Novel Formulations of Flexibility Index and Design Centering for Design Space Definition

Fei Zhao, M. Paz Ochoa, Ignacio E. Grossmann*

Center for Advanced Process Decision-Making, Department of Chemical Engineering, Carnegie Mellon University, Pittsburgh, PA 15213

Salvador García Muñoz, Stephen D. Stamatis

Synthetic Molecule Design and Development, Lilly Research Laboratories, Indianapolis, IN 46285

Abstract

Design space definition is one of the key parts in pharmaceutical research and development. Flexibility index and design centering are two complementary ways to estimate a candidate design space. In this study, we first propose a novel formulation of flexibility index based on a direction search method, which is applied to any shape of feasible region. Then, we propose two methods for design centering problems. The vertex direction search method is developed as a single-level optimization model, which is applicable for convex regions. A derivative-free optimization (DFO) method is developed based on the proposed flexibility index model, which is applicable to convex and nonconvex problems. In order to find near global solutions, Latin Hypercube Sampling (LHS) is used to generate multiple starting points for the DFO solver. The optimal nominal point is the candidate point with the largest flexibility index. Several cases demonstrate the efficiency of the proposed methods.

Keywords: design space; flexibility index; design centering; derivative-free optimization.

*Correspondence concerning this article should be addressed to grossmann@cmu.edu.

1. Introduction

In order to increase manufacturing flexibility in the pharmaceutical industry, Quality by Design (QbD)[1] was launched by the US Food and Drug Administration (FDA). The design space is defined as "the multidimensional combination and interaction of input variables and process parameters that have been demonstrated to provide assurance of quality"[2]. The applicant must demonstrate that the product quality meets specifications as long as the process parameters vary within the proposed design space. Process parameters[3] correspond to degrees of freedom or variables that can be manipulated in the operation of a manufacturing process, and which can be measured and set within the controller tolerance for a desired value. Design spaces are defined using predictions from a model (either empirical or deterministic). Thus, the uncertainty in the model parameters plays an important role. The values of the model parameters are estimated from data, and commonly assumed to follow a Gaussian distribution from which confidence intervals can be defined.

Early approaches to identify the design space were solely based on experiments and empirical functions. By performing extensive experiments, the relationships of process parameters and critical quality attributes (CQAs) can be established through regression, and the process parameters that have medium/high impacts on the CQAs can be determined. The design space can be depicted by response surface modeling and further be verified by additional experiments[4]. This method requires performing extensive experiments, and it is generally very time-consuming and expensive. To lower the cost of developing design spaces, mechanistic models that contain relationships of process parameters and CQAs can be formulated in advance and parametrized with less data. Regarding the use of mechanistic models for design space identification, Goyal and Ierapetritou [5] proposed an approach based on outer approximation to identify the operating envelopes where process operation is feasible, safe and profitable. García Muñoz et al.[4] defined the probabilistic design space by creating a grid of sample points for the process parameters and a Monte Carlo simulation to propagate the uncertainty of the model parameters (exhaustive sampling). Kucherenko et al.[6] proposed an acceptance-rejection method that outperforms exhaustive sampling by achieving a two-order of magnitude speed-up with metamodeling and adaptive sampling in the design space determination. Apart from using numerical computation methods to estimate the contour of the design space, Zhao and Chen[7] proposed representing the design model as an existential quantifier formula, and then applied a symbolic computation method to accurately describe the design space and explicitly express the functional relationships between uncertain parameters. Due to the heavy computational burden incurred by symbolic computation, the method is only applicable to relatively small-scale problems.

Moreover, the optimization approaches based on models have been intensively studied to describe the design space[8]. Characterizing a design space for a process design model is analogous to the flexibility analysis problem in the chemical process industry[9]. Two classical flexibility analysis problems are flexibility test and flexibility index[10,11]. The former can verify if feasible operation can be obtained for a given range of uncertainty scenarios. The latter can be used to describe an operational range, which represents a maximum scaled departure of all process parameters from the given nominal conditions, such as a largest rectangle inscribed in the feasible space within which steady-state operation can be attained by adjusting control variables. Since the design space is limited by the qualification ranges for process parameters, the result of flexibility index can be used to approximate the design space as an inscribed largest feasible region, which may be a rectangle, ellipse or other appropriate and acceptable shapes. Moreover, if the nominal conditions of the process parameters are unknown, the flexibility index problem can be extended to a design centering problem, which focuses on determining the optimal nominal conditions while maximizing the feasible operating region. Notably, while the center of the design is key to identify the flexibility region, this center point may not always represent the best nominal conditions to operate the process (i.e. the chosen nominal condition is not necessarily the center of the design space).

Flexibility index and design centering are two important problems in the design space definition. In this work, we propose novel formulations to efficiently deal with these two problems. The rest of this article is organized as follows. Sections 2 and 3 provide problem statements and reviews of previous methods for flexibility index and design centering. Section 4 describes the formulation of flexibility index based on the direction search method, and the shapes of feasible region are discussed. Section 5 proposes two methods for design centering, including vertex direction search method and derivative-free optimization method. Several numerical examples and cases are provided in Section 6 to illustrate the proposed methods. Section 7 concludes the paper.

2. Problem statements

Flexibility index problems are commonly formulated as multi-level optimization models with existing approaches relying on mixed-integer linear or nonlinear programming solvers[12], and all the model constraints will be complied with. Before solving the optimization problems, the nominal conditions of the process parameters should be given. In this work, for the flexibility index problem we will not consider recourse decisions as this is not common in the pharmaceutical industry, and in order to simplify the multi-level optimization formulation, the model parameters will be fixed at their mean values of the Gaussian distribution.

For a given plant design, the flexibility constraint with no recourse can be described as a logic expression as follows[9]:

$$\forall \boldsymbol{\theta} \in T_P \{\forall j \in J[g_j(\boldsymbol{\theta}, \boldsymbol{x}) \leq 0], \forall i \in I[h_i(\boldsymbol{\theta}, \boldsymbol{x}) = 0]\} \quad (1)$$

where $\boldsymbol{\theta}$ and $\boldsymbol{x}$ represent process parameters and state variables, respectively. Eq. (1) states that for any possible realization of the process parameters in T_P, all of the individual constraints should be satisfied. Eq. (1) can be equivalently reformulated by the use of global max operator, leading to Eq. (2).

$$\begin{aligned} &\chi = \max_{\boldsymbol{\theta} \in T_p} \max_{j \in J} g_j(\boldsymbol{\theta}, \boldsymbol{x}) \leq 0 \\ &s.t.\ \ h_i(\boldsymbol{\theta}, \boldsymbol{x}) = 0, \ \ \forall i \in I \end{aligned} \tag{2}$$

where the maximization problem in χ determines the worst constraint violation. The flexibility index problem with no recourse can be described by the following model[10].

$$\begin{aligned} &F = \max_{\delta \in \mathbb{R}^+} \delta \\ &s.t.\ \ \chi = \max_{\boldsymbol{\theta} \in T_P} \max_{j \in J} g_j(\boldsymbol{\theta}, \boldsymbol{x}) \leq 0 \\ &\qquad h_i(\boldsymbol{\theta}, \boldsymbol{x}) = 0, \ \ \forall i \in I \\ &\qquad T_P(\boldsymbol{\theta}) = \{\boldsymbol{\theta} : \boldsymbol{\theta}^N - \delta \Delta \boldsymbol{\theta}^- \leq \boldsymbol{\theta} \leq \boldsymbol{\theta}^N + \delta \Delta \boldsymbol{\theta}^+\} \end{aligned} \tag{3}$$

The flexibility index F is defined as the largest value of δ for the set of process parameters, and in Eq. (3), the set $T_P(\boldsymbol{\theta})$ is described as a rectangle. Note that, χ requires that all of the process parameters in $T_P(\boldsymbol{\theta})$ should satisfy the model constraints g_j and h_i, which is a semi-infinite programming problem. To solve Eq. (3), the complementarity conditions with mixed-integer constraints are commonly used, and Haar condition[13] is assumed to hold, i.e., the no recourse case states that the number of active constraints is equal to one. This condition ensures that the Karush-Kuhn-Tucker (KKT) conditions are necessary and sufficient. Geometrically, χ in Eq. (3) represents a bi-level optimization model to define a rectangle inscribed within the feasible region. Generally, for the flexibility index problem there is only one vertex (for a convex feasible region) or side (for a nonconvex feasible region) of that rectangle that lies on the boundary. In a design centering problem, the nominal point, $\boldsymbol{\theta}^N$, with the greatest flexibility index is sought by the optimizer. The generic formulation of design centering can be described as follows.

$$
\begin{aligned}
F = \max_{\boldsymbol{\theta}^N} \delta \\
s.t.\ & g_j(\boldsymbol{\theta}^N, \boldsymbol{x}) \le 0, \ \ \forall j \in J \\
& h_i(\boldsymbol{\theta}^N, \boldsymbol{x}) = 0, \ \ \forall i \in I \\
& \max_{\delta \in \mathbb{R}^+} \delta \\
& s.t.\ \chi = \max_{\boldsymbol{\theta} \in T_P} \max_{j \in J} g_j(\boldsymbol{\theta}, \boldsymbol{x}) \le 0 \\
& \qquad h_i(\boldsymbol{\theta}, \boldsymbol{x}) = 0, \ \ \forall i \in I \\
& \qquad T_P(\boldsymbol{\theta}) = \{\boldsymbol{\theta}: \boldsymbol{\theta}^N - \delta\Delta\boldsymbol{\theta}^- \le \boldsymbol{\theta} \le \boldsymbol{\theta}^N + \delta\Delta\boldsymbol{\theta}^+\}
\end{aligned}
$$

The flexibility index model is applied for each candidate nominal point. Various optimization methods can be employed to solve the design centering problem. We review several candidates in the next section and demonstrate reformulations of MINLP and DFO in Sections 4 and 5 respectively.

3. Review of previous methods

If the shape of the feasible operating region is specified as a rectangle, the computationally expensive vertex direction search method[10] can be employed as a straightforward brute force search tactic that is guaranteed to be rigorous for convex regions, though it cannot guarantee a rigorous solution for nonconvex regions. However, for nonconvex design spaces, the vertex search method cannot guarantee to provide rigorous solutions. In order to avoid the convexity assumption, Grossmann and Floudas[14] developed an active constraint strategy, where the flexibility index problem can be reformulated as a mixed-integer linear or nonlinear programming model by applying the KKT conditions. However, for a large-scale problem, it is often challenging to solve the corresponding MINLP model to global optimality. Li et al.[15] developed a direction matrix to search the critical points. By incorporating a simulated annealing algorithm and a decoupling strategy, the flexibility index of a large-scale system can be obtained.

A number of approaches have been proposed to quantify system flexibility, and an extensive review is provided by Grossmann et al.[12]. Apart from the rectangular uncertainty set, Pulsipher and Zavala[16] proposed to use of multivariate Gaussian random variables, i.e., applying an ellipsoidal set to capture correlations of process parameters. The flexibility index can be computed by solving a mixed-integer conic programming (MICP) problem. This method also can be generalized to capture different shapes of uncertainty sets. Pulsipher et al.[17] presented a computational framework for analyzing and quantifying system flexibility, which can generalize the uncertainty sets to consider compositions of sets, compute a suitable nominal point, and identify and rank limiting constraints. Since the rectangle representations of the uncertainty set cannot adequately capture correlations of the parameters[18], the ellipsoid shapes may have a larger potential for application.

Director and Hachtel[19] addressed the design centering problem of choosing a nominal design point to maximize the number of circuits that satisfy performance tolerances. The authors proposed the simplicial approximation approach, based on the explicit approximation of the boundary of an *n*-parameter design space by a polyhedron made up of *n*-dimensional simplices. Optimization approaches based on models can also be applied to solve design centering problems. From a mathematical view, the design centering problem is a classical generalized semi-infinite programming (GSIP) problem[20,21]. A GSIP problem is characterized by a finite number of decision variables and an infinite number of inequality constraints. Since the nominal point is not given, the location of the feasible region is unknown. All of the points within the feasible region must satisfy all the model constraints, which means that feasibility must be guaranteed for an infinite number of constraints. Stein[22] showed that the Reduction Ansatz of semi-infinite programming generically holds at each solution of the reformulated design centering problem and proved a new first order necessary optimality condition for design centering model. Hardwood and Barton[23] formulated the design centering problem as a GSIP model and discussed reformulations to simpler problems that lead to finite nonlinear programs (NLPs) or standard semi-infinite programs (SIP). Following the GSIP methods, the obvious drawback for design centering is that they give rise to complex mathematical programs with either semi-infinite or chance constraints that are computationally hard to tackle rigorously.

In order to avoid solving the complex GSIP problems, in this work, we propose to use the flexibility index formulation to address the design centering problem with a DFO method. First, a novel bi-level optimization model of flexibility index based on direction search method is proposed, which can be extended to any description of uncertainty set. By applying the KKT conditions, the flexibility index can be transformed into a single-level formulation. Based on this, a DFO method using multiple starting points is adopted to search for the nominal point within the design space with the largest feasible region.

4. New formulations for flexibility index

In order to circumvent the semi-infinite programming problem for flexibility index, we propose a new and simpler formulation based on the following direction search formulation.

$$\boldsymbol{\theta} = \boldsymbol{\theta}^N + \delta\widetilde{\boldsymbol{\theta}} \tag{4}$$

where the vector $\widetilde{\boldsymbol{\theta}}$ represents a direction from the nominal point[10]. Along this direction, if $\boldsymbol{\theta}$ can satisfy all the constraints and make at least one inequality constraint active, i.e.,

$$\begin{cases} g_j(\boldsymbol{\theta}) \leq 0, & \forall j \in J \\ g_s(\boldsymbol{\theta}) = 0, & \exists s \in J \end{cases} \tag{5}$$

δ is the largest value along this direction. Figure 1 shows two directions from $\boldsymbol{\theta}^N$, i.e., $\widetilde{\boldsymbol{\theta}}_1$ and $\widetilde{\boldsymbol{\theta}}_2$. $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ are farthest feasible points along the directions.

$$\begin{matrix} \boldsymbol{\theta}_1 = \boldsymbol{\theta}^N + \delta_1\widetilde{\boldsymbol{\theta}}_1 \\ \boldsymbol{\theta}_2 = \boldsymbol{\theta}^N + \delta_2\widetilde{\boldsymbol{\theta}}_2 \end{matrix} \quad \text{and} \quad \begin{matrix} g(\boldsymbol{\theta}_1) \leq 0 \\ g(\boldsymbol{\theta}_2) \leq 0 \end{matrix}$$

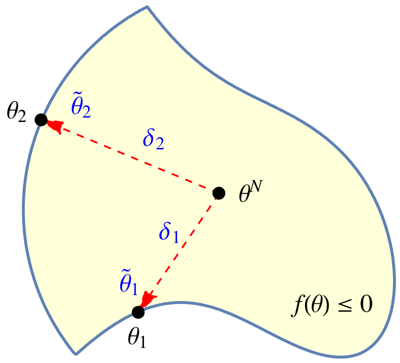


Figure 1. Geometric interpretation of direction search.

The optimization model Eq. (6) is presented, which can be used to calculate the largest δ for each direction $\tilde{\boldsymbol{\theta}}$ from the nominal point $\boldsymbol{\theta}^N$.

$$\begin{aligned} & F\left(\theta_p^N, \tilde{\theta}_p\right) = \max_{\delta, \theta_p} \delta \\ & s.t. \;\; g_j(\boldsymbol{\theta}, \boldsymbol{x}) \le 0, \;\; \forall j \in J \\ & \qquad h_i(\boldsymbol{\theta}, \boldsymbol{x}) = 0, \;\; \forall i \in I \\ & \qquad \theta_p = \theta_p^N + \delta \tilde{\theta}_p, \;\; \forall p \in P \\ & \qquad \delta \ge 0 \end{aligned} \tag{6}$$

4.1. Shapes of feasible region

Compared with $T_p(\boldsymbol{\theta})$ in Eq. (3), the rectangle that is used for direction search can be simply defined by Eq. (7), which is not relevant to δ; thus, we can extend the shape from rectangle to any other shape so long as the shape can be explicitly formulated. For instance, a rectangle is formulated as follows.

$$\text{rectangle:} \;\; -\Delta\theta_p^- \le \tilde{\theta}_p \le \Delta\theta_p^+, \;\; \forall p \in P \tag{7}$$

A standard ellipse is formulated as Eq. (8).

$$\text{ellipse:} \;\; \sum_{p=1}^{P} \left(\frac{\tilde{\theta}_p}{\Delta\theta_p}\right)^2 = 1 \tag{8}$$

where $\Delta\theta_p$ represents the given radius for each parameter p, which determines the shape of the ellipse. If all the $\Delta\theta_p$ $\forall p \in P$ are the same, say equal to $\Delta\theta^*$, the ellipse becomes a circle.

$$\text{Circle:} \quad \sum_{p=1}^{P} \left(\frac{\tilde{\theta}_p}{\Delta\theta^{*2}} \right)^2 = 1 \tag{9}$$

Since the circle is a special case of ellipse, only rectangle and ellipse are considered further in this work.

4.2. Single-level formulation of flexibility index

Once the shape of feasible region is specified, the optimization problem shown in Eq. (6) can be executed to calculate the largest δ for each direction from the given nominal point, and the directions are along the boundary of the shape. Based on the above, a new flexibility index formulation with no recourse is proposed as follows.

$$\begin{aligned} F\left(\theta_p^N\right) = \min_{\tilde{\theta}_p} \max_{\delta, \theta_p} \delta \\ s.t.\ g_j(\boldsymbol{\theta}, \boldsymbol{x}) \leq 0, \ \ \forall j \in J \\ h_i(\boldsymbol{\theta}, \boldsymbol{x}) = 0, \ \ \forall i \in I \\ \theta_p = \theta_p^N + \delta\tilde{\theta}_p, \ \ \forall p \in P \\ shape\left(\tilde{\boldsymbol{\theta}}\right) \in Q \\ \delta \geq 0 \end{aligned} \tag{10}$$

where $shape\left(\tilde{\boldsymbol{\theta}}\right)$ represents the formulation of a specified shape of the feasible region, Q = {Eq. (7), Eq. (8), and any other shapes}; δ represents a scale factor of the shape. The flexibility index F is defined as the minimum value of δ for all of the directions along the shape of the feasible region. Eq. (10) can be equivalently expressed as the following bi-level optimization model.

$$\begin{aligned} F\left(\theta_p^N\right) = \min_{\tilde{\theta}_p} \delta \\ s.t.\ shape\left(\tilde{\boldsymbol{\theta}}\right) \in Q \\ \max_{\delta, \theta_p} \delta \\ s.t.\ g_j(\boldsymbol{\theta}, \boldsymbol{x}) \leq 0, \ \ \forall j \in J \\ h_i(\boldsymbol{\theta}, \boldsymbol{x}) = 0, \ \ \forall i \in I \\ \theta_p = \theta_p^N + \delta\tilde{\theta}_p, \ \ \forall p \in P \\ -\delta \leq 0 \end{aligned} \tag{11}$$

To solve this bi-level optimization model, the inner problem can be replaced by the KKT conditions and complementarity conditions. The Lagrange function is

$$\mathcal{L} = -\delta + \sum_j \lambda_{1,j} \cdot g_j(\boldsymbol{\theta}, \boldsymbol{x}) - \lambda_2 \delta + \sum_i \mu_{1,i} \cdot h_i(\boldsymbol{\theta}, \boldsymbol{x}) + \sum_p \mu_{2,p} \left[\theta_p - \theta_p^N - \delta \tilde{\theta}_p\right] \quad (12)$$

The stationary conditions of the Lagrange function with respect to δ, process parameters θ_p and state variables x_k are as follows:

$$\begin{aligned}
&\frac{\partial \mathcal{L}}{\partial \delta} = -1 - \lambda_2 - \sum_p \mu_{2,p} \tilde{\theta}_p = 0 \\
&\frac{\partial \mathcal{L}}{\partial \theta_p} = \sum_j \lambda_{1,j} \cdot \frac{\partial g_j}{\partial \theta_p} + \sum_i \mu_{1,i} \cdot \frac{\partial h_i}{\partial \theta_p} + \mu_{2,p} = 0, \quad \forall p \in P \\
&\frac{\partial \mathcal{L}}{\partial x_k} = \sum_j \lambda_{1,j} \cdot \frac{\partial g_j}{\partial x_k} + \sum_i \mu_{1,i} \cdot \frac{\partial h_i}{\partial x_k} = 0, \quad \forall k \in K
\end{aligned} \quad (13)$$

The complementarity conditions are

$$\begin{aligned}
&\lambda_{1,j} \cdot g_j(\boldsymbol{\theta}, \boldsymbol{x}) = 0, \quad j \in J \\
&\lambda_2 \cdot \delta = 0 \\
&\lambda_{1,j} \geq 0, \quad j \in J \\
&\lambda_2 \geq 0
\end{aligned} \quad (14)$$

which can be expressed with mixed-integer constraints. *M* corresponds to a big-M value, *s* are slack variables and *y* are binary variables to indicate if the corresponding constraints are active. Thus, a single-level MINLP model can be obtained, as shown in Eq. (15).

$$
\begin{aligned}
& F\left(\theta_p^N\right) = \min \delta \\
& s.t.\ shape\left(\widetilde{\boldsymbol{\theta}}\right) \in Q \\
& \quad -1 - \lambda_2 - \sum_p \mu_{2,p}\, \tilde{\theta}_p = 0 \\
& \quad \sum_j \lambda_{1,j} \cdot \frac{\partial g_j}{\partial \theta_p} + \sum_i \mu_{1,i} \cdot \frac{\partial h_i}{\partial \theta_p} + \mu_{2,p} = 0,\ \ \forall p \in P \\
& \quad \sum_j \lambda_{1,j} \cdot \frac{\partial g_j}{\partial x_k} + \sum_i \mu_{1,i} \cdot \frac{\partial h_i}{\partial x_k} = 0,\ \ \forall k \in K \\
& \quad g_j + s_{1,j} = 0,\ \ \forall j \in J \\
& \quad -\delta + s_2 = 0 \\
& \quad \theta_p = \theta_p^N + \delta \tilde{\theta}_p,\ \ \forall p \in P \\
& \quad s_{1,j} \leq M\left(1 - y_{1,j}\right),\ \ \ \forall j \in J \\
& \quad \lambda_{1,j} - y_{1,j} \leq 0,\ \ \ \forall j \in J \\
& \quad s_2 \leq M(1 - y_2) \\
& \quad \lambda_2 - y_2 \leq 0 \\
& \quad -\delta \leq 0 \\
& \quad \lambda_{1,j} \geq 0 \\
& \quad \lambda_2 \geq 0 \\
& \quad s_{1,j} \geq 0 \\
& \quad s_2 \geq 0 \\
& \quad y_{1,j} \in \{0,1\} \\
& \quad y_2 \in \{0,1\}
\end{aligned}
\qquad (15)
$$

Compared with the traditional flexibility index model shown in Eq. (3), the proposed model has three characteristics:

(1) Compared with Eq. (3), the proposed Eq. (10) is a bi-level model. Although it involves new bilinear terms, the corresponding MINLP model will be simpler because it only needs KKT reformulation once.

(2) The Haar condition is not required. In Eq. (3), a rectangle is usually defined as an expression of $\boldsymbol{\theta}$, $\boldsymbol{\theta}^N$ and δ. Since Eq. (3) requires all the process parameters in the whole rectangle restricted in the feasible region, it is a semi-infinite programming problem; thus, the Haar condition is required for finding the active constraints. By contrast, Eq. (10) has ability to find the direction corresponding to the active constraint directly, and Haar condition becomes unnecessary.

(3) Eq. (10) can be extended to any shape of feasible region, as long as its representation in terms of $\tilde{\boldsymbol{\theta}}$ can be provided.

5. Design centering problem

Another important aspect of the design space definition is the design centering problem, where the objective is to select the nominal conditions of the process parameters while maximizing the feasible region of operation. It can be geometrically interpreted as the problem of inscribing the largest shape of the uncertainty set of the process parameters within the feasible region of operation given by the model constraints. Thus, the difference between the flexibility index calculation and the design centering problem is that for the former problem the nominal point is given, whereas in the latter the flexibility index is maximized whilst selecting the optimal nominal point. The nominal point corresponds to a feasible point, therefore it must satisfy all the model constraints, i.e.,

$$\begin{cases} g_j(\boldsymbol{\theta}^N, \boldsymbol{x}) \leq 0, & \forall j \in J \\ h_i(\boldsymbol{\theta}^N, \boldsymbol{x}) = 0, & \forall i \in I \end{cases} \tag{16}$$

5.1. Bi-level formulation of design centering

Based on the flexibility index model shown in Eq. (15), the design centering problem can be formulated as the following bi-level optimization model. The inner level is the minimization problem of flexibility index, and the outer level is the maximization problem where the nominal point is searched within the feasible region.

$$\begin{aligned} & \max_{\boldsymbol{\theta}^N} \delta \\ & s.t. \ \ g_j(\boldsymbol{\theta}^N, \boldsymbol{x}) \leq 0, \ \ \forall j \in J \\ & \qquad h_i(\boldsymbol{\theta}^N, \boldsymbol{x}) = 0, \ \ \forall i \in I \\ & \qquad \text{Eq. (15)} \end{aligned} \tag{17}$$

Note that, if directly reformulating Eq. (15) by the KKT conditions and complementarity conditions, the obtained single-level optimization model cannot generate a correct result[17]. The result of a linear case is shown in Appendix, and the final nominal point is placed on the boundary of $\boldsymbol{\theta}$.

Since Eq. (17) is a typical bi-level optimization problem[24], the objective of the lower-level problem is to calculate the flexibility index for a given nominal point, and the upper-level problem is to search the nominal points within the feasible region. Generally, the procedure to solve Eq. (17) mainly contains three steps:

(1) Choose an initial nominal point $\boldsymbol{\theta}_0^N$ at the upper level.

(2) Solve the lower-level problem, and find the global minimum solution of δ.

(3) Based on the value of δ, apply a search method to generate a new nominal point $\boldsymbol{\theta}_k^N$, until the stop criterion is attained.

In this section, two different methods are proposed to solve the design centering problems. The first method is vertex direction search, which can be applied for the special case of finding a largest rectangle within a convex feasible region, and the method does not require solving the flexibility index model; the second method is based on DFO, which is applicable to general cases, and a strategy of multiple starts is developed to improve the global optimality.

5.2. Method 1: Vertex direction search for convex cases

The vertex direction search method for design centering is based on the theorem by Swaney and Grossmann[10] that establishes that if the constraint functions are jointly convex in the process parameters and control variables, then the solution of the flexibility constraint has its global optimal solution at a vertex of the polyhedral region that describes the process parameter set. The basic idea of this method is to maximize the flexibility index δ and to determine the nominal condition of the process parameters, θ_p^N , by simultaneously evaluating feasibility over all vertex directions, which is formulated as Eq. (18).

$$
\begin{aligned}
&\max_{\delta \in \mathbb{R}^+, \theta_p^N} \delta \\
&s.t.\ \ g_{j,v}\left(\theta_{p,v}, \boldsymbol{x}\right) \leq 0,\ \ \forall\, j \in J,\ \ v \in VD \\
&\qquad h_{i,v}\left(\theta_{p,v}, \boldsymbol{x}\right) = 0,\ \ \forall i \in I,\ \ v \in VD \\
&\qquad \theta_{p,v} = \theta_p^N + \delta \cdot dev_{p,v},\ \ \forall\, p \in P,\ \ v \in VD
\end{aligned} \tag{18}
$$

where subscripts p and v stand for process parameter and vertex directions, respectively; $\theta_{p,v}$ is the process parameter at each vertex direction. $dev_{p,v}$ is a parameter that contains all vertex directions $v \in VD$. $\Delta\theta_p^+$ and $\Delta\theta_p^-$ represent the allowable ranges of operation for each process parameter, $p \in P$. For the case of two process parameters, VD = $\{(\Delta\theta_1^+, \Delta\theta_2^+), (-\Delta\theta_1^-, \Delta\theta_2^+), (\Delta\theta_1^+, -\Delta\theta_2^-), (-\Delta\theta_1^-, -\Delta\theta_2^-)\}$. As shown in Figure 2, each process parameter at four vertex directions will be added to the optimization model; thus, the total number of the constraints is $(|I| + |J|) \cdot 2^P$. The limitation of this method is that it only allows finding vertex solutions. Furthermore, the size of the LP/NLP problem in Eq. (18) grows exponentially with the number of process parameters, i.e., 2^P. However, the structure of the problem can be exploited by a decomposition scheme when necessary.

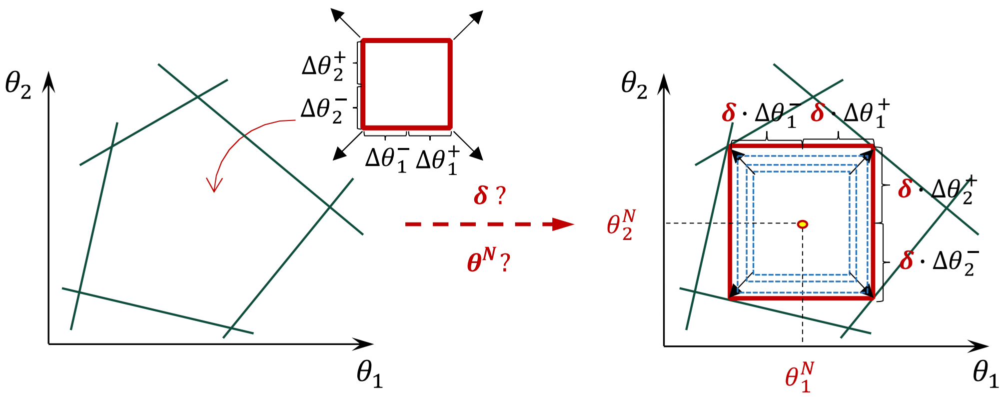


Figure 2. Vertex search method for convex feasible regions.

5.3. Method 2: Derivative-free optimization using multiple starting points

The goal of a design centering problem is to find an optimal nominal point, which corresponds to the largest feasible operating region. Eq. (17) shows that the design centering problem is a bi-level optimization model. In the upper-level problem, the nominal point is searched within the feasible region, and in the lower-level problem, an exact flexibility index should be calculated for each nominal point. For a general bi-level optimization problem, the most important issue is how to guarantee finding the global optimal solution of the lower-level model at each iteration. Similarly, for the design centering problem, the key issue is to guarantee locating the global optimal flexibility index for each nominal point.

Since the MINLP model in Eq. (15) can be solved by the GAMS/BARON solver to obtain the global optimal solution, we can define Eq. (15) as an implicit function of $\boldsymbol{\theta}^N$, i.e.,

$$\delta = F(\boldsymbol{\theta}^N) \tag{19}$$

which indicates that, for an arbitrary nominal point, an exact flexibility index δ can be obtained. Thus, Eq. (17) can be rewritten as Eq. (20).

$$\begin{aligned} &\max_{\boldsymbol{\theta}^N} F(\boldsymbol{\theta}^N) \\ &s.t.\ \ g_j(\boldsymbol{\theta}^N, \boldsymbol{x}) \leq 0, \ \ \forall j \in J \\ &\qquad h_i(\boldsymbol{\theta}^N, \boldsymbol{x}) = 0, \ \ \forall i \in I \end{aligned} \tag{20}$$

which can be viewed as a single-level optimization model with a black-box objective $F(\boldsymbol{\theta}^N)$. Therefore, a DFO method can be applied to solve this model. However, the presence of the state variables implies that the feasible region of nominal points is described by a set of multivariate functions of $\boldsymbol{\theta}^N$ and $\boldsymbol{x}$. Thus, the maximum constraint violation (MCV) of all the constraints is defined in order to identify the feasible region.

$$\begin{aligned} &MCV(\boldsymbol{\theta}^N) = \min u \\ &s.t.\ \ g_j(\boldsymbol{\theta}^N, \boldsymbol{x}) \leq u, \ \ \forall j \in J \\ &\qquad h_i(\boldsymbol{\theta}^N, \boldsymbol{x}) = 0, \ \ \forall i \in I \end{aligned} \tag{21}$$

To make all the constraints feasible, *MCV* should be less than 0, i.e.,

$$MCV(\boldsymbol{\theta}^N) \leq 0 \tag{22}$$

Eq. (20) can then be written as

$$\begin{aligned} &\max_{\boldsymbol{\theta}^N} F(\boldsymbol{\theta}^N) \\ &s.t.\ \ MCV(\boldsymbol{\theta}^N) \leq 0 \end{aligned} \tag{23}$$

Moreover, in order to convert Eq. (23) to a form that is easily handled by general DFO solvers, a penalty coefficient M is introduced in the objective function. The penalty coefficient simply serves as a way to scale the constraint violation and its value need only be tuned for numerical stability. Thus, the final design centering model is

$$\begin{aligned} &\min_{\boldsymbol{\theta}^N} -F(\boldsymbol{\theta}^N) + M \cdot \max\left(0, MCV(\boldsymbol{\theta}^N)\right) \\ &s.t.\ \theta_p^{NL} \le \theta_p^N \le \theta_p^{NU},\ \ p \in P \end{aligned} \tag{24}$$

where the bound constraints of θ_p^N are also given. The objective function in Eq. (24) is a type of exact penalty function. Fiacco[25] proved that if M is sufficiently large, this penalty function will be exact, i.e., it is capable of finding the exact optimal solution. Eq. (24) is a DFO model with a black-box objective function and a box constraint, which can be handled by most DFO solvers. In the special case of a problem without state variables, we do not need to calculate the constraint violations by solving optimization problems. Thus, Eq. (24) can be simplified as

$$\begin{aligned} &\min_{\boldsymbol{\theta}^N} -F(\boldsymbol{\theta}^N) + M \cdot \sum_j^J \left(\max\left(0, g_j(\boldsymbol{\theta}^N)\right)\right) \\ &s.t.\ \theta_p^{NL} \le \theta_p^N \le \theta_p^{NU},\ \ p \in P \end{aligned} \tag{25}$$

For an initial nominal point, the DFO solution strategy to solve the design centering problem is summarized in Figure 3. At the kth iteration, the nominal point $\boldsymbol{\theta}_k^N$ is used to solve a MINLP model and an NLP model, and the results are used to evaluate the objective function. If it does not meet the stopping criteria of the selected DFO solver, e.g., the maximum number of objective evaluations, a new nominal point is generated.

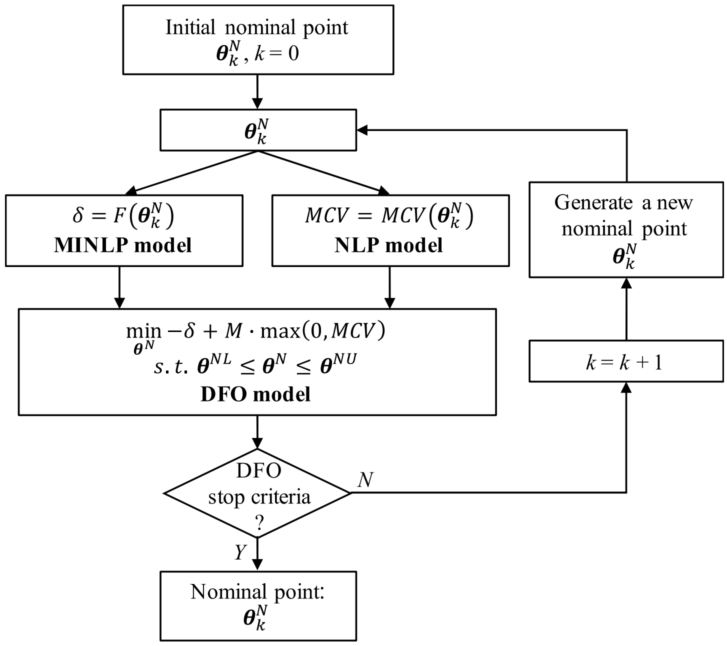


Figure 3. Flowchart of DFO solution strategy to design centering for an initial nominal point.

Theoretically, based on search strategies, DFO methods[26] can be grouped into two types: *direct search* methods, which determine the search directions directly from the function evaluation data, and *model-based* methods, which typically use a trust-region framework for selecting new iterations. In addition, DFO methods[27] can be divided into *local search* methods, which start from an initial guess and move within a local trust region, and *global search* methods, which search the entire bounded variable space. However, neither local nor global search methods are guaranteed to find the global optimum.

In this work, a DFO solver, Py-BOBYQA, which is a Python implementation of the BOBYQA Fortran solver by Powell[28], is employed to solve design centering problems. Py-BOBYQA is designed for the optimization models like Eq. (26).

$$\begin{aligned} &\min_{x \in \mathbb{R}^n} f(x) \\ &s.t.\ a \leq x \leq b \end{aligned} \tag{26}$$

Py-BOBYQA is based on the trust-region method, which can find local solutions of nonlinear, nonconvex, least-squares minimization problems (with box constraints), without requiring derivatives of the objective. Py-BOBYQA approximates the function *f*(*x*) using a quadratic function, which matches the function value of *f*(*x*) at certain interpolation points chosen by the algorithm. The quadratic function is then used in a trust region procedure for updating the decision variables. A more detailed description of the algorithm can be found in [29].

Since Py-BOBYQA was developed based on the trust-region method, the initial value has great influence on the final result of the DFO model. In order to take a more complete assessment of the design space, the Latin hypercube sampling (LHS) strategy is applied to generate a set of initial points in the process parameter space. Then, a feasibility check is performed through evaluating the model constraints at each LHS point. If a point is infeasible, it will be removed. To summarize, this method contains four main steps:

(1) For a given number of sampling points, perform the LHS strategy over the space of process parameters, where upper and lower bounds are required.

(2) Check the feasibility of each sampling point $\boldsymbol{\theta}_{sp}^{N}$ through solving the following NLP model.

$$\begin{aligned} u_{sp} &= \min u \\ s.t.\, g_j\left(\boldsymbol{\theta}_{sp}^{N}, \boldsymbol{x}\right) &\leq u, \quad \forall j \in J \\ h_i\left(\boldsymbol{\theta}_{sp}^{N}, \boldsymbol{x}\right) &= 0, \quad \forall i \in I \end{aligned} \tag{27}$$

where $u_{sp} \leq 0$ indicates that the point is feasible.

(3) Solve the DFO model of design centering problem for each feasible LHS point by using Eq. (24), and the obtained result for each point is stored.

(4) The optimal nominal point is the one that has the largest value of flexibility index.

Note that the number of the LHS sampling points needs to be specified in advance, and the more LHS feasible points we can obtain, the more rigorous the solution may be. The design centering model for each LHS point is actually calculated independently, so the problem can be solved in parallel. The pseudocode of the above DFO strategy using multiple starting points is described in Algorithm 1.

Algorithm 1: DFO method with multiple starting points

1: Perform LHS method to discretize the process parameter space:
$\boldsymbol{\theta}^{N} = \{\boldsymbol{\theta}_{sp}^{N}, \ \forall sp \in S\}$

2: **for** each nominal point $\boldsymbol{\theta}_{sp}^{N}$, $sp \in S$

```
    Check feasibility of the nominal point. (Eq. (27)):
    if nominal point is feasible u_sp ≤ 0 then
        Conserve point θ^N_sp in the S'.
    else
        Exclude point θ^N_sp from S.
    end
end
for each feasible nominal point θ^N_sp, sp ∈ S'
    Solve the DFO model at θ^N_sp:
     Set the initial conditions for the DFO solver.
     while the stop criteria of the DFO solver does not meet:
        Calculate the flexibility index: F(θ^N_sp). (Eq. (15))
        Calculate the maximum constraint violation: MCV(θ^N_sp). (Eq. (21))
        Solve the DFO model and store δ_sp.
    end
end
The solution of the design centering problem is θ^{N*}_sp such that max{δ_sp, ∀sp ∈ S'}.
```

6. Case studies

Four cases are presented to illustrate the flexibility index and design centering methods. Pyomo (Python-based open-source software package)[30,31] is applied to define the models. The MINLP model can be automatically deduced within the function module. The GAMS solver, BARON, is called to solve the MINLP model through the interface of Pyomo and GAMS. Rectangular and elliptical sets for the process parameters are considered in each case.

6.1. Linear case

Consider the following linear inequalities,

$$g_1: \theta_2 - \theta_1 \leq 0$$
$$g_2: -\theta_2 - \frac{\theta_1}{3} + \frac{4}{3} \leq 0$$
$$g_3: \theta_2 + \theta_1 - 4 \leq 0$$

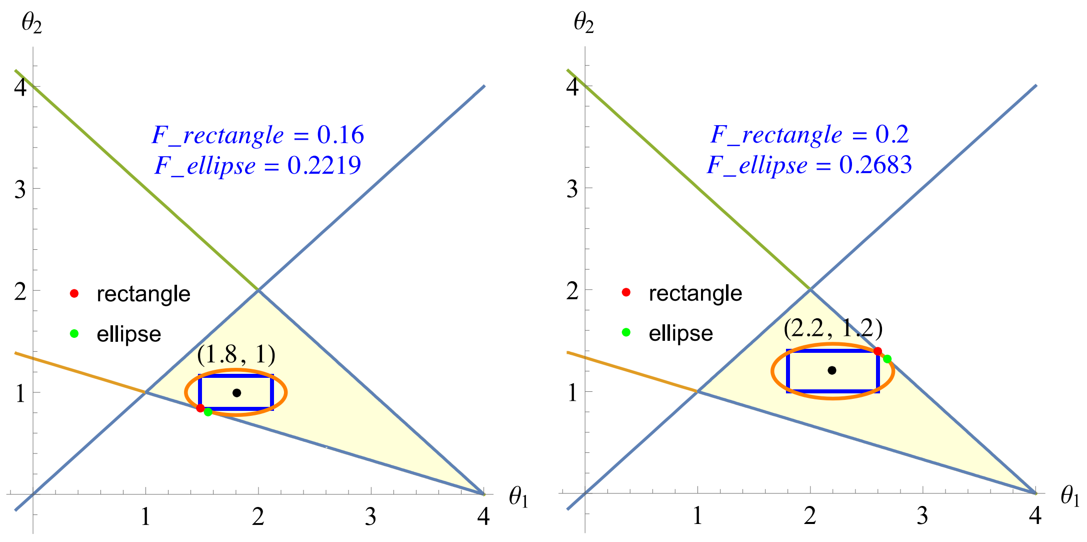
For the flexibility index problem, θ_1 and θ_2 are regarded as process parameters. The feasible region is shown in yellow, and the nominal point of (θ_1,θ_2) is specified as (1.8, 1), as shown in Figure 4(a). The rectangle which is used for direction search is defined as

$$-2 \le \tilde{\theta}_1 \le 2$$
$$-1 \le \tilde{\theta}_2 \le 1$$

The formulation of the ellipse is defined as

$$\left(\frac{\tilde{\theta}_1}{2}\right)^2 + \left(\frac{\tilde{\theta}_2}{1}\right)^2 = 1$$

According to Eq. (15), the MINLP model can be implemented in Pyomo and solved by BARON. For the rectangle case, the result of flexibility index is F = 0.16, the direction that can find the active constraint, i.e., $(\tilde{\theta}_1,\tilde{\theta}_2)$, is $(-2,-1)$, which corresponds to a vertex of the rectangle, and the critical point of (θ_1,θ_2) is (1.48, 0.84). For the ellipse case, F = 0.2219, the critical point is (1.5538, 0.8153), and the corresponding direction is $(-1.11,-0.83)$. In order to further test the proposed flexibility index formulation, the nominal point is specified as (2.2, 1.2). Through solving the MINLP model, for the rectangle and ellipse cases, the results of flexibility index are F = 0.2 and F = 0.2683, respectively. As shown in Figure 4(b), the critical points are (2.6, 1.4) and (2.68, 1.32), which are located at g_3. The corresponding directions $(\tilde{\theta}_1,\tilde{\theta}_2)$ are (2, 1) and $(1.7888,\ 0.4472)$, respectively.



(a) (b)

Figure 4. Flexibility index of the linear example.

For this linear case, both of the proposed design centering methods can be used for solving the design centering problem. Method 1 requires simultaneously evaluating the feasibility over all vertex directions, i.e., all four vertices are restricted within the design space.

$$
\begin{aligned}
&\max_{\delta\in\mathbb{R}^+,\theta^N} \delta \\
s.t.\ & g_1^{LL}: (\theta_2^N - \delta\cdot\Delta\theta_2^-) - (\theta_1^N - \delta\cdot\Delta\theta_1^-) \le 0 \\
& g_2^{LL}: -(\theta_2^N - \delta\cdot\Delta\theta_2^-) - \frac{(\theta_1^N - \delta\cdot\Delta\theta_1^-)}{3} + \frac{4}{3} \le 0 \\
& g_3^{LL}: (\theta_2^N - \delta\cdot\Delta\theta_2^-) + (\theta_1^N - \delta\cdot\Delta\theta_1^-) - 4 \le 0 \\
& g_1^{LU}: (\theta_2^N + \delta\cdot\Delta\theta_2^+) - (\theta_1^N - \delta\cdot\Delta\theta_1^-) \le 0 \\
& g_2^{LU}: -(\theta_2^N + \delta\cdot\Delta\theta_2^+) - \frac{(\theta_1^N - \delta\cdot\Delta\theta_1^-)}{3} + \frac{4}{3} \le 0 \\
& g_3^{LU}: (\theta_2^N + \delta\cdot\Delta\theta_2^+) + (\theta_1^N - \delta\cdot\Delta\theta_1^-) - 4 \le 0 \\
& g_1^{UL}: (\theta_2^N - \delta\cdot\Delta\theta_2^-) - (\theta_1^N + \delta\cdot\Delta\theta_1^+) \le 0 \\
& g_2^{UL}: -(\theta_2^N - \delta\cdot\Delta\theta_2^-) - \frac{(\theta_1^N + \delta\cdot\Delta\theta_1^+)}{3} + \frac{4}{3} \le 0 \\
& g_3^{UL}: (\theta_2^N - \delta\cdot\Delta\theta_2^-) + (\theta_1^N + \delta\cdot\Delta\theta_1^+) - 4 \le 0 \\
& g_1^{UU}: (\theta_2^N + \delta\cdot\Delta\theta_2^+) - (\theta_1^N + \delta\cdot\Delta\theta_1^+) \le 0 \\
& g_2^{UU}: -(\theta_2^N + \delta\cdot\Delta\theta_2^+) - \frac{(\theta_1^N + \delta\cdot\Delta\theta_1^+)}{3} + \frac{4}{3} \le 0 \\
& g_3^{UU}: (\theta_2^N + \delta\cdot\Delta\theta_2^+) + (\theta_1^N + \delta\cdot\Delta\theta_1^+) - 4 \le 0
\end{aligned}
$$

where $\Delta\theta_1^{\mp}$ and $\Delta\theta_2^{\mp}$ correspond to 2 and 1, respectively. According to Eq. 18, the flexibility index, F = 0.2857, can be obtained. Figure 5 shows the result for the selected design center $\theta_1 = 2$, $\theta_2 = 1.1429$.

In order to execute the LHS method for Method 2, the sampling ranges are set as [0, 4] and [0, 2] for θ_1 and θ_2, respectively. A total of four points are sampled, and three of them are feasible, which are listed in Table 1. The DFO method with multiple starting points is applied to solve the design centering problem. Three feasible points are selected as initial values for the DFO solver, and three results of design centering are obtained. The results for the rectangle and ellipse cases are also listed in Table 1. The result corresponding to the largest flexibility index defines the optimal nominal point. Figure 6 indicates the correctness of the results.

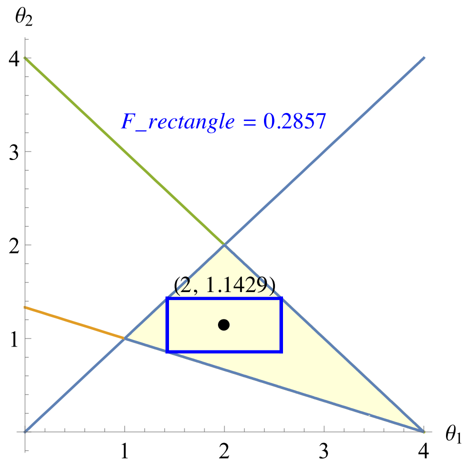
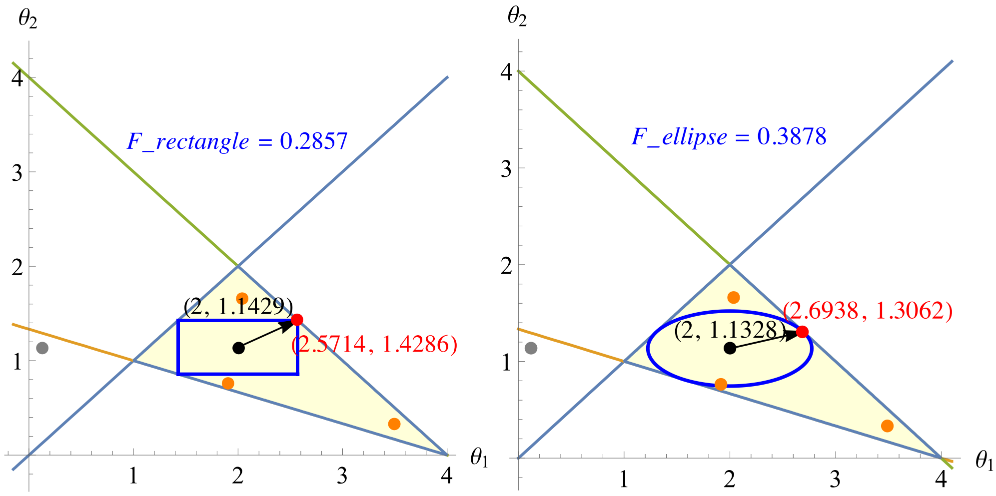


Figure 5. Linear example of design centering using Method 1.



(a) (b)

Figure 6. Linear example of design centering using Method 2.

6.2. Nonlinear case

To further test the performance of the proposed method, the following nonlinear and nonconvex example is considered:

$$g_1: (\theta_2 - 2)^2 + (\theta_1 - 2)^3 + (\theta_2 - 2)(\theta_1 - 2) - \frac{1}{2} \leq 0$$
$$g_2: (\theta_2 - 2)^2 + (\theta_1 - 2)^2 - 2 \leq 0$$

The formulations of the rectangle and ellipse are defined similarly as in the above linear case. The nominal points of (θ_1,θ_2) are set to two different points $(1.5, 1.7)$ and $(2.1, 1.7)$. Figure 7(a) shows that, for the nominal point $(1.5, 1.7)$, the results of flexibility index are F = 0.2760 and $F = 0.2771$. The corresponding critical points are $(1.5619, 1.4240)$ and $(1.5396, 1.4236)$, and the directions $(\tilde{\theta}_1, \tilde{\theta}_2)$ are $(0.2244, -1)$ and $(0.1428, -0.9974)$, respectively. Similarly, for the second nominal point $(2.1, 1.7)$, as shown in Figure 7(b), the obtained flexibility indices are $F = 0.2760$ and $F = 0.3507$, the critical points are $(1.5622, 1.4240)$ and $(1.7513, 1.3958)$, and the corresponding directions are $(-1.9489, -1)$ and $(-0.9945, -0.8676)$, respectively. Figure 7 indicates that the proposed method is also effective for the nonconvex cases. The proposed flexibility index model does not require the Haar condition, because the directions are searched along the boundary of the given rectangle or ellipse, which means that it finds the active constraints directly.

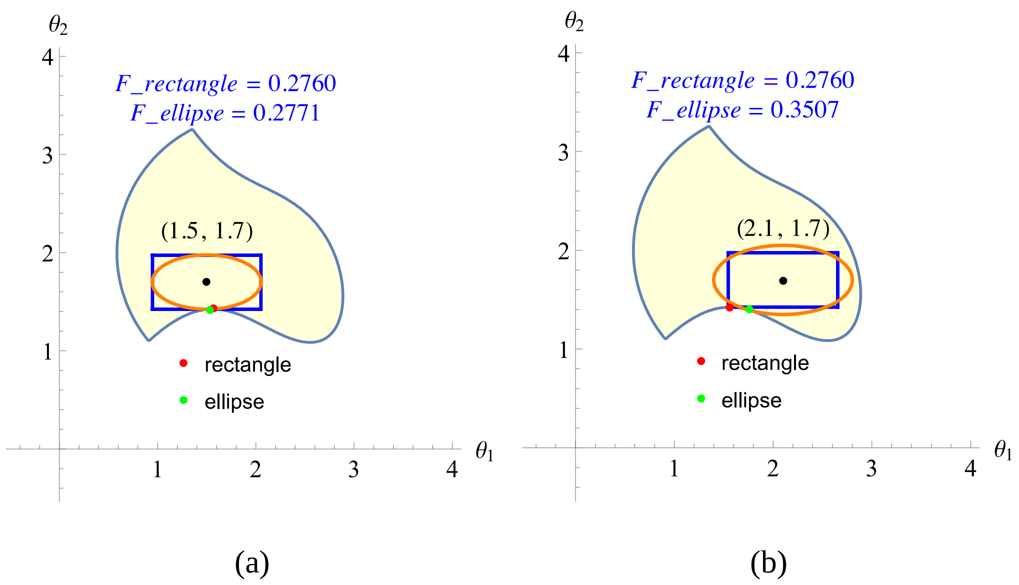
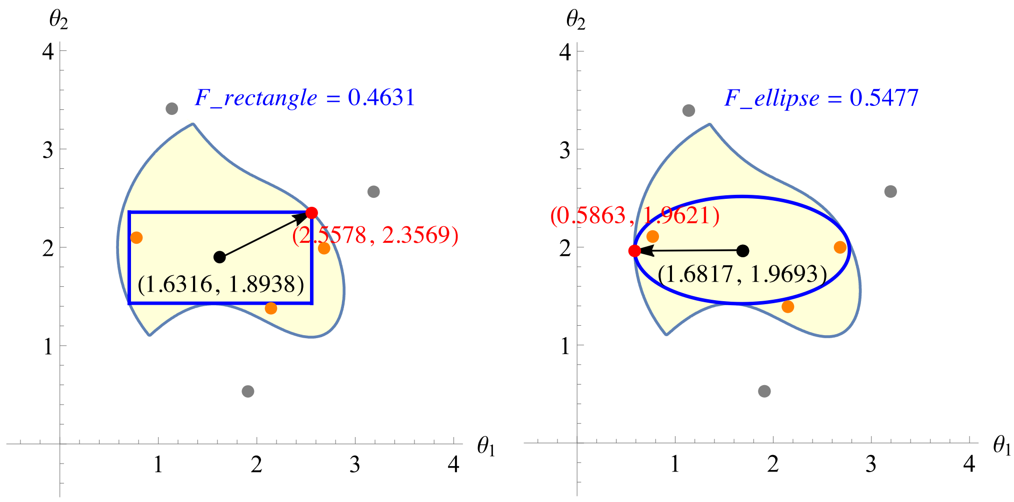


Figure 7. Flexibility index of the nonlinear example.

Similarly, for the design centering problem, the sampling ranges are set to [0.5, 3.5] and [0.5, 3.5] for θ_1 and θ_2, respectively. A total of six points are sampled, and three of them are feasible, which are listed in Table 1. After setting each feasible point as the initial values for the DFO solver, the results of the design centering problem can be obtained. The results for the rectangle and ellipse corresponding to the largest flexibility index are shown in Table 1. Figure 8 can verify the correctness of the results. Table 1 lists the the computational times of these two examples for Method 2. The results of the flexibility index and design centering problems show that the flexibility index of ellipse region is larger than the one of rectangle region.



(a) (b)

Figure 8. Nonlinear example of design centering using Method 2.

Table 1. Results of design centering for Method 2.

		3 feasible starting points for DFO solver (in 10 LHS points)	Rectangle		Ellipse	
			Flexibility index	Nominal point	Flexibility index	Nominal point
Linear example	1	(3.4952, 0.3313)	**0.2857**	**(2, 1.1429)**	**0.3878**	**(2, 1.1328)**
	2	(2.0344, 1.6559)	0.2857	(2, 1.1429)	0.3878	(2, 1.1328)
	3	(1.9093, 0.7600)	0.2857	(2, 1.1429)	0.3878	(2, 1.1326)
		Time (s)	*88.41*		*69.77*	
Nonlinear example	1	(2.6784, 1.9934)	0.2169	(2.2730, 1.9511)	0.5459	(1.6813, 1.9771)
	2	(2.1405, 1.3861)	0.3057	(2.1614, 1.7413)	**0.5477**	**(1.6817, 1.9693)**
	3	(0.7713, 2.0994)	**0.4631**	**(1.6316, 1.8938)**	0.5477	(1.6815, 1.9703)
		Time (s)	*50.85*		*74.46*	

6.3. CSTR reaction

This case is a 2-step consecutive reaction. The process is described by the mechanism of reaction provided by Chen *et al.*[8,32]

$$A + B \overset{k_1}{\to} C, \qquad r_1 = k_1 \cdot c_A \cdot c_b$$
$$C \overset{k_2}{\to} D + E, \qquad r_2 = k_2 \cdot c_c$$

where r_j are the reaction rates. Two process parameters are residence time, τ, and the ratio of the concentration of B to A, R. k_j are model parameters, which are fixed as their mean value {0.31051, 0.026650}. The feasible range of τ and R are described as follows.

$$0 \le \tau \le 550$$
$$0 \le R \le 6$$

The mass balance of the CSTR is given by the following set of equations.

$$c_A^0 - c_A + \tau \cdot (-r_1) = 0$$
$$c_B^0 - c_B + \tau \cdot (-r_1) = 0$$
$$c_C^0 - c_C + \tau \cdot (r_1 - r_2) = 0$$
$$c_D^0 - c_D + \tau \cdot r_2 = 0$$
$$c_E^0 - c_E + \tau \cdot r_2 = 0$$

where c_i^0 are the initial concentrations $\{c_A^0 = 0.53, c_B^0 = 0.53 \cdot R, c_C^0 = 0, c_D^0 = 0, c_E^0 = 0\}$. The quality specifications are minimum yield of product D and minimum ratio of D to unreacted species, that is,

$$\frac{c_D}{c_A^0 - c_A} \ge 0.9$$
$$\frac{c_D}{c_A + c_B + c_C} \ge 0.2$$

Before calculating the feasible operating region over the process parameters, the formulations of the rectangle and ellipse are defined by using the entire given ranges.

$$-275 \le \tilde{\tau} \le 275$$
$$-3 \le \tilde{R} \le 3$$

$$\left(\frac{\tilde{\tau}}{275}\right)^2 + \left(\frac{\tilde{R}}{3}\right)^2 = 1$$

Table 2. Results of flexibility index for CSTR reaction.

Nominal points		Flexibility index	
		Rectangle	Ellipse
1	(527, 2.4)	0.7366	0.7395

2	(444, 3.8)	0.5246	0.5683
3	(350, 4.2)	0.3913	0.3938

Table 3. Results of design centering of multiple starting points for CSTR reaction.

7 feasible starting points for DFO solver		Rectangle		Ellipse	
		Flexibility index	Nominal point	Flexibility index	Nominal point
1	(526.9448, 2.4281)	0.8639579303	(543.7643, 2.7819)	**0.8998934294**	(530.4755, 2.8811)
2	(382.2102, 1.3103)	0.8639492781	(414.0928, 2.7819)	0.8828508761	(428.8189, 2.8335)
3	(495.1392, 2.3676)	0.8639579283	(479.5287, 2.7819)	0.8946968608	(494.9951, 2.8671)
4	(444.4222, 3.7893)	0.8639579263	(430.0968, 2.7819)	0.8732584496	(388.0840, 2.8068)
5	(482.5902, 3.5384)	0.8639579333	(448.9690, 2.7819)	0.8880814238	(455.3512, 2.8481)
6	(419.0243, 0.2167)	**0.8639579423**	(437.0818, 2.7819)	0.8856877694	(442.7762, 2.8415)
7	(350.5285, 4.2073)	0.8639579370	(368.1792, 2.7819)	0.8674989006	(368.8375, 2.7936)

Table 4. Final results of design centering for CSTR reaction.

	Rectangle	Ellipse
Flexibility index	**0.8639579423**	**0.8998934294**
Nominal point	**(437.0818, 2.7819)**	**(530.4755, 2.8811)**
Critical point	(337.7111, 5.3738)	(512.9353, 5.5740)
Critical direction	(-115.0181, 3.0)	(-19.4914, 2.9925)
Feasible region	τ: [199.4934, 674.6703] R: [0.1901, 5.3738]	$((\tau\text{-}530.4755)/275)^2 +$ $((R\text{-}2.8811)/3)^2 \leq 0.8998934294^2$
Time (s)	*168.17*	*187.2*

In order to test the flexibility index problem, as shown in Table 2, three different feasible points are chosen as nominal points. The results show that, for the same nominal point, the flexibility index of ellipse feasible region is larger than the one of rectangle feasible region. For the design centering problem, the sampling ranges are set as [0, 550] and [0, 6], respectively. With the LHS method, 20 points are sampled, which 7 of them correspond to feasible operating points, listed in Table 3. Each feasible point is set as an initial value for the DFO solver, and the results of the rectangle and ellipse cases for design centering problem can be obtained. The result corresponding to the largest flexibility index is the final optimal nominal point. The final critical point, critical direction, feasible region and computational time are also summarized in Table 4. Moreover, note that the given ranges of τ and R are [0, 550] and [0, 6], respectively, which are used to formulate the rectangle and ellipse for direction search. The obtained feasible regions shown in Table 4 indicate that, for the feasible range of τ, [199.4934, 674.6703], the upper bound is above 550, which means that range of τ should be updated, and it can be feasible in a larger scope.

6.4. Michael Addition Reaction

This case is a Michael Addition Reaction with kinetics[8] described in the following equations.

$$AH + B \xrightarrow{k_1} A^- + BH^+, \qquad r_1 = k_1 \cdot c_{AH} \cdot c_B$$

$$A^- + C \xrightarrow{k_2} AC^-, \qquad r_2 = k_2 \cdot c_{A^-} \cdot c_C$$

$$AC^- \xrightarrow{k_3} A^- + C, \qquad r_3 = k_3 \cdot c_{AC^-}$$

$$AC^- + AH \xrightarrow{k_4} A^- + P, \qquad r_4 = k_4 \cdot c_{AC^-} \cdot c_{AH}$$

$$AC^- + BH^+ \xrightarrow{k_5} P + B, \qquad r_5 = k_5 \cdot c_{AC^-} \cdot c_{BH^+}$$

where r_i are reaction rates; AH (Michael donor) and C (Michael acceptor) are starting materials; B is a base; $BH+$, A^- and AC^- are reaction intermediates; P is the product; the rate constants k_i are model parameters, fixed at their mean value: [49.7796, 8.9316, 1.3177, 0.3109, 3.8781]. The CSTR mass balance over the reactions are described as follows.

$$c_{AH}^0 - c_{AH} + \tau \cdot (-r_1 - r_4) = 0$$

$$c_B^0 - c_B + \tau \cdot (-r_1 + r_5) = 0$$

$$c_C^0 - c_C + \tau \cdot (-r_2 + r_3) = 0$$

$$c_{A^-}^0 - c_{A^-} + \tau \cdot (r_1 - r_2 + r_3 + r_4) = 0$$

$$c_{AC^-}^0 - c_{AC^-} + \tau \cdot (r_2 - r_3 - r_4 - r_5) = 0$$

$$c_{BH^+}^0 - c_{BH^+} + \tau \cdot (r_1 - r_5) = 0$$

$$c_P^0 - c_P + \tau \cdot (r_4 + r_5) = 0$$

Two quality constraints are the conversion of *C* must be greater than 90%, the concentration of AC^- in the outlet must be less than 0.002.

$$\frac{c_C^0 - c_C - c_{AC^-}}{c_C^0} \geq 0.9$$

$$c_{AC^-} \leq 0.002$$

The initial concentrations $\{c_{AH}^0, c_B^0, c_C^0, c_{A^-}^0, c_{AC^-}^0, c_{BH^+}^0, c_P^0\}$ are set to be {0.3955, 0.3955/*R*, 0.25, 0, 0, 0, 0}. The process parameters are the residence time τ and the molar ratio *R*, and the feasible range of τ and *R* are described as follows.

$$400 \leq \tau \leq 1400$$
$$10 \leq R \leq 30$$

The formulations of the rectangle and ellipse are defined by using the entire given ranges.

$$-500 \leq \tilde{\tau} \leq 500$$
$$-10 \leq \tilde{R} \leq 10$$

$$\left(\frac{\tilde{\tau}}{500}\right)^2 + \left(\frac{\tilde{R}}{10}\right)^2 = 1$$

Table 5. Results of flexibility index for Michael addition reaction.

Nominal points		Flexibility index	
		Rectangle	Ellipse
1	(1300, 12)	0.7409177641	0.9915670609
2	(800,15)	0.9835211248	1.0816141062
3	(1000, 20)	0.0823404880	0.4861080749

Table 6. Results of design centering of multiple starting points for Michael addition reaction.

7 feasible starting points for DFO solver		Rectangle		Ellipse	
		Flexibility index	Nominal point	Flexibility index	Nominal point
1	(1358.0814, 18.0937)	**1.0588514558**	**(1400.0, 14.2467)**	1.0604982864	(1400.0, 17.3367)
2	(1094.9277, 14.3677)	0.9409177641	(1400.0, 10.0)	**1.2632318763**	**(1400.0, 10.0)**

3	(1300.2531, 17.8920)	0.9409177641	(1400.0, 10.0)	1.0727437634	(1400.0, 17.1645)
4	(1208.0404, 22.6311)	0.5927054415	(1321.3945, 22.6464)	1.1071006470	(1310.0050, 10.0)
5	(1277.4367, 21.7948)	0.9409177641	(1400.0, 10.0)	**1.2632318763**	**(1400.0, 10.0)**
6	(1161.8625, 10.7225)	0.9409177641	(1400.0, 10.0)	**1.2632318763**	**(1400.0, 10.0)**
7	(1037.3246, 24.0243)	0.0842832149	(1037.4618, 22.0136)	0.5184748470	(942.7930, 19.9183)

Table 7. Final results of design centering for CSTR reaction.

	Rectangle	Ellipse
Flexibility index	**1.0588514558**	**1.2632318763**
Nominal point	**(1400.0, 14.2467)**	**(1400.0, 10.0)**
Critical point	(1025.0418, 24.8352)	(849.2243, 16.1834)
Critical direction	(-354.1178, 10.0)	(-436.0052, 4.8949)
Feasible region	τ: [870.5742, 1929.4257] R: [3.6582, 24.8352]	$((\tau\text{-}1400)/500)^2$ + $((R\text{-}10)/10)^2 \leq 1.2632318763^2$
Time (s)	*2443.6*	*695.92*

As shown in Table 5, three different feasible points are chosen as nominal points. The results of flexibility index also indicate show that, for the same nominal point, the flexibility index of ellipse is larger than the one of rectangle. For the design centering problem, the sampling ranges are set as [400, 1400] and [10, 30], respectively. 20 sampling points are generated, and 7 of them are feasible, which are listed in Table 6. The result corresponding to the largest flexibility index for all the feasible points is the final optimal nominal point. All the results are also summarized in Table 7. Similarly, the obtained feasible rectangle region shown in Table 7 indicates that, for the feasible range of τ, [870.5742, 1929.4257], the upper bound is above the given bound of 1400, for the feasible range of R, [3.6582, 24.8352], the lower bound is below the given one of 10. which means that the feasible region actually has a larger scope.

7. Conclusions

In this study, we propose a novel bi-level optimization formulation of flexibility index based on a direction search method, which can be applied to any shape of feasible operating region. For simplicity, only rectangle and ellipse representations of the process parameter set were illustrated. Through the KKT conditions, the flexibility index problem can be transformed into a single-level optimization model. For design centering problems, we propose two methods with different levels of complexity. The vertex direction search method is developed as a single-level optimization model, which can be applicable to a rectangle feasible region for convex models. The DFO method is developed based on the proposed flexibility index model. In order to improve the quality of the solution, the LHS method is used to generate multiple starting points for the DFO solver. The optimal nominal point corresponding to the largest flexibility index can be then determined. The results of the various case studies show that the proposed method is applicable to both convex and nonconvex cases.

Acknowledgements

The authors gratefully acknowledge the financial support from Eli Lilly and Company and the Center for Advanced Process Decision-making (CAPD) from Carnegie Mellon University.

Appendix

For each feasible nominal point, Eq. (15) can give an exact result of flexibility index. A possible strategy to solve Eq. (17) is to reformulate the bi-level optimization problem into a single-level model by applying the KKT conditions to transform Eq. (15) into a MINLP model. Applying this single-level formulation to a linear case, as shown in Figure 9, the final result of flexibility index is F = 1.7778, and the obtained nominal point is (4, 0), which is placed on the boundary. It is obvious that this result is incorrect, because the rectangle lies beyond the feasible region. The reason is that KKT is a necessary rather than a sufficient condition. Eq. (15) is nonconvex, and its KKT condition is not guaranteed to provide a global minimum of δ for each nominal point. Consequently, the outer maximization is operating on an incorrect system.

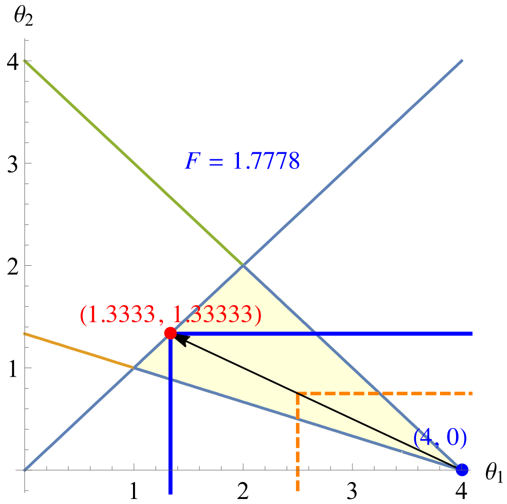


Figure 9. Result of design centering based on KKT reformulation.

Literature Cited

---

1 Lawrence XY, Amidon G, Khan MA, Hoag SW, Polli J, Raju GK, Woodcock J. Understanding pharmaceutical quality by design. *AAPS J*. 2014;16(4):771-783.

2 Kusumo KP, Gomoescu L, Paulen R, García-Muñoz S, Pantelides CC, Shah N, Chachuat B. Bayesian Approach to Probabilistic Design Space Characterization: A Nested Sampling Strategy. *Ind Eng Chem Res*. 2019;59(6):2396-2408.

3 Hakemeyer C, McKnight N, John RS, Meier S, Trexler-Schmidt M, Kelley B, Zettl F, Puskeiler R, Kleinjans A, Lim F, Wurth, C. Process characterization and design space definition. *Biologicals*. 2016;44(5):306-318.

4 Garcia-Munoz S, Luciani CV, Vaidyaraman S, Seibert KD. Definition of design spaces using mechanistic models and geometric projections of probability maps. *Org Process Res Dev*. 2015;19(8):1012-1023.

5 Goyal V, Ierapetritou MG. Determination of operability limits using simplicial approximation. *AIChE J*. 2002;48(12):2902-2909.

6 Kucherenko S, Giamalakis D, Shah N, García-Muñoz S. Computationally efficient identification of probabilistic design spaces through application of metamodeling and adaptive sampling. *Comput Chem Eng*. 2020;132:106608.

7 Zhao F, Chen X. Analytical and triangular solutions to operational flexibility analysis using quantifier elimination. *AIChE J*. 2018;64(11):3894-3911.

8 Laky D, Xu S, Rodriguez JS, Vaidyaraman S, García-Muñoz S, Laird C. An optimization-based framework to define the probabilistic design space of pharmaceutical processes with model uncertainty. *Processes*, 2019;7(2):96.

9 Halemane KP, Grossmann IE. Optimal process design under uncertainty. *AIChE J*. 1983;29(3):425-433.

10 Swaney RE, Grossmann IE. An index for operational flexibility in chemical process design. Part I: Formulation and theory. *AIChE J*. 1985;31(4):621-630.

11 Swaney RE, Grossmann IE. An index for operational flexibility in chemical process design. Part II: Computational algorithms. *AIChE J*. 1985;31(4):631-641

12 Grossmann IE, Calfa BA, Garcia-Herreros P. Evolution of concepts and models for quantifying resiliency and flexibility of chemical processes. *Comput Chem Eng*. 2014;70:22-34.

13 Ochoa MP, Grossmann IE. Novel MINLP formulations for flexibility analysis for measured and unmeasured uncertain parameters. *Comput Chem Eng*. 2020;135:106727.

14 Grossmann IE, Floudas CA. Active constraint strategy for flexibility analysis in chemical processes. *Comput Chem Eng.* 1987;11(6): 675-693.

15 Li J, Du J, Zhao Z, Yao P. Efficient method for flexibility analysis of large-scale nonconvex heat exchanger networks. *Ind Eng Chem Res*. 2015;54(43):10757-10767.

16 Pulsipher JL, Zavala VM. A mixed-integer conic programming formulation for computing

the flexibility index under multivariate gaussian uncertainty. *Comput Chem Eng*. 2018;119: 302-308.

17 Pulsipher JL, Rios D, Zavala VM. A computational framework for quantifying and analyzing system flexibility. *Comput Chem Eng*. 2019;126:342-355.

18 Straub DA, Grossmann IE. Integrated stochastic metric of flexibility for systems with discrete state and continuous parameter uncertainties. *Comput Chem Eng*. 1990;14(9):967-985.

19 Director S, Hachtel G. The simplicial approximation approach to design centering. *IEEE Trans Circuits Syst*.1977;24(7):363-372.

20 Still G. Generalized semi-infinite programming: Theory and methods. *Eur J Oper Res*. 1999;119(2):301-313.

21 Vázquez FG, Rückmann JJ, Stein O, Still G. Generalized semi-infinite programming: a tutorial. *J Comput Appl Math*. 2008;217(2):394-419.

22 Stein O. A semi-infinite approach to design centering. In Optimization with multivalued mappings. Springer, Boston. 2006.

23 Harwood SM, Barton PI. How to solve a design centering problem. *Math Methods Oper Res*. 2017;86(1):215-254.

24 Colson B, Marcotte P, Savard G. An overview of bilevel optimization. Ann Oper Res. 2007;153(1):235-256.

25 Fiacco AV. Penalty methods for mathematical programming in E^n with general constraint sets. *J Optim Theory Appl*. 1970;6(3):252–268.

26 Larson J, Menickelly M, Wild SM. Derivative-free optimization methods. *Acta Numerica*, 2019;28:287-404.

27 Rios LM, Sahinidis NV. Derivative-free optimization: a review of algorithms and comparison of software implementations. *J Glob Optim*. 2013;56(3):1247-1293.

28 Powell MJD. The BOBYQA Algorithm for Bound Constrained Optimization Without

Derivatives. Technical report, Department of Applied Mathematics and Theoretical Physics, University of Cambridge. 2009.

29 Cartis C, Roberts L, Sheridan-Methven O, Escaping local minima with derivative-free methods: a numerical investigation. Technical report, University of Oxford, 2018.

30 Hart WE, Laird CD, Watson JP, Woodruff DL, Hackebeil GA, Nicholson BL, Siirola JD. Pyomo-optimization modeling in Python. Second Edition. Springer, 2017.

31 Hart WE, Watson JP, Woodruff DL. Pyomo: modeling and solving mathematical programs in Python. *Math Program Comput.* 2011;3(3):219-260.

32 Chen W, Biegler LT, García-Muñoz S. An approach for simultaneous estimation of reaction kinetics and curve resolution from process and spectral data. *J Chemom*. 2016;30(9):506-522.