure, thereby enabling the accurate representation of process constraints hard to model and/or handle numerically. The resulting hybrid flexibility approach was applied to protein-A chromatography and an ethanol production process in fed-batch operation mode. The surrogate equations could accurately reproduce the complicating constraints, as evidenced by their ability to explain the data variance, making them suitable for simplifying such equations in the deterministic flexibility formulation. One drawback of the applied regression tool is the significant training time required for model building, which might be improved in the future as faster SR algorithms become available.

Nevertheless, having a closed-form expression at hand pays off in several aspects: The first is that global solvers can be used, which can guarantee global optimality compared to heuristics or stochastic solvers. Additionally, the surrogate model training is decoupled from the flexibility index problem. This makes the study of different process conditions very simple because the solution time of the optimization problem is often within seconds using existing approaches to compute the flexibility index of fully analytical process models. However, we stress that our method focuses on the traditional flexibility index, so more complex flexibility metrics would require alternative methods. In the end, the most suitable approach for a given flexibility problem will depend on its features and the goal and scope of the analysis. Future work will focus on exploring alternative symbolic regression algorithms and a wider range of applications within chemical engineering and beyond.

CRediT authorship contribution statement

Tim Forster: Methodology, Software, Validation, Formal analysis, Data curation, Writing – original draft, Writing – review & editing, Visualization. **Daniel Vázquez:** Conceptualization, Methodology, Writing – review & editing. **Isabela Fons Moreno-Palancas:** Conceptualization, Methodology. **Gonzalo Guillén-Gosálbez:** Conceptualization, Methodology, Writing – review & editing, Supervision, Project administration.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgment

The authors would like to acknowledge support from the Swiss National Science Foundation (Project LEARN-D, MINT 200021_214877).

Supplementary materials

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.compchemeng.2024.108630.

References

Banerjee, I., Ierapetritou, M.G., 2005. Feasibility evaluation of nonconvex systems using shape reconstruction techniques. Ind. Eng. Chem. Res. 44, 3638–3647. https://doi.org/10.1021/ie049294d.

Baur, D., Angarita, M., Müller-Späth, T., Morbidelli, M., 2016. Optimal model-based design of the twin-column CaptureSMB process improves capacity utilization and productivity in protein A affinity capture. Biotechnol. J. 11, 135–145. https://doi.org/10.1002/biot.201500223.

Ben-Tal, A., Nemirovski, A., 2002. Robust optimization – methodology and applications. Math. Program. 92, 453–480. https://doi.org/10.1007/s101070100286.

Ben-Tal, A., El Ghaoui, L., Nemirovskiĭ, A.S., 2009. Robust optimization, Princeton series in Applied Mathematics. Princeton University Press. Princeton.

Birge, J.R., Louveaux, F., 2011. Introduction to stochastic programming. Springer Series in Operations Research and Financial Engineering, 2nd ed. Springer. New York.

Bishop, C.M., 2006. Pattern recognition and machine learning, EAI/Springer innovations in communication and computing.

Boukouvala, F., Ierapetritou, M.G., 2012. Feasibility analysis of black-box processes using an adaptive sampling Kriging-based method. Comput. Chem. Eng. 36, 358–368. https://doi.org/10.1016/j.compchemeng.2011.06.005.

Boukouvala, F., Muzzio, F.J., Ierapetritou, M.G., 2010. Design space of pharmaceutical processes using data-driven-based methods. J. Pharm. Innov. 5, 119–137. https://doi.org/10.1007/s12247-010-9086-y.

Boukouvala, F., Muzzio, F.J., Ierapetritou, M.G., 2011. Feasibility analysis of black-box processes using an adaptive sampling kriging based method. Computer Aided Chemical Engineering. Elsevier B.V., pp. 432–436. https://doi.org/10.1016/B978-0-444-53711-9.50087-0

Bynum, M.L., Hackebeil, G.A., Hart, W.E., Laird, C.D., Nicholson, B.L., Siirola, J.D., Watson, J.-P., 2021. Pyomo - optimization modeling in Python. Springer Optimization and It's Applications, Third edition. Springer. https://doi.org/10.1007/978-3-030-68928-5. Cham, Switzerland.

Carey, G.F., Finlayson, B.A., 1975. Orthogonal collocation on finite elements. Chem. Eng. Sci. 30, 587–596. https://doi.org/10.1016/0009-2509(75)80031-5.

Cococcioni, M., Fiaschi, L., 2021. The Big-M method with the numerical infinite M. Optim. Lett. 15, 2455–2468. https://doi.org/10.1007/s11590-020-01644-6.

Costa, L., Oliveira, P., 2001. Evolutionary algorithms approach to the solution of mixed integer non-linear programming problems. Comput. Chem. Eng. 25, 257–266. https://doi.org/10.1016/S0098-1354(00)00653-0.

Cozad, A., Sahinidis, N.V., 2018. A global MINLP approach to symbolic regression. Math. Program. 170, 97–119. https://doi.org/10.1007/s10107-018-1289-x.

Cranmer, M., Sanchez-Gonzalez, A., Battaglia, P., Xu, R., Cranmer, K., Spergel, D., Ho, S., 2020. Discovering symbolic models from deep learning with inductive biases. In: Proceedings of the Advances in Neural Information Processing Systems 2020-December, pp. 1–14.

Cranmer, M., 2020. PySR: fast And parallelized symbolic regression in Python/Julia. https://doi.org/10.5281/zenodo.4041459.

Cranmer, M., 2023. Interpretable machine learning for science with PySR and SymbolicRegression.jl.

Ding, C., Ierapetritou, M., 2021. A novel framework of surrogate-based feasibility analysis for establishing design space of twin-column continuous chromatography. Int. J. Pharm. 609, 121161 https://doi.org/10.1016/j.ijpharm.2021.121161.

Diveev, A., Shmalko, E., 2021. Machine Learning Control By Symbolic Regression. Springer International Publishing, Cham. https://doi.org/10.1007/978-3-030-83213-1. Machine Learning Control by Symbolic Regression.

Dormand, J.R., Prince, P.J., 1980. A family of embedded Runge–Kutta formulae. J. Comput. Appl. Math. 6, 19–26. https://doi.org/10.1016/0771-050X(80)90013-3.

Ehrenstein, M., Wang, C.-H., Guillén-Gosálbez, G., 2019. Strategic planning of supply chains considering extreme events: novel heuristic and application to the petrochemical industry. Comput. Chem. Eng. 125, 306–323. https://doi.org/10.1016/j.compchemeng.2019.03.020.

Ferreira, J., Pedemonte, M., Torres, A.I., 2019a. A genetic programming approach for construction of surrogate models. Computer Aided Chemical Engineering. Elsevier, pp. 451–456. https://doi.org/10.1016/B978-0-12-818597-1.50072-2.

Ferreira, J., Torres, A.I., Pedemonte, M., 2019b. A comparative study on the numerical performance of Kaizen programming and genetic programming for symbolic regression problems. In: Proceedings of the 2019 IEEE Latin American Conference on

Computational Intelligence (LA-CCI), pp. 1–6. https://doi.org/10.1109/LA-CCI47412.2019.9036755.
Floudas, C.A., Gümüş, Z.H., Ierapetritou, M.G., 2001. Global optimization in design under uncertainty: feasibility test and flexibility index problems. Ind. Eng. Chem. Res. 40, 4267–4282. https://doi.org/10.1021/ie001014g.
Forster, T., Vázquez, D., Guillén-Gosálbez, G., 2023. Algebraic surrogate-based process optimization using Bayesian symbolic learning. AIChE J. e18110. https://doi.org/10.1002/aic.18110.
Goyal, V., Ierapetritou, M.G., 2002. Determination of operability limits using simplicial approximation. AIChE J. 48, 2902–2909. https://doi.org/10.1002/aic.690481217.
Goyal, V., Ierapetritou, M.G., 2003. Framework for evaluating the feasibility/operability of nonconvex processes. AIChE J. 49, 1233–1240. https://doi.org/10.1002/aic.690490514.
Grünwald, P.D., 2007. The Minimum Description Length Principle. The MIT Press.
Grossmann, I.E., Floudas, C.A., 1987. Active constraint strategy for flexibility analysis in chemical processes. Comput. Chem. Eng. 11, 675–693. https://doi.org/10.1016/0098-1354(87)87011-4.
Grossmann, I.E., Halemane, K.P., Swaney, R.E., 1983. Optimization strategies for flexible chemical processes. Comput. Chem. Eng. 7, 439–462. https://doi.org/10.1016/0098-1354(83)80022-2.
Grossmann, I.E., Calfa, B.A., Garcia-Herreros, P., 2014. Evolution of concepts and models for quantifying resiliency and flexibility of chemical processes. Comput. Chem. Eng. 70, 22–34. https://doi.org/10.1016/j.compchemeng.2013.12.013. Manfred Morari Special Issue.
Grossmann, I.E., Apap, R.M., Calfa, B.A., García-Herreros, P., Zhang, Q., 2016. Recent advances in mathematical programming techniques for the optimization of process systems under uncertainty. In: Proceedings of the Computers & Chemical Engineering, 12th International Symposium on Process Systems Engineering & 25th European Symposium of Computer Aided Process Engineering (PSE-2015/ESCAPE-25), 31 May - 4 June 2015. Copenhagen, Denmark, 91, pp. 3–14. https://doi.org/10.1016/j.compchemeng.2016.03.002.
Guillén-Gosálbez, G., Miró, A., Alves, R., Sorribas, A., Jiménez, L., 2013. Identification of regulatory structure and kinetic parameters of biochemical networks *via* mixed-integer dynamic optimization. BMC. Syst. Biol. 7, 113. https://doi.org/10.1186/1752-0509-7-113.
Guimerà, R., Reichardt, I., Aguilar-Mogas, A., Massucci, F.A., Miranda, M., Pallarès, J., Sales-Pardo, M., 2020. A Bayesian machine scientist to aid in the solution of challenging scientific problems. Sci. Adv. 6 https://doi.org/10.1126/sciadv.aav6971.
Halemane, K.P., Grossmann, I.E., 1983. Optimal process design under uncertainty. AIChE J. 29, 425–433. https://doi.org/10.1002/aic.690290312.
Hansen, M.H., Yu, B., 2001. Model selection and the principle of minimum description length. J. Am. Stat. Assoc. 96, 746–774. https://doi.org/10.1198/016214501753168398.
Hart, W.E., Watson, J.-P., Woodruff, D.L., 2011. Pyomo: modeling and solving mathematical programs in Python. Math. Prog. Comp. 3, 219–260. https://doi.org/10.1007/s12532-011-0026-8.
Hastings, W.K., 1970. Monte Carlo sampling methods using markov chains and their applications. Biometrika 57, 97. https://doi.org/10.2307/2334940.
Hedengren, J.D., Shishavan, R.A., Powell, K.M., Edgar, T.F., 2014. Nonlinear modeling, estimation and predictive control in APMonitor. Comput. Chem. Eng. 70, 133–148. https://doi.org/10.1016/j.compchemeng.2014.04.013.
Ierapetritou, M.G., Pistikopoulos, E.N., 1994. Novel optimization approach of stochastic planning models. Ind. Eng. Chem. Res. 33, 1930–1942. https://doi.org/10.1021/ie00032a007.
Ierapetritou, M.G., 2001. New approach for quantifying process feasibility: convex and 1-D quasi-convex regions. AIChE J. 47, 1407–1417. https://doi.org/10.1002/aic.690470616.
Jog, S., Vázquez, D., Santos, L.F., Caballero, J.A., Guillén-Gosálbez, G., 2023. Hybrid analytical surrogate-based process optimization *via* Bayesian symbolic regression. Comput. Chem. Eng. 108563 https://doi.org/10.1016/j.compchemeng.2023.108563.
Keane, M.A., Koza, J.R., Rice, J.P., 1993. Finding an impulse response function using genetic programming. In: Proceedings of the 1993 American Control Conference. IEEE, pp. 2345–2350. https://doi.org/10.23919/ACC.1993.4793307.
Krige, D.G., 1951. A statistical approach to some basic mine valuation problems on the Witwatersrand. J. Chem. Metal. Min. Soc. S. Afr. 119–139. https://doi.org/10.10520/AJA0038223X_4792.
Kubic, W.L., Stein, F.P., 1988. A theory of design reliability using probability and fuzzy sets. AIChE J. 34, 583–601. https://doi.org/10.1002/aic.690340408.
Li, C., Grossmann, I.E., 2021. A review of stochastic programming methods for optimization of process systems under uncertainty. Front. Chem. Eng. 2, 1–20.
Li, Z., Ierapetritou, M., 2008a. Process scheduling under uncertainty: review and challenges. Comput. Chem. Eng. 32, 715–727. https://doi.org/10.1016/j.compchemeng.2007.03.001. Festschrift devoted to Rex Reklaitis on his 65th Birthday.
Li, Z., Ierapetritou, M.G., 2008b. Robust optimization for process scheduling under uncertainty. Ind. Eng. Chem. Res. 47, 4148–4157. https://doi.org/10.1021/ie071431u.
Li, Z., Ierapetritou, M.G., 2012. Capacity expansion planning through augmented Lagrangian optimization and scenario decomposition. AIChE J. 58, 871–883. https://doi.org/10.1002/aic.12614.
Li, Z., Ding, R., Floudas, C.A., 2011. A comparative theoretical and computational study on robust counterpart optimization: I. Robust linear optimization and robust mixed integer linear optimization. Ind. Eng. Chem. Res. 50, 10567–10603. https://doi.org/10.1021/ie200150p.
Lin, X., Janak, S.L., Floudas, C.A., 2004. A new robust optimization approach for scheduling under uncertainty: I. Bounded uncertainty. Comput. Chem. Eng. 28, 1069–1085. https://doi.org/10.1016/j.compchemeng.2003.09.020. FOCAPO 2003 Special issue.
Marti, K., Kall, P. (Eds.), 1995. Stochastic programming: numerical techniques and engineering applications: proceedings of the 2nd GAMM/IFIP-Workshop on "Stochastic Optimization: numerical Methods and Technical Applications", held at the Federal Armed Forces University Munich, Neubiberg/München, Germany, June 15-17, 1993, Lecture Notes in Economics and Mathematical systems. Presented at the GAMM/IFIP-Workshop On "Stochastic Optimization: Numerical Methods and Technical Applications," Springer, Berlin; New York.
McKay, B., Willis, M., Barton, G., 1997. Steady-state modelling of chemical process systems using genetic programming. Comput. Chem. Eng. 21, 981–996. https://doi.org/10.1016/S0098-1354(96)00329-8.
McKay, B., Willis, M., Searson, D., Montague, G., 1999. Non-linear continuum regression using genetic programming. In: Proceedings of the Genetic and Evolutionary Computation Conf. (GECCO)-99, 2, pp. 1106–1111.
Metta, N., Ramachandran, R., Ierapetritou, M., 2021. A novel adaptive sampling based methodology for feasible region identification of compute intensive models using artificial neural network. AIChE J. 67, e17095. https://doi.org/10.1002/aic.17095.
Migdalas, A., Pardalos, P.M., Värbrand, P., 1998. Multilevel Optimization: Algorithms and Applications, Nonconvex Optimization and Its Applications. Springer US. https://doi.org/10.1007/978-1-4613-0307-7. Boston, MA.
Morari, M., Grimm, W., Oglesby, M.J., Prosser, I.D., 1985. Design of resilient processing plants—VII. Design of energy management system for unstable reactors—new insights. Chem. Eng. Sci. 40, 187–198. https://doi.org/10.1016/0009-2509(85)80058-0.
Murphy, K.P., 2013. Machine Learning: A Probabilistic Perspective. The MIT Press.
Negri, V., Vázquez, D., Sales-Pardo, M., Guimerà, R., Guillén-Gosálbez, G., 2022. Bayesian symbolic learning to build analytical correlations from rigorous process simulations: application to CO_2 capture technologies. ACS Omega 7, 41147–41164. https://doi.org/10.1021/acsomega.2c04736.
Ochoa, M.P., Grossmann, I.E., 2020. Novel MINLP formulations for flexibility analysis for measured and unmeasured uncertain parameters. Comput. Chem. Eng. 135, 106727 https://doi.org/10.1016/j.compchemeng.2020.106727.
Ostrovsky, G.M., Volin, Yu.M., Barit, E.I., Senyavin, M.M., 1994. Flexibility analysis and optimization of chemical plants with uncertain parameters. Comput. Chem. Eng. 18, 755–767. https://doi.org/10.1016/0098-1354(93)E0013-Y.
Petkov, S.B., Maranas, C.D., 1997. Multiperiod planning and scheduling of multiproduct batch plants under demand uncertainty. Ind. Eng. Chem. Res. 36, 4864–4881. https://doi.org/10.1021/ie970259z.
Pistikopoulos, E.N., Ierapetritou, M.G., 1995. Novel approach for optimal process design under uncertainty. Comput. Chem. Eng. 19, 1089–1110. https://doi.org/10.1016/0098-1354(94)00093-4.
Pistikopoulos, E.N., Mazzuchi, T.A., 1990. A novel flexibility analysis approach for processes with stochastic parameters. Comput. Chem. Eng. 14, 991–1000. https://doi.org/10.1016/0098-1354(90)87055-T.
Pistikopoulos, E.N., 1995. Uncertainty in process design and operations. Comput. Chem. Eng. Eur. Symp. Comput. Aided Process Eng. 3-5 (19), 553–563. https://doi.org/10.1016/0098-1354(95)87094-6.
Prékopa, A., 2011. Stochastic Programming. Springer. Dordrecht.
Pulsipher, J.L., Rios, D., Zavala, V.M., 2019. A computational framework for quantifying and analyzing system flexibility. Comput. Chem. Eng. 126, 342–355. https://doi.org/10.1016/j.compchemeng.2019.04.024.
Rasmussen, C.E., Williams, C.K.I., 2006. Gaussian Processes For Machine learning, Adaptive Computation and Machine Learning. MIT Press. Cambridge, Mass.
Rogers, A., Ierapetritou, M., 2015a. Feasibility and flexibility analysis of black-box processes part 2: surrogate-based flexibility analysis. Chem. Eng. Sci. 137, 1005–1013. https://doi.org/10.1016/j.ces.2015.06.026.
Rogers, A., Ierapetritou, M., 2015b. Feasibility and flexibility analysis of black-box processes Part 1: surrogate-based feasibility analysis. Chem. Eng. Sci. 137, 986–1004. https://doi.org/10.1016/j.ces.2015.06.014.
Sachio, S., Kontoravdi, C., Papathanasiou, M.M., 2023. A model-based approach towards accelerated process development: a case study on chromatography. Chem. Eng. Res. Des. 197, 800–820. https://doi.org/10.1016/j.cherd.2023.08.016.
Sahinidis, N.V., 1996. BARON: a general purpose global optimization software package. J. Glob. Optim. 8, 201–205. https://doi.org/10.1007/BF00138693.
Sahinidis, N.V., 2004. Optimization under uncertainty: state-of-the-art and opportunities. Comput. Chem. Eng. 28, 971–983. https://doi.org/10.1016/j.compchemeng.2003.09.017.
Schmidt, M., Lipson, H., 2009. Distilling free-form natural laws from experimental data. Science 324, 81–85. https://doi.org/10.1126/science.1165893 (1979).
Shapiro, A., Dentcheva, D., Ruszczyński, A.P., 2021. Lectures on stochastic programming: modeling and theory. MOS-SIAM series on optimization, 3rd ed. The Society for Industrial and Applied Mathematics and the Mathematical Optimization Society, Philadelphia.
Straub, D.A., Grossmann, I.E., 1990. Integrated stochastic metric of flexibility for systems with discrete state and continuous parameter uncertainties. Comput. Chem. Eng. 14, 967–985. https://doi.org/10.1016/0098-1354(90)87053-R.
Straub, D.A., Grossmann, I.E., 1993. Design optimization of stochastic flexibility. Comput. Chem. Eng. Int. J. Comput. Appl. Chem. Eng. 17, 339–354. https://doi.org/10.1016/0098-1354(93)80025-I.
Swaney, R.E., Grossmann, I.E., 1985a. An index for operational flexibility in chemical process design. Part I: formulation and theory. AIChE J. 31, 621–630. https://doi.org/10.1002/aic.690310412.

Swaney, R.E., Grossmann, I.E., 1985b. An index for operational flexibility in chemical process design. Part II: computational algorithms. AIChE J. 31, 631–641. https://doi.org/10.1002/aic.690310413.
TuringBot, 2023. Symbolic regression software.
US Food and Drug Administration (FDA), 2010. International Conference on Harmonisation (ICH) Q8 guidance for industry on pharmaceutical development.
Vázquez, D., Guimerà, R., Sales-Pardo, M., Guillén-Gosálbez, G., 2022. Automatic modeling of socioeconomic drivers of energy consumption and pollution using Bayesian symbolic regression. Sustain. Prod. Consum. 30, 596–607. https://doi.org/10.1016/j.spc.2021.12.025.
Wang, Z., Ierapetritou, M., 2017. A novel feasibility analysis method for black-box processes using a radial basis function adaptive sampling approach. AIChE J. 63, 532–550. https://doi.org/10.1002/aic.15362.
Wilson, Z.T., Sahinidis, N.V., 2017. The ALAMO approach to machine learning. Comput. Chem. Eng. 106, 785–795. https://doi.org/10.1016/j.compchemeng.2017.02.010.
Zhang, Q., Grossmann, I.E., Lima, R.M., 2016. On the relation between flexibility analysis and robust optimization for linear systems. AIChE J. 62, 3109–3123. https://doi.org/10.1002/aic.15221.