Received: 13 August 2020 | Revised: 23 November 2020 | Accepted: 20 December 2020
DOI: 10.1002/aic.17175

PROCESS SYSTEMS ENGINEERING

AIChE JOURNAL

A generalized cutting-set approach for nonlinear robust optimization in process systems engineering

Natalie M. Isenberg[1] | Paul Akula[2] | John C. Eslick[3] | Debangsu Bhattacharyya[2] | David C. Miller[3] | Chrysanthos E. Gounaris[1]

[1]Department of Chemical Engineering, Carnegie Mellon University, Pittsburgh, Pennsylvania

[2]Department of Chemical and Biomedical Engineering, West Virginia University, Morgantown, West Virginia

[3]The National Energy Technology Laboratory, Pittsburgh, Pennsylvania

Correspondence
Chrysanthos E. Gounaris, Carnegie Mellon University, Department of Chemical Engineering, Pittsburgh, PA 15212.
Email: gounaris@cmu.edu

Funding information
U.S. Department of Energy's Office of Fossil Energy, Grant/Award Number: Institute for the Design of Advanced Energy System

Abstract

We propose a novel computational framework for the robust optimization of highly nonlinear, non-convex models that possess uncertainty in their parameter data. The proposed method is a generalization of the robust cutting-set algorithm that can handle models containing irremovable equality constraints, as is often the case with models in the process systems engineering domain. Additionally, we accommodate general forms of decision rules to facilitate recourse in second-stage (control) variables. In particular, we compare and contrast the use of various types of decision rules, including quadratic ones, which we show in certain examples to be able to decrease the overall price of robustness. Our proposed approach is demonstrated on three process flow sheet models, including a relatively complex model for amine-based CO_2 capture. We thus verify that the generalization of the robust cutting-set algorithm allows for the facile identification of robust feasible designs for process systems of practical relevance.

KEYWORDS
CO_2 capture, cutting-set algorithm, process systems engineering, robust optimization

1 | INTRODUCTION

Data in mathematical optimization models are often subject to some level of uncertainty. The latter can originate from measurement errors and the use of empirical data, economic stochasticity (e.g., market prices), or variations in the process environment (e.g., feedstock quality). For chemical process models, uncertainty most often originates from a lack of knowledge regarding underlying physical properties, such as thermodynamic and kinetic properties, or constants associated with the prevailing heat and mass transport phenomena. In the context of process systems engineering, where critical design and control decisions are made by solving models that are subject to such uncertainties, it is especially important to understand the effects of parametric uncertainties on the performance of the chosen solutions, and if significant, to mitigate uncertainty during the optimization phase so that any resulting design is safely and robustly implemented.

Due to the ubiquitous existence of uncertainty in process systems engineering models, there exists a breadth of literature in developing and applying risk-averse optimization approaches. Early work in the field introduced two-stage nonlinear programming formulations with bounded uncertain parameters,[1] two-stage stochastic programming approaches,[2,3] chance-constrained optimization,[4] and flexibility analysis formulations and algorithms[5-7] for handling process systems engineering design under uncertainty. More recently, work by Li and Grossmann[8] proposed a novel algorithmic approach for solving convex, nonlinear stochastic programming problems with mixed-integer recourse with applications in batch plant design and planning with uncertainties in demands and prices. Kelley et al[9] developed a framework to account for uncertainty via chance-constraints in dynamically-constrained scheduling problems, demonstrating this methodology on a complex air separation unit. Finally, Wang et al[10] devised an approach for handling parameter uncertainty in solid–liquid batch reactors wherein worst-case values for parameters in the uncertainty space are iteratively added as scenarios to the optimal control problem.

Within the field of mathematical programming, robust optimization (RO) is a well-established approach for formulating and solving risk-averse models. The vast majority of the RO literature investigates its application on linear and convex models. Specifically, linear and convex RO and adjustable robust optimization (ARO),[11] wherein a subset of variables become "wait-and-see" decisions via functional dependence on the uncertainty, have had much success in identifying robust solutions to a variety of problems such as process scheduling,[12-16] model-predictive control,[17,18] vehicle routing,[19] project scheduling,[20] industrial steam system optimization,[21] and resilient network design,[22] among many other settings. It has also been shown by Zhang et al[23] that there are theoretical similarities between RO and flexibility analysis when applied to linear systems, which highlights the fact that optimization under uncertainty has long been an area of focus within chemical engineering that has led to the development of novel methodologies.

In the chemical process design context, however, models typically possess complex nonlinearities, including many non-convexities originating from physical and chemical equations. These nonlinearities can be in terms of both decision variables and uncertain parameters in the model, meaning that traditional duality-based reformulation methods in the RO literature may lead to either overly conservative or non-robust solutions due to violations of certain underlying assumptions. To address this, Bertsimas et al[24] proposed a local search algorithm for identifying robust feasible solutions to uncertain optimization problems with non-convex inequality constraints. Additionally, there have been recent advances in the development of novel methods and applications of RO methods to nonlinear process systems engineering models, including general nonlinear programming robust counterpart formulations,[25] robust counterparts with local linearization of nonlinear uncertain constraints and a novel sampling algorithm,[26] application to the pooling problem utilizing a cutting-plane solution algorithm,[27] application to water treatment network operation,[28] robust counterpart derivation for the synthesis of fuel refineries under cost uncertainty,[29] and design and operation of process systems with resilience to disruptive events,[30] to name but a few.

A less widely utilized RO solution approach is the robust cutting-set algorithm (RCS), which was first proposed by Mutapcic and Boyd[31] as an adaptation of Kelley's cutting-set approach[32] for application to inequality-only constrained RO problems. In the RCS algorithm, the robust counterpart is solved by iterating between two subproblems, namely the *master* and the *separation* problems. In the master subproblem, optimal designs are identified that are robust against a carefully chosen finite set of uncertain parameter realizations. Then, given a master subproblem solution, the separation subproblem is solved to identify violating parameter realizations that are to be added back to the master problem, with the process repeating until no more violations can be found. This algorithmic solution approach is generally applicable to any continuous optimization problem, so long as the model possesses only inequality constraints and/or equality constraints that can be reformulated via direct state-variable elimination.

We acknowledge that there remains a practical need to develop general RO approaches that can identify robust solutions in nonlinear, non-convex process models that consist mostly of equality constraints or state equations, which cannot be readily simplified or solved out of the formulation. Such constraints ubiquitously arise in process design models due to the extensive use of empirical property correlations and the presence of recycle streams in process flow sheets, among other reasons. For these classes of problems, there is currently no RO solution approach that guarantees robust solutions against the entire uncertainty space. To that end, we propose here an extension to the robust-cutting plane method proposed by Mutapcic and Boyd,[31] which we refer to as the generalized robust cutting-set (GRCS) approach. We aim for the latter to be capable of certifying fully robust solutions to non-convex optimization problems with a large contingent of equality constraints, as well as be valid for models with nonlinearities and non-convexities from both the decision variables and uncertain parameters. The GRCS algorithm has two key features. First, it handles equality constraints systematically and without reformulation. To achieve this, the algorithm sequentially hedges against realizations of parametric uncertainties by maintaining copies of state variables and equations for each added uncertain parameter realization to ensure the feasibility of the master problem state equations. Second, it uses general decision rules (DR), as applied in the area of adjustable RO,[11] in order to handle control variables, which can be thought of as second-stage variables in process design contexts.

The key contributions of this work are thus threefold. First, we provide a formal RO framework in the context of complex, highly non-convex, equality-constrained process design models via the GRCS algorithm. Second, we demonstrate the effective use of nonlinear DR functions in decreasing the adaptivity gap in solving the two-stage problem, that is, increasing second-stage flexibility to approach true two-stage optimality. Finally, we illustrate the tractability of our proposed approach on a number of case studies, including a complex equation-oriented flow sheet model for an amine solvent-based carbon capture process.

The remainder of the paper is structured as follows. In Section 2, we explain the details of our proposed approach, including the problem formulation and solution algorithm. Next, we discuss our implementation in Section 3. In Section 4, we discuss our methodology to properly evaluate the statistical quality of our two-stage robust solutions. We then showcase several case studies in Section 5 to illustrate the performance of the GRCS algorithm on real process systems models, before finally concluding with some remarks in Section 6.

2 | METHODOLOGY

2.1 | The robust counterpart to a process design formulation

We begin the derivation of the robust counterpart with the definition of variables, parameters, and function mappings. For a process optimization model, we define *design* variables $x \in \mathcal{X} \subseteq \mathbb{R}^m$, *control* variables

$z \in \mathbb{R}^n$, *state* variables $y \in \mathbb{R}^a$, and all potentially uncertain input data $q \in \mathbb{R}^w$. The domain of the design variables, $x \in \mathcal{X}$, is defined here abstractly to represent non-uncertain constraints involving just these variables. Most often, this domain incorporates the applicable variable bounds. The design variables (e.g., equipment sizes) are also referred to as first-stage variables, as they have to be committed upon before the true realization of the parameters q is known. In contrast, the control variables (e.g., flow rates, as manipulated via a valve) are also referred to as second-stage variables, as in principle their values can be adjusted after this realization is known. Finally, the state variables y are those second-stage variables that do not constitute degrees of freedom, rather they depend on the values of x, z, and q.

In process design optimization models, the objective function considered is most often an economic one, such as some net present value or some equivalent annual cost (EAC) (or profit). Its general form is shown in Equation (1), where we split the objective into two parts, first-stage (i.e., investment) costs, $f_1(x)$, where $f_1 : \mathbb{R}^m \mapsto \mathbb{R}$, and second-stage (i.e., operational) costs, $f_2(x, z, y, q)$, where $f_2 : \mathbb{R}^{m+n+a} \mapsto \mathbb{R}$. Here, ζ represents the objective function value, which is to be minimized.

$$\zeta = f_1(x) + f_2(x,z,y,q) \tag{1}$$

Note how, by definition, the first-stage costs depend exclusively on the design variables and have no dependence on uncertain parameters. At the same time, both first- and second-stage variables are allowed to inform second-stage costs, which represents the most general case. For example, one may choose to not explicitly model the design of a pump, thus eliminating the freedom to control a given flow rate. However, the state variable for that flow rate, which is evaluated at a particular design and control setting, could still induce an operational cost. There is also clear motivation to allow first-stage design variables to effect second-stage costs. For example, the height of the distillation column is chosen at the first stage, yet this height factors into calculating power consumption and the corresponding fluid pumping costs at the second stage.

Given q^0 to be a specific realization of the input data q, the deterministic formulation of a generic process design model is shown in Equations (2a)–(2c). Constraints (2b) correspond to a set of inequality constraints $g_i : \mathbb{R}^{m+n+a} \mapsto \mathbb{R}$, $i \in \mathcal{I}$, to which we will be referring to as *performance constraints*. The latter typically express desirable levels of system performance metrics, such as product yields, utilities usage, or safety thresholds, but a few examples. Explicit bounds on the z variables are also considered as part of the g_i constraints, for notational convenience. The equality constraints (2c), $h_j : \mathbb{R}^{m+n+a} \mapsto \mathbb{R}$, $j \in \mathcal{J}$, are a system of state equations that define the state variables y. Note that we pose no restriction on the nature of the objective or constraints, meaning they may be linear or nonlinear (convex or non-convex) in either the model variables, parameters, or both.

$$\min_{\substack{x \in \mathcal{X},\\ z \in \mathbb{R}^n, y \in \mathbb{R}^a}} \quad f_1(x) + f_2\left(x,z,y;q^0\right) \tag{2a}$$

$$\text{s.t. } g_i\left(x,z,y;q^0\right) \le 0 \quad \forall i \in \mathcal{I} \tag{2b}$$

$$h_j\left(x,z,y;q^0\right) = 0 \quad \forall j \in \mathcal{J} \tag{2c}$$

The above formulation is deterministic because it only considers a single value for each parameter in the model. However, if the input parameter data are indeed uncertain, and if that uncertainty is properly characterized, then the above deterministic model can be used as the basis to casting a RO model. More specifically, let us postulate that the uncertain data q may attain values from within a general uncertainty set, $\mathcal{Q} \subset \mathbb{R}^w$. The form of the (often multidimensional) uncertainty set is chosen by the modeler in each case, usually with the help of a suitable parameter estimation method. Typically, the "shape" of the set is such that it captures known correlations among the uncertain parameters, while the "size" of the set is tuned to reflect a desirable confidence interval for their realization. In principle, the uncertainty sets in our framework can take on any form, for example, a continuous convex or non-convex set, or a disjoint set (e.g., a set of discrete points representing scenarios). Without loss of generality, however, in the remainder of this work we will limit our attention to uncertainty sets that are continuous and compact. Convexity is not a necessary property for our sets.

Given uncertain parameters and an associated uncertainty set, $q \in \mathcal{Q}$, Equations (3a)–(3c) represent the robust counterpart formulation of the previously shown deterministic model. This robust counterpart constitutes a two-stage, min-max-min formulation, where decisions x are taken *before* the realized value of the uncertain parameters is known, while decisions z are taken *after* this is the case. Due to their nature as state variables, values for variables y are also chosen *after* the realization of the uncertainty. We assume that all state variables are non-trivial, meaning they are coupled to uncertain parameters q or second-stage variables z and cannot be simply solved out of the model equations.

$$\min_{x \in \mathcal{X}} \max_{q \in \mathcal{Q}} \min_{z \in \mathbb{R}^n, y \in \mathbb{R}^a} f_1(x) + f_2(x,z,y,q) \tag{3a}$$

$$\text{s.t.} \quad g_i(x,z,y,q) \le 0 \quad \forall i \in \mathcal{I} \tag{3b}$$

$$h_j(x,z,y,q) = 0 \quad \forall j \in \mathcal{J} \tag{3c}$$

In its current form, the robust counterpart allows total flexibility for the control variables within the inner minimization problem. To simplify the two-stage robust counterpart, we employ DR, which have been ubiquitously used in the area of ARO to convert second-stage variables into a function of the uncertain parameters q and new first-stage decision variables d (the parameterization of the DR themselves). ARO with affine DR, that is, an affine relationship between uncertain parameters and second-stage variables, was first proposed by Ben-Tal et al.[11] In process systems engineering applications, affine DR have also been used to solve two-stage RO problems in the contexts of water treatment networks[28] and steel-making processes.[33] ARO with generalized affine DR for mixed-integer linear optimization models has also been recently demonstrated by Avraamidou and Pistikopoulos[34] via a multi-parametric programming approach.

We note that the motivation for strictly affine DR to preserve linear model structure does not apply here, as most chemical process models possess nonlinearities. Here, we present a general functional relationship for DR, as shown in Equation (4), to highlight the fact that the proposed approach can admit any general, nonlinear DR function. In this equation, each control variable z_ℓ, $l \in \{1, \ldots, n\}$, has a functional dependence on the uncertain parameters q and the corresponding first-stage variables, d_ℓ. More specifically, $d_\ell \in \mathbb{R}^p$, $\forall \ell \in \{1, \ldots, n\}$, are mutually exclusive subvectors of d that are only referenced in the specific DR function $v_\ell : \mathbb{R}^{p+w} \mapsto \mathbb{R}$ associated with each z_ℓ. In Section 2.3, we will specify the general form of Equation (4) to consider constant, affine, and quadratic DR, which we will later employ in our computational studies.

$$z_\ell = v_\ell(d_\ell, q) \quad \forall \ell \in \{1, \ldots, n\} \tag{4}$$

The application of this general DR relationship modifies the robust counterpart to formulation *RC*, which is shown in Equations (5a)–(5e). Note how the functional dependence of the control policy on the realization of uncertainty, as expressed via the DR, is chosen at the first stage. Furthermore, an auxiliary epigraph variable $\zeta \in \mathbb{R}$ has been incorporated to push the objective function to the set of constraints, for convenience. It is assumed from this point forward that any given values of x, z and q map to a unique value of y. Under this assumption, y are simply evaluated in the maximization step.* Furthermore, the inner minimization problem possesses no degrees of freedom and merely evaluates the (robust) feasibility and (worst-case) objective value resulting from our first-stage decisions.

$$(RC): \min_{\substack{x \in \mathcal{X}, \\ d_\ell \in \mathbb{R}^p \forall \ell}} \max_{\substack{q \in \mathcal{Q}, \\ z \in \mathbb{R}^n, y \in \mathbb{R}^a}} \min_{\zeta \in \mathbb{R}} \zeta \tag{5a}$$

$$\text{s.t.} \quad \zeta \geq f_1(x) + f_2(x, z, y, q) \tag{5b}$$

$$g_i(x, z, y, q) \leq 0 \quad \forall i \in \mathcal{I} \tag{5c}$$

$$\text{s.t.} \quad h_j(x, z, y, q) = 0 \quad \forall j \in \mathcal{J} \tag{5d}$$

$$z_\ell = v_\ell(d_\ell, q) \quad \forall \ell \in \{1, \ldots, n\} \tag{5e}$$

The above model enforces the robust feasibility of the overall design under the postulated control policy, which suffices to qualify the design as robust. However, we remark that the decision maker need not commit a priori to following this policy, as it may lead to overly restrictive control actions in light of certain realizations. In practice, the recourse actions to be followed will be determined a posteriori by solving an optimal control (or operational) optimization problem after the uncertainty parameters have revealed their true values for the operating period of interest. We elaborate further on these issues in Section 4, where we calculate expected second-stage variable values to a posteriori assess the overall performance and cost of the robust designs.

2.2 | The generalized RCS

In this section, we devise a cutting-set based solution approach to address formulation *RC*. The GRCS defines certain subproblems: a master problem and a set of separation problems, one for each constraint (5b) and (5c). As this is an iterative solution approach, the master and separation problems are solved in alternating fashion until converging to the robust solution. To that end, these problems are denoted and indexed as MP_k, SP_i^{perf}, $\forall i \in \mathcal{I}$, and SP^{obj}. Here, k is the iteration index, i is used for the separation problems to denote which performance constraint g_i (out of a total of $|\mathcal{I}|$ such constraints) the problem refers to, while the superscript "obj" implies that the last problem is associated with the epigraph constraint (5b). A flowchart representing the algorithm is shown in Figure 1.

The initial master problem, MP_0, is initialized to be the deterministic model, defined for some nominal value of the uncertain parameters, q^0. In each iteration, the master problem MP_k is solved and requires that the solution is feasible against all realizations q^k, $k \in \mathcal{K}$, that have been identified during the GRCS algorithm's progression. In the separation problems SP_i^{perf}, we search for new realizations of uncertainty that render the master problem solution infeasible, leading to a constraint violation in one or more of the g_i inequality constraints. Then, the parameter realization that corresponds to the largest relative constraint violation, q^*, is identified. If a violation is indeed identified in the current iteration of the algorithm, a full copy of the second stage variables is defined and all constraints (i.e., objective epigraph, performance constraints, state equations, DR), instantiated for the offending realization q^*, are added back to the master problem using these variables (along with the original, common set of first stage variables). Conversely, if there are no violations identified in SP_i^{perf}, for any $i \in \mathcal{I}$, when solved to global optimality, then the current design x^* is deemed robust feasible. At that point, the separation effort could switch focus to solving problem SP^{obj} so as to identify realizations q^* that yield worse objective value than the one currently at hand, in order to reach worst-case optimality.

The formulations and solution approaches for these subproblems will be explained in more detail in the following sections, but before we do so, it is important to discuss convergence properties. We start by noting that Mutapcic and Boyd[31] provide a proof of convergence for their original cutting-set method. In that proof, it is argued that convexity of the model is not required to prove convergence, but it is assumed to ensure tractable subproblem solving steps. Hence, by following a similar proof as in the above reference, we can guarantee convergence in terms of total number of iterations for the GRCS algorithm presented here, as long as any and all master and separation subproblems that arise are tractable. Of course, the latter is an assumption that may or may not hold in the context of process design models of interest to this work, given that the subproblems generated by our algorithm will be non-convex, in general. Indeed, the presence

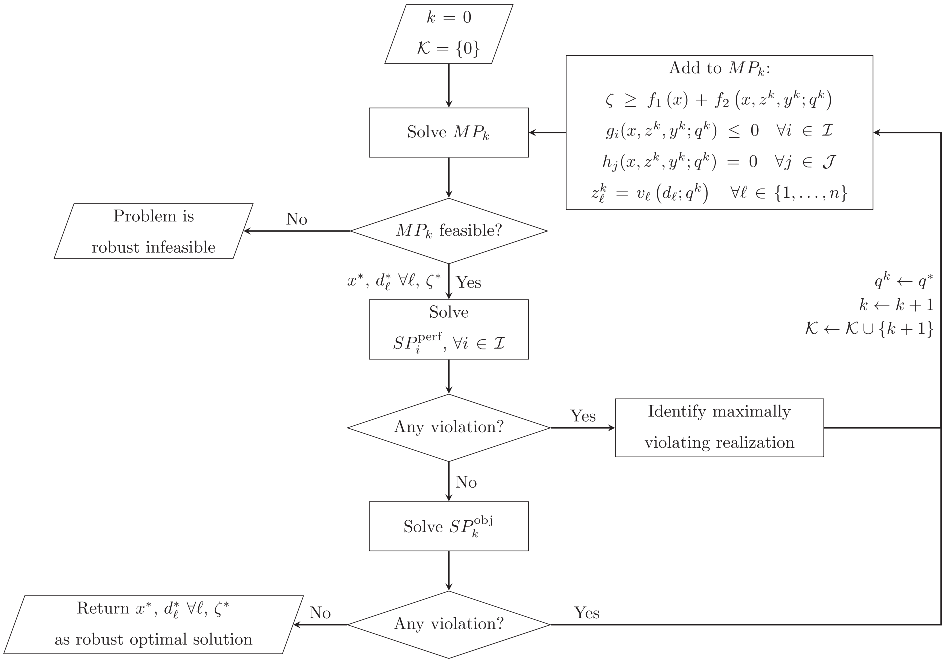

$k = 0$
$\mathcal{K} = \{0\}$
Solve MP_k
Add to MP_k:
$\zeta \geq f_1(x) + f_2(x, z^k, y^k; q^k)$
$g_i(x, z^k, y^k; q^k) \leq 0 \quad \forall i \in \mathcal{I}$
$h_j(x, z^k, y^k; q^k) = 0 \quad \forall j \in \mathcal{J}$
$z^k_\ell = v_\ell(d_\ell; q^k) \quad \forall \ell \in \{1, \ldots, n\}$
Problem is robust infeasible
No
MP_k feasible?
x^*, d^*_ℓ $\forall \ell$, ζ^*
Yes
$q^k \leftarrow q^*$
$k \leftarrow k+1$
$\mathcal{K} \leftarrow \mathcal{K} \cup \{k+1\}$
Solve SP^{perf}_i, $\forall i \in \mathcal{I}$
Any violation?
Yes
Identify maximally violating realization
No
Solve SP^{obj}_k
Return x^*, d^*_ℓ $\forall \ell$, ζ^* as robust optimal solution
No
Any violation?
Yes

FIGURE 1 The generalized robust cutting-set algorithm

of non-linear irremovable state equations h_j, $j \in \mathcal{J}$, in the proposed master and separation problem formulations will generally preempt the non-convexity of the subproblems. From this perspective, if a pathological subproblem is generated during the progression of the GRCS algorithm such that the subordinate (local or global) nonlinear programming solver cannot converge to an optimal solution, then the overall algorithm will also stall. Despite this possibility, however, we should mention that we have had empirical success in using the GRCS algorithm to solve various rather complex process systems models in reasonable time scales, as we later demonstrate in Section 5.

2.2.1 | The master problem

The general form of the master problem is shown in Equations (6a)–(6e). In this formulation, control variables, state variables and uncertain parameters are indexed over a set $\mathcal{K}$, which is the set of iterations of the GRCS algorithm.

$$(MP_k): \quad \min_{\substack{x \in \mathcal{X}, \zeta \in \mathbb{R} \\ d_\ell \in \mathbb{R}^p \forall \ell, \\ z^k \in \mathbb{R}^n, y^k \in \mathbb{R}^a \forall k}} \quad \zeta \tag{6a}$$

$$\text{s.t. } \zeta \geq f_1(x) + f_2\left(x, z^k, y^k; q^k\right) \quad \forall k \in \mathcal{K} \tag{6b}$$

$$g_i\left(x, z^k, y^k; q^k\right) \leq 0 \quad \forall k \in \mathcal{K}, \forall i \in \mathcal{J} \tag{6c}$$

$$h_j\left(x, z^k, y^k; q^k\right) = 0 \quad \forall k \in \mathcal{K}, \forall j \in \mathcal{J} \tag{6d}$$

$$z^k_\ell = v_\ell\left(d_\ell; q^k\right) \quad \forall k \in \mathcal{K}, \forall \ell \in \{1, \ldots, n\} \tag{6e}$$

Note how the constraints in formulation MP_k are cast for all iterations $k \in \mathcal{K}$, meaning that the number of constraints in the master problem increases by $(1 + |\mathcal{J}| + n)$ in each iteration. Furthermore, note how a separate set of state variables y^k has been defined for each uncertain realization q^k that is explicitly referenced in this formulation, which is necessary due to the implicit dependence of variables y on uncertain parameters q in the robust counterpart. Similarly, separate sets of second-stage variables are also considered to reflect the fact that different values (z^k) for the control action might have to be chosen, if a different realization of the uncertain parameters (q^k) prevails. We remark that, unlike the case of implicit state variables, the introduction of separate copies of the control variables in MP_k is not strictly necessary, as in the actual implementation, one may simply substitute variables z^k_ℓ out of the formulation using Equation (6e).

2.2.2 | The separation problems

Each separation problem SP_i^{perf} seeks to identify, if one exists, a realization of the uncertain parameters belonging to the uncertainty set, that is, $q \in \mathcal{Q}$, which renders the optimal design (and associated control policy) from the most recently solved master problem, $MP_{|\mathcal{K}|-1}$, infeasible with respect to performance constraint g_i. The general formulation for these separation problems is shown in Equations (7a)–(7c). A strictly positive optimal objective value in this problem reveals that there exists a realization of the uncertain parameters (the separation problem's optimal solution itself) that makes the master problem solution, (x^*, d^*), infeasible with respect to constraint g_i. Conversely, a non-positive globally optimal value is a certificate that the master solution will satisfy constraint g_i for all realizations in the uncertainty set. Note that a separation problem of this type must be solved successively for each performance constraint g_i, $i \in \mathcal{I}$, in order to verify the overall robustness of the master solution.

$$\left(SP_i^{\text{perf}}\right): \max_{\substack{q \in \mathcal{Q}, \\ z \in \mathbb{R}^n, y \in \mathbb{R}^a}} \quad g_i(z, y, q; x^*) \tag{7a}$$

$$\text{s.t. } h_j(z, y, q; x^*) = 0 \quad \forall j \in \mathcal{J} \tag{7b}$$

$$z_\ell = v_\ell\left(q; d_\ell^*\right) \quad \forall \ell \in \{1, \ldots, n\} \tag{7c}$$

In a similar fashion, the separation problem SP^{obj} seeks to identify, if one exists, a realization of the uncertain parameters that would lead the most recently identified design (and associated control policy) to a worse objective value than the one the master problem solution suggested. The general formulation for this separation problem is shown in Equations (8a)–(8c). The optimal objective value of this problem being strictly positive signifies that there exists a realization of the uncertain parameters (the separation problem's optimal solution itself) that would have evaluated the master problem solution, (x^*, d^*), to a worse objective, and hence, the solution has not been properly adjudicated in terms of its worst-case performance. Conversely, a non-positive globally optimal value is a certificate that this has occurred.

$$\left(SP^{\text{obj}}\right): \max_{\substack{q \in \mathcal{Q}, \\ z \in \mathbb{R}^n, y \in \mathbb{R}^a}} \quad f_1(x^*) + f_2(z, y, q; x^*) - \zeta^* \tag{8a}$$

$$\text{s.t. } h_j(z, y, q; x^*) = 0 \quad \forall j \in \mathcal{J} \tag{8b}$$

$$z_\ell = v_\ell\left(q; d_\ell^*\right) \quad \forall \ell \in \{1, \ldots, n\} \tag{8c}$$

We highlight that the above formulations are parameterized over the set of first-stage design variables and DR, which are fixed to the optimal values from the previous master problem solution, x^* and d^*. At the same time, the uncertain parameters q constitute here decision variables that are free to take on any value in the uncertainty set $\mathcal{Q}$. This form of the separation problem differs from the original cutting-plane algorithm proposed by Mutapcic and Boyd[31] due to the necessity of carrying though the block of state equations h_j (Equations 7b and 8b) to evaluate the state variables y that are associated with the solution of the separation problem. Additionally, DR relationships (Equations 7c and 8c) must also be included in the formulation so as order to evaluate the control actions in the context of the separation problem's optimal solution.

2.3 | Decision rules

In this section, we specify the form of the DR functions v_ℓ we consider in this study. More specifically, we consider three forms that we refer to as the static approximation, affine DR, and quadratic DR. We highlight that Bertsimas et al[35] have explored polynomial DR relationships, including cubic ones, for linear dynamical systems affected by uncertainty. To the best of our knowledge, however, this work is the first to apply nonlinear DR in the context of nonlinear optimization models such as those arising in process design applications. For convenience, we will partition the variables d_ℓ into intercepts, d_ℓ^0, coefficients for linear terms d_ℓ^1, and coefficients for quadratic terms d_ℓ^2 to include both squares and bilinear terms.

$$v_\ell\left(d_\ell^0\right) = d_\ell^0 \quad \forall \ell \in \{1, \ldots, n\} \tag{9}$$

$$v_\ell\left(d_\ell^0, d_\ell^1, q\right) = d_\ell^0 + \sum_{r=1}^{w} d_{\ell,r}^1 q_r \quad \forall \ell \in \{1, \ldots, n\} \tag{10}$$

$$v_\ell\left(d_\ell^0, d_\ell^1, d_\ell^2, q\right) = d_\ell^0 + \sum_{r=1}^{w} d_{\ell,r}^1 q_r + \sum_{r=1}^{w} \sum_{s=r}^{w} d_{\ell,r,s}^2 q_r q_s \quad \forall \ell \in \{1, \ldots, n\} \tag{11}$$

The form of the static approximation is shown in Equation (9). In this case, each control variable is chosen to a single value, and there is no flexibility in the control to respond to the uncertainty, once the latter is revealed. Effectively, the control variables are treated as first-stage decision variables. The affine DR are presented in Equation (10), and they constitute an affine relationship between the (first-stage) DR variables, d, and the uncertain parameters, q. Finally, we show the form for the quadratic DR in Equation (11), which constitutes a more general, nonlinear function that features also square and bilinear terms in the uncertain parameters. The modeler selects whether to utilize the static approximation, affine or quadratic DR in each case. Typically, this selection will be based on experience with the specific application of interest, in terms of the trade-off between computational tractability and extent of (and desire to reduce) the two-stage adaptivity gaps that arise.

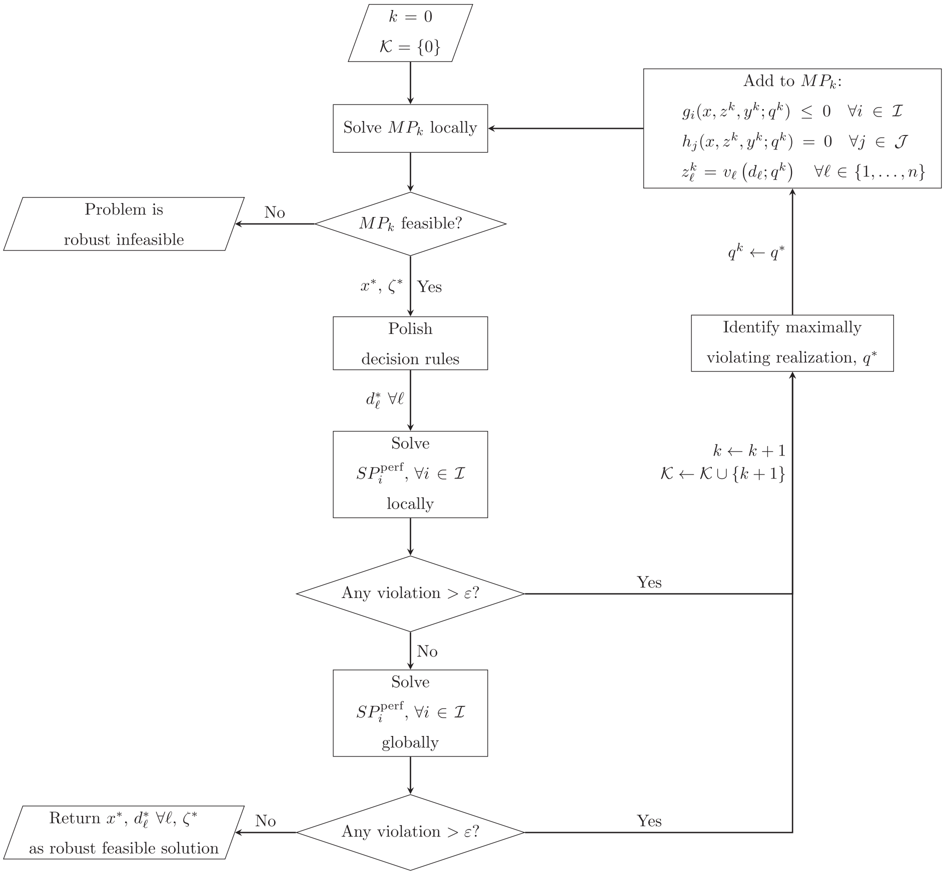
3 | IMPLEMENTATION DETAILS

In this section, we outline various important details regarding our implementation of the proposed GRCS algorithm. For many aspects of the algorithm, we recognize that there exist other options for implementation, and we elaborate on which options we selected in each case. Those choices have been largely informed by their effect on overall tractability, as assessed via the computational case studies we conducted to address various complex process models. Those studies are presented later in Section 5. A modified and more detailed flowchart illustrating the implementation details outlined in this section is presented in Figure 2.

3.1 | Solving master problems

We first note that, in our implementation of the GRCS algorithm, we choose to solve all master problems MP_k locally, using a local nonlinear programming (NLP) solver. This is done given the relative inability of

FIGURE 2 Implementation details of the generalized robust cutting-set algorithm


$k = 0$
$\mathcal{K} = \{0\}$
Solve MP_k locally
Add to MP_k:
$g_i(x, z^k, y^k; q^k) \le 0 \quad \forall i \in \mathcal{I}$
$h_j(x, z^k, y^k; q^k) = 0 \quad \forall j \in \mathcal{J}$
$z_\ell^k = v_\ell(d_\ell; q^k) \quad \forall \ell \in \{1, \dots, n\}$
Problem is robust infeasible
No
MP_k feasible?
$q^k \leftarrow q^*$
x^*, ζ^*
Yes
Polish decision rules
Identify maximally violating realization, q^*
d_ℓ^* $\forall \ell$
Solve SP_i^{perf}, $\forall i \in \mathcal{I}$ locally
$k \leftarrow k + 1$
$\mathcal{K} \leftarrow \mathcal{K} \cup \{k + 1\}$
Any violation $> \varepsilon$?
Yes
No
Solve SP_i^{perf}, $\forall i \in \mathcal{I}$ globally
Return x^*, d_ℓ^* $\forall \ell$, ζ^* as robust feasible solution
No
Any violation $> \varepsilon$?
Yes

today's global NLP solvers to solve to zero gap the complex process design models we are interested in addressing in this study, even in their deterministic version. This shortfall is of course exacerbated by the fact that the master problem size increases in each iteration. The choice to solve master problems locally comes at the expense of not being able to assert traditional *robust optimality* of the final solutions, rather only their *robust feasibility*. We note, however, that the GRCS algorithm could in principle prove robust optimality, provided all master problems are solved globally at each iteration.

In light of the above, it becomes superfluous to seek to solve problems SP^{obj} at the interest of separating realizations that lead to worst-case objectives. Therefore, in this work, we only separate problems SP_i^{perf}, which suffices to guarantee robust feasibility of the final solutions. Furthermore, we target to identify master problem solutions that perform best in the *nominal case*, that is, $q \leftarrow q^0$. This is done by setting the set of realizations associated with constraints (6b) to only include q^0; that is, $\mathcal{K} \leftarrow \{0\}$ for (only) constraints 6b when solving every iteration of the master problem. The choice of the nominal values of the uncertain parameters, q^0, is in principle left for the modeler. Here, we consider q^0 as the most likely realization of q, as determined in expectation, which is also often used in process systems contexts for "deterministic" optimization. In addition to solving master problems targeting to minimize nominal costs, care can be taken to limit the variance of second-stage costs, for example, by including *p-robust constraints*[36] to limit their increase in other, non-nominal scenarios.

3.2 | Separation approach

In order to guarantee the robust feasibility of the final solution determined by the GRCS algorithm, the separation problems must be solved to global optimality for each and every performance constraint. This ensures that there are no realizations of the uncertain parameters (within the uncertainty set) that render the final robust design infeasible. Unlike master problems, the size and dimensionality of separation problems are much smaller, making them tractable for global optimization in this setting. Regardless, the repeated execution of global optimization runs might still add up to significant computational burden. To that end, we choose in our implementation to first solve each separation problem locally. If violating parameters can be identified in this manner, the algorithm can proceed with its next iteration, defining the new master problem based on the local solution of a separation problem. However, whenever the local search returns no violation for each and every performance constraint, we do proceed to solve the separation problem using a global solver, in order to assert robust feasibility. This protocol reduces the overall number of expensive calls to a global optimization solver, and has been found to improve overall algorithm performance.

We now offer some remarks regarding the selection of which violating realization is chosen to iterate the GRCS algorithm when more than one performance constraints g_i exist in a model (i.e., $|\mathcal{J}| > 1$). In such a case, there may be different violating uncertainty realizations identified in separation. Whereas any and all of them could be chosen as critical scenarios against which to explicitly insure feasibility in the subsequent master problem iteration, we recognize that choosing more than one scenario would contribute to rapid increase of the master problem size beyond what is absolutely necessary. Therefore, in our implementation, we choose to only add a single violation, which is selected as follows. Let $\mathscr{I}^V \subseteq \mathscr{I}$ be the subset of performance constraints that can be violated in the context of the most recent master problem solution. More specifically, given ($z^{i,*}$, $y^{i,*}$, $q^{i,*}$) as the optimal solution of problem SP_i^{perf}, we consider $i \in \mathscr{I}^V$ when $g_i(z^{i,*}, y^{i,*}, q^{i,*}; x^*) > \varepsilon$, where $\varepsilon \in \mathbb{R}_+$ is a small tolerance. Once the realizations $q^{j,*}$ that maximally violate each of the constraints $j \in \mathscr{I}^V$ have been determined, we generate a matrix of dimensions $|\mathscr{I}^V| \times |\mathscr{I}^V|$, where each entry $e_{i,j}$ represents the violation of constraint g_i associated with row i under the uncertain parameter realization $q^{j,*}$ associated with column j; that is, $e_{i,j} := \max\{g_i(z^{j,*}, y^{j,*}, q^{j,*}; x^*), 0\}$. The entries are normalized by dividing each of them with the largest value in their row, leading to the maximum entry in each row being equal to one. The maximally violating realization to be added back to the next iteration of the master problem, q^*, is then picked as the one associated with the column possessing the largest sum of entries; that is, $q^* = q^{\gamma,*}$, where $\gamma := \arg\max_{j \in \mathscr{I}^V} \left\{ \sum_{i \in \mathscr{I}^V} e_{i,j} \right\}$.

3.3 | DR polishing

It is a well-known fact that the use of DR often leads to solution degeneracy. In particular, when one solves the master problem at a given iteration k, there may exist many equivalently optimal combinations of DR coefficients d that satisfy Equation (6e). To that end, when using affine and quadratic DR, we augment our implementation with an additional post-processing step after solving the master problem, which we refer to as the *DR polishing* step. The purpose of this step is to comb through the set of equivalently optimal DR and to judiciously pick a specific, desirable policy. In our context, we consider DR coefficient values, $d_\ell \in \mathbb{R}^p$, $\ell \in \{1, \ldots, n\}$, to be more desirable, if their relative magnitude is collectively smaller. To achieve this, we solve the auxiliary optimization problem shown in Equations (12a)–(12h) as soon as an optimal solution x^* with optimal objective value ζ^* is obtained from the master problem in each iteration k.

$$\min_{\substack{d_\ell \in \mathbb{R}^p \forall \ell, \\ \tau_\ell^0 \in \mathbb{R}_+, \tau_\ell^1 \in \mathbb{R}_+^w, \tau_\ell^2 \in \mathbb{R}_+^{w \times w} \forall \ell, \\ z^k \in \mathbb{R}^n, y^k \in \mathbb{R}^a \forall k}} \sum_{l=1}^{n} \left(\tau_l^0 + \sum_{r=1}^{w} \tau_{\ell,r}^1 + \sum_{r=1}^{w} \sum_{s=r}^{w} \tau_{\ell,r,s}^2 \right) \tag{12a}$$

$$\text{s.t.} \quad \zeta^* \geq f_1(x^*) + f_2(z^0, y^0; x^*, q^0) \tag{12b}$$

$$g_i(z^k, y^k; x^*, q^k) \leq 0 \quad \forall k \in \mathcal{K}, \forall i \in \mathscr{I} \tag{12c}$$

$$h_j(z^k, y^k; x^*, q^k) = 0 \quad \forall k \in \mathcal{K}, \quad \forall j \in \mathcal{J} \tag{12d}$$

$$z_\ell^k = v_\ell(d_\ell; q^k) \quad \forall k \in \mathcal{K}, \quad \forall \ell \in \{1, \ldots, n\} \tag{12e}$$

$$-\tau_\ell^0 \leq d_\ell^0 \leq +\tau_\ell^0 \quad \forall \ell \in \{1, \ldots, n\} \tag{12f}$$

$$-\tau_{\ell,r}^1 \leq d_{\ell,r}^1 q_r^0 \leq +\tau_{\ell,r}^1 \quad \forall \ell \in \{1, \ldots, n\}, \quad \forall r \in \{1, \ldots, w\} \tag{12g}$$

$$-\tau_{\ell,r,s}^2 \leq d_{\ell,r,s}^2 q_r^0 q_s^0 \leq +\tau_{\ell,r,s}^2 \quad \forall \ell \in \{1, \ldots, n\}, \quad \forall r,s \in \{1, \ldots, w\} : \{s \geq r\} \tag{12h}$$

In the above formulation, the objective function (12a) minimizes the L_1-norm of the vector of terms appearing in the applicable DR function, where τ are non-negative auxiliary variables introduced to represent the absolute values of each and every such term. Here, we focus the presentation on the case of quadratic DR as per Equation (11), noting that a reduced version of this formulation applicable for the case of affine DR can be obtained by simply fixing all τ_ℓ^2 variables to zero.† We remark that the objective of the above formulation focuses on the DR evaluated under the nominal realization of uncertainty q^0, corresponding to control actions z^0, but other selections (e.g., an average of all realizations q^k) could be readily used instead. Constraint (12b) is added to ensure that we are searching over the set of optimal DR policies, that is, those that induce the same objective value as the master problem solution,‡ while constraints (12c)–(12e) are added to ensure that we are searching over DR policies that remain feasible for the original master problem. Finally, constraints (12f)–(12h) achieve the desired definition for variables τ as the absolute values of the corresponding DR function terms.

4 | EVALUATION OF ROBUST SOLUTION QUALITY

The GRCS is designed to address primarily two-stage decision problems, wherein recourse decisions are permitted to adapt to undesirable deviations in performance caused by parameter uncertainty. These recourse decisions are represented in this presentation by the variables z, where for modeling convenience, we chose to limit them to values attained via DR relationships. In principle, however, an operator making decisions in the second stage is not beholden to the optimal DR functions and has the ability to respond to the actual realization of the uncertainty in an unrestricted fashion. Therefore, from a practical perspective, it is important to quantify the range of possible values for second-stage decision variables and the corresponding expectation and variance of the second-stage costs they might induce. These metrics allow the modeler to assess the accuracy of the DR approximation for a given recourse action, and to explicitly build confidence regarding the overall economic performance of the robust design at hand. In this section, we illustrate how to compute the expected operating costs and expected control variable values, as well as the associated variances, for a given robust feasible design x^*.

To compute these expected values, we solve a set of optimization problems where the control variables are free to take on any feasible value. We start with the deterministic formulation in Equations (2a)–(2c) and fix the first-stage design variables to the robust design, $x \leftarrow x^*$. Then, we assign a randomly sampled value to the uncertain parameters, $q \leftarrow q_s$, resulting in the model shown in Equations (13a)–(13c).

$$\min_{z \in \mathbb{R}^n, y \in \mathbb{R}^a} f_2(z,y;x^*,q_s) \tag{13a}$$

$$\text{s.t. } g_i(z,y;x^*,q_s) \le 0 \quad \forall i \in \mathcal{I} \tag{13b}$$

$$h_j(z,y;x^*,q_s) = 0 \quad \forall j \in \mathcal{J} \tag{13c}$$

This optimization model is then solved for a set $\mathcal{S}$ of uncertainty scenarios, q_s, $s \in \mathcal{S}$, which have been suitably defined to this purpose.[§] In order to determine desirable metrics about their distribution, the optimal solutions, (z^*, y^*), and optimal second-stage costs, $f_2(z^*, y^*; x^*, q_s)$, are thus recorded under each scenario.[¶] More specifically, given probabilities p_s for all sampled scenarios $s \in \mathcal{S}$, Equations (14) and (15) are used to compute the expected operating costs and their *SD*. The expected second-stage costs may be considered when comparing robust feasible designs obtained via the GRCS algorithm to any deterministically optimal design, in order to elucidate the overall cost increase for insuring design robustness.

$$\mathbb{E}[f_2] = \sum_{s \in \mathcal{S}} p_s f_2(z^*, y^*; x^*, q_s) \tag{14}$$

$$\sigma[f_2] = \sqrt{\sum_{s \in \mathcal{S}} p_s \left(f_2(z^*, y^*; x^*, q_s) - \mathbb{E}[f_2] \right)^2} \tag{15}$$

Using similar formulas, the above-described after-the-fact analysis is also useful to understand the range of control variable actions, namely $\mathbb{E}[z_\ell]$ and $\sigma[z_\ell]$ for all $\ell \in \{1, \ldots, n\}$, that shall be required to achieve robust feasibility in the context of a given design. We demonstrate such analyses in the computational case studies we present in the next section.

5 | CASE STUDIES

We present a set of three case studies to illustrate the performance of the GRCS algorithm in identifying robust feasible solutions to nonlinear process optimization problems. In Case Study I, we focus on a reactor–separator process. In Case Study II, we study a reactor–heater process. Finally, in Case Study III, we consider a highly complex, high-fidelity flow sheet model for amine solvent-based CO_2 separation from flue gas. All models were implemented using Pyomo[37,38] and tools from the Institute for the Design of Advanced Energy Systems (IDAES) integrated framework.[39,40] We used IPOPT[41] paired with the HSL MA27[42] linear solver as the local optimization solver. Given the built-in flexibility of our implementation, the global solver Antigone[43] was utilized to confirm robust feasibility in Case Study I, while the global solver BARON[44] was employed for Case Studies II and III. Each case study was solved on a desktop computer featuring four Intel i7-6700 3.4 GHz processors and 16 GB RAM. All solvers were configured with an optimality tolerance of 1×10^{-6}, while the tolerance for identifying normalized violations of performance constraints via solving separation problems was set to 1×10^{-4}. In Case Studies I and II, we showcase the capabilities of all three DR types, as the overall model sizes were amenable to the addition of more complex and nonlinear DR functions. In the final case study, we only consider the static approximation and affine DR policies. All references to sections, equations, tables and figures that begin with the letter “S” refer to the supplementary material that accompanies this manuscript.

5.1 | Case Study I: Reactor–separator

The flow sheet illustrating the reactor–separator system is shown in Figure 3a, and has been previously studied in Grossmann and Sargent,[1] Rooney and Biegler,[45] and Yuan et al.[26] In this design problem, we are modeling the isothermal, liquid-phase conversion of reactant *A* into a desired product *C* via a set of four first-order chemical reactions, which are outlined in Figure 3b. Products *D* and *E* are undesirable side-products in the reaction network.

Using this reaction network information and the process flow sheet, we can cast the deterministic model shown in Section S1.1. The objective function in Equation (S1a) corresponds to the equivalent annualized cost using a capital recovery factor of 0.09, which corresponds to an operating lifetime of 25 years under an 8% annual interest rate.[46] The inequality constraint of Equation (S1h) states that feasible designs must yield at least 40 $\frac{\text{mol}}{\text{hr}}$ of product *C*. An additional inequality constraint, Equation (S1i), limits the amount of byproduct *D* recycled. Both of these inequalities are considered performance constraints that are subject to separation steps in the GRCS algorithm. All other constraints are either state equations or second-stage variable bounds, the latter of which are also subject to separation.

In summary, the deterministic reactor–separator model consists of 9 decision variables, 6 equality constraints, 2 inequality constraints (excluding variable bounds), and 12 parameters of which four will be considered in this work as being uncertain. More specifically, the first-stage decision variable is the volume of the reactor, *V*, while the second-stage control variables are the recycle ratios δ and β, which represent the fractions of *A* and *B*, and *D* and *E* recycled, respectively. The mole fractions (x_a, …, x_e) as well as the flow rate *F* constitute state variables. For this study, the known parameters are the inlet concentration of the reactant $C_{a0} = 10 \frac{\text{mol}}{\text{m}^3}$ and the inlet flow rate of the reactant $F_{a0} = 100 \frac{\text{mol}}{\text{hr}}$. The uncertain data are the reaction rate constants k_i, $i = \{1, 2, 3, 4\}$, which are correlated in a four-dimensional ellipsoidal uncertainty set. The original data for describing this set (mean, *SD*) can be found in Rooney and Biegler[45] and are also reported in Section S1.3, for convenience. Thus, for the application of the GRCS algorithm, $x = (V)$, $z = (\delta, \beta)$, all other variables are state variables *y*, while $q = (k_1, k_2, k_3, k_4)$.

5.1.1 | Case Study I results

Results for the optimal first-stage variables and costs for the deterministic and robust feasible cases are shown in Table 1. The reactor volume, V, identified in the robust solutions is larger than that of the deterministic solution, in order to insure against the postulated uncertainty. This is likely due to the fact that feasible robust solutions must permit sufficient production of product C and sufficient recycle of byproduct D across the range of values for k_i, $i = \{1, 2, 3, 4\}$, which in turn requires increased reactor sizes than the deterministic case. However, it is important to note that, as recourse flexibility increases via the use of more involved DR, the reactor volume as well as the first-stage cost decreases.

The optimal values for second-stage variables and costs for the deterministic and robust feasible cases are shown in Table 2. When we compare the values of the control variables δ and β in the deterministic case against the robust feasible values corresponding to the nominal realization of the uncertainty, we see that the values for the recycle ratio δ increases in the latter case. This leads to an increase in nominal second-stage costs when compared to the deterministic solution. However, if we consider the expected second-stage variable values and costs in Table 3, they more closely match the deterministic case solution. In particular, full-flexibility in second-stage control has a significant economic benefit over more restrictive DR policies. Interestingly, if we pay attention to the expected robust feasible objective values, representing the total expected cost $\mathbb{E}[\zeta] = f_1(x^*) + \mathbb{E}[f_2]$, the affine and quadratic DR solutions result in overall lower costs than the static approximation policy, even when considering the associated variance.

A noteworthy observation in this case study is the fact that $\mathbb{E}[\zeta]_{\text{SA}} > \mathbb{E}[\zeta]_{\text{ADR}} > \mathbb{E}[\zeta]_{\text{QDR}}$. This means that, for this system, there is an economic benefit in utilizing more flexible, nonlinear DR over the traditional affine or inflexible cases. We also note that, when compared to the deterministic optimal EAC, ζ_{det} = 18, 816.59 (\$/year), the robust feasible designs incur in expectation a comparable cost. Indeed, $\mathbb{E}[\zeta]/\zeta_{\text{det}} \approx 1$ and $\sigma[f_2]/\mathbb{E}[\zeta] \approx 0.066$ for all three recourse policies, indicating that all robust designs are likely to yield actual costs that do not significantly deviate from those of the deterministic design.

The total number of iterations of the GRCS and the total CPU time for completing the GRCS algorithm are shown in Table 4. This table also reports the fraction of total time spent on master and separation subproblems via calls to subordinate solvers, as well as the residual fraction spent on code overhead, which primarily includes time for Pyomo model manipulations. Clearly, for this case study, which features a high-dimensional uncertainty set and multiple inequality constraints, the number of iterations increases with affine and quadratic DR. This hints to the likelihood of decreased tractability when more flexible DR are used in conjunction with complex models, though the relatively small size of the underlying model in this case study caused no computational issues. Whereas the total times are generally small, a significant portion is spent in solving separation problems. This is due to the fact that there are multiple such problems to be solved in each iteration, as well as the fact that a number of calls to global solvers are required to confirm robustness.

Additionally, we plot in Figure 4 the GRCS progression toward robust feasibility at each iteration of the GRCS for the static approximation recourse policy. The trajectories of the algorithm under affine and quadratic DR policies are deferred in Figures S1 and S2. Each plot depicts the performance of designs at a given iteration k, as determined from the master problem MP_k. The plots are generated by evaluating each optimal design via the same set $\mathcal{S}$ of 200 uniformly sampled realizations $q_s \in \mathcal{Q}$, $s \in \mathcal{S}$. Feasibility of each design at a realization q_s is determined via one of the two constraints in Equations (S1h) and (S1i). We remark that, although the constraints for bounds on control variables δ and β were included in the set of constraints subject to separation, they never led to a violation across the range of values within the uncertainty set, and are hence not referenced in this analysis. The constraint for which feasibility is shown in a given plot is determined by which constraint yielded a maximum violation in the separation problem $SP_{k,i}$.

For each point in the plots, the solution is either feasible and represented as a blue dot, or is infeasible and represented as a red dot.

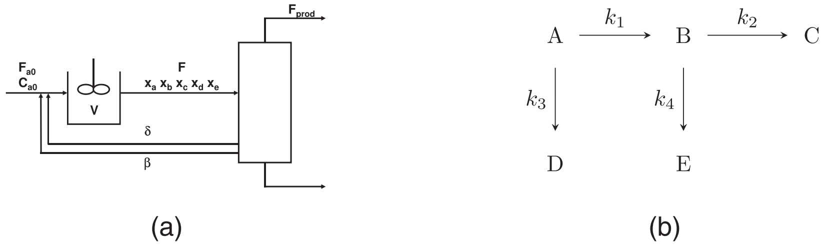


FIGURE 3 Flow sheet (a) and reaction mechanism (b) representing the reactor–separator system considered in Section 5.1, as adapted from Grossmann and Sargent[1]

TABLE 1 Optimal values of first-stage variables and costs for the reactor–separator model

	Deterministic	Static approx.	Affine DR	Quadratic DR
V (m^3)	103.18	105.50	103.56	103.20
$f_1(x^*)$ (\$/year)	9,973.32	10,426.31	10,047.29	9,978.13

Abbreviation: DR, decision rules.

TABLE 2 Optimal values of second-stage variables and costs, under the nominal realization of uncertainty, for the reactor–separator model

	Deterministic	Static approx.	Affine DR	Quadratic DR
δ $\left(\frac{\text{mol A + mol B}}{\text{total mol}}\right)$	0.46	0.68	0.66	0.64
β $\left(\frac{\text{mol D + mol E}}{\text{total mol}}\right)$	0.016	0.017	0.017	0.018
$f_2(z^*, y^*; q^0)$ (\$/year)	8,843.27	13,250.84	12,881.24	12,614.63

Abbreviation: DR, decision rules.

TABLE 3 Expected values and *SD*s of second-stage variables and costs for the reactor–separator model

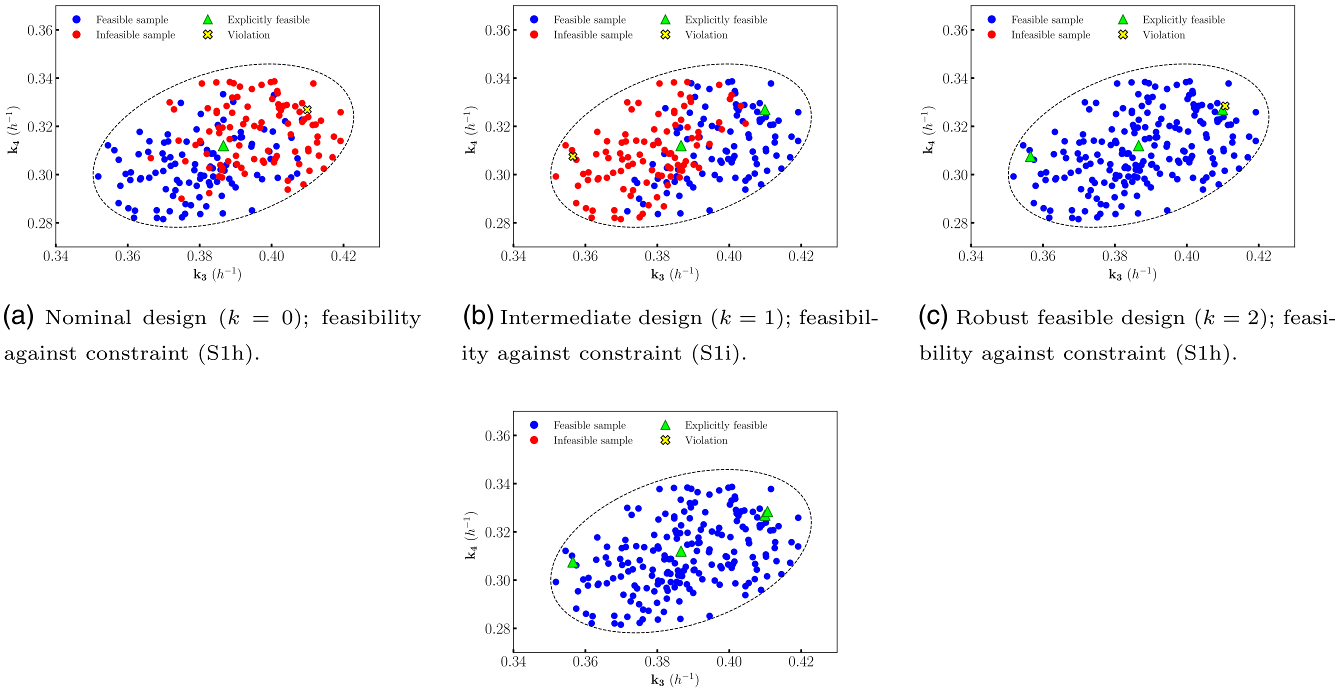
	Static approx.	Affine DR	Quadratic DR
$\mathbb{E}[\delta] \pm \sigma[\delta]$ $\left(\frac{\text{mol A + mol B}}{\text{total mol}}\right)$	0.45 ± 0.07	0.46 ± 0.07	0.46 ± 0.07
$\mathbb{E}[\beta] \pm \sigma[\beta]$ $\left(\frac{\text{mol D + mol E}}{\text{total mol}}\right)$	0.016 ± 0.0004	0.016 ± 0.0004	0.016 ± 0.0004
$\mathbb{E}[f_2] \pm \sigma[f_2]$ (\$/year)	8,395.78 ± 1,240.87	8,758.91 ± 1,245.64	8,829.64 ± 1,246.56
$\mathbb{E}[\zeta] \pm \sigma[f_2]$ (\$/year)	18,822.10 ± 1,240.87	18,806.20 ± 1,245.64	18,805.77 ± 1,246.56

Abbreviation: DR, decision rules.

TABLE 4 Total number of iterations and CPU time spent within the GRCS algorithm when addressing the reactor–separator model

	Static approx.	Affine DR	Quadratic DR
# of GRCS iterations	4	9	9
Total CPU time (s)	10.3	11.4	33.2
% spent on master problems	0.1	0.2	0.1
% spent on separation	81.8	75.9	91.2
% overhead time	18.1	23.9	8.7

Note: The total time includes the time to execute the algorithm and subordinate solver calls. The percentage of time spent on master and separation problems only includes the total execution time for the respective subordinate solvers.
Abbreviations: DR, decision rules; GRCS, generalized robust cutting-set.



(a) Nominal design ($k = 0$); feasibility against constraint (S1h).

(b) Intermediate design ($k = 1$); feasibility against constraint (S1i).

(c) Robust feasible design ($k = 2$); feasibility against constraint (S1h).

(d) Final GRCS design ($k = 3$); fully robust feasible.

FIGURE 4 Evolution during the generalized robust cutting-set (GRCS) algorithm of the robust feasibility of the reactor–separator designs using the static approximation policy [Color figure can be viewed at wileyonlinelibrary.com]

All points shown as a green triangle are the realizations at which the current design is explicitly robust against.** All points represented as yellow crosses are violating points identified in solving $SP_{k,i}$ at the current iteration k. Constraints under these realizations will thus be appended in subsequent iterations of the master problem. We also note that, because the uncertainty set considered in the reactor–separator model is four-dimensional, we merely project to the two dimensions with the largest variance, namely k_3 and k_4, and the black dotted lines represent the projection of the boundary of the uncertainty set in these two dimensions.

The plots reveal two regions in which each of the inequality constraints are active and lead to constraint violations. The main region of infeasibility for constraint (S1h) is in the upper-right end of the uncertainty set, while the region where violations for constraint (S1i) occur in the lower-left end. The trajectory toward robustness against each performance constraint is easily seen over progressive iterations of the algorithm for each recourse policy case, as the region of red infeasible sample points becomes blue. The amount of infeasibility that persists in each iteration is quantified in Table S1.

5.2 | Case Study II: Reactor–heater

In this case study, we consider the flow sheet shown in Figure 5, which represents a reactor–heater system previously studied in Halemane and Grossmann,[47] Varvarezos et al,[48] Rooney and Biegler,[49] and Yuan et al.[26] This system consists of a reactor and heat exchanger with a single first-order, exothermic reaction to convert reactant A to product B. The task is to identify a design that achieves the target of 90% conversion of reactant A, while minimizing EAC.

The deterministic reactor–heater model consists of 10 decision variables, five equality constraints, four inequality constraints (excluding variable bounds), and 10 parameters of which two will be considered uncertain. The complete NLP model for the reactor–heater flow sheet, as well as a table of pertinent data, are shown in Sections S2.1 and S2.2, respectively. The objective function in Equation (S2a) seeks to minimize the EAC, including annualized capital cost and yearly operating cost terms taken from Varvarezos et al.[48] The set of state equations $\mathcal{J}$ includes Equations (S2b)–(S2f), while the set of performance constraints $\mathcal{I}$ includes Equations (S2j)–(S2n) as well as the bounds on second-stage variables in Equations (S2o) and (S2p).

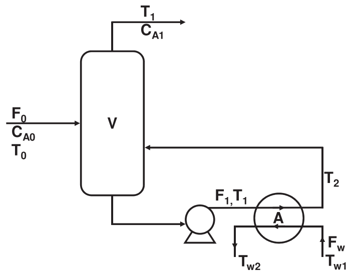


FIGURE 5 Flow sheet representing the reactor–heater system considered in Section 5.2, as adapted from Halemane and Grossmann[47]

In this optimization model, the size of the reactor V and the area of the heat exchanger A are the first-stage decision variables, while the two second-stage variables are the flow rates F_1 and F_w dictating utility usage. The remaining variables, x_A, T_1, T_2, T_{w1}, T_{w2}, and $\Delta T_{ln}^{\dagger\dagger}$ constitute state variables. The uncertainty considered in this example lies in the Arrhenius rate constant for the reaction taking place in the reactor, k_0, and overall heat-transfer coefficient in the heat exchanger, U. Thus, $x = (V, A)$, $z = (F_1, F_w)$, all other variables are state variables y, and $q = (k_0, U)$. For this study, we postulate that the two aforementioned uncertain parameters are independent to each other, and therefore utilize a simple two-dimensional box uncertainty set. The data for the nominal values and uncertainty deviations of k_0 and U can be found in Section S2.3.

5.2.1 | Case Study II results

Results for the optimal first-stage variables and costs for the deterministic and robust feasible cases are shown in Table 5. In this case study, the GRCS algorithm returned the same robust solution for the affine and quadratic DR. This outcome is possible due to the fact that a feasible solution with affine DR is also a feasible solution under quadratic DR, and it just so happens that, for this particular model and uncertainty set, greater flexibility via quadratic DR does not improve upon the robust feasible solution. This observation alludes to the possibility that the extra computational burden to consider more involved recourse policies might not always yield a payoff. To that end, the modeler has a critical role to play in judiciously selecting the form of DR to be employed in the context of each particular model. In Section 5.4, we provide additional remarks on the selection of recourse policy.

When comparing the deterministic first-stage variables to the robust solutions, we see that there is an increase in reactor volume and heat exchanger area for robust solutions. The robust designs must satisfy the constraint regarding production of product A for all uncertain parameter values within the uncertainty set, and this is accomplished via larger capacity facilitated by larger equipment.

The optimal values for second-stage variables and costs for the deterministic and robust feasible cases are shown in Table 6, where we note that these do not change substantially under the case of the nominal realization. When we compare the values of the control variables, F_1 and F_w, in the deterministic case against their robust feasible values, we notice that the values of these flow rates increase, which leads to an increase in second-stage costs. This can be explained by the fact the reaction producing product A is exothermic, requiring increased circulation through the cooler in the robust designs. In regards to expected second-stage variable values and costs shown in Table 7, we observe that these values are actually greater than the nominal values in the previous table. This may be an indicator of the optimizer taking advantage of the nominal second-stage cost, $f_2(x, z, y; q^0)$, in the objective. In the case of the nominal uncertain parameter

TABLE 5 Optimal values of first-stage variables and costs for the reactor–heater model

	Deterministic	Static approx.	Affine DR	Quadratic DR
V (m^3)	4.43	4.98	4.94	4.94
A (m^2)	9.70	9.97	9.92	9.92
$f_1(x^*)$ (\$/year)	5,374.66	5,596.62	5,575.26	5,575.26

Abbreviation: DR, decision rules.

TABLE 6 Optimal values of second-stage variables and costs, under the nominal realization of uncertainty, for the reactor–heater model

	Deterministic	Static approx.	Affine DR	Quadratic DR
F_1 (kmol/hr)	94.19	95.77	95.69	95.69
F_w (kmol/hr)	1,754.75	1,782.49	1,784.21	1,784.21
$f_2(z^*, y^*; q^0)$ (\$/year)	4,107.47	4,175.15	4,177.78	4,177.78

Abbreviation: DR, decision rules.

TABLE 7 Expected values and *SD*s of second-stage variables and costs for the reactor–heater model

	Static approx.	Affine DR	Quadratic DR
$\mathbb{E}[F_1] \pm \sigma[F_1]$ (kmol/hr)	96.98 ± 9.82	96.80 ± 9.63	96.80 ± 9.63
$\mathbb{E}[F_w] \pm \sigma[F_w]$ (kmol/hr)	1,809.47 ± 97.89	1,798.57 ± 94.64	1,798.57 ± 94.64
$\mathbb{E}[f_2] \pm \sigma[f_2]$ (\$/year)	4,236.47 ± 264.11	4,214.15 ± 256.51	4,214.15 ± 256.51
$\mathbb{E}[\zeta] \pm \sigma[f_2]$ (\$/year)	9,833.10 ± 264.11	9,789.41 ± 256.51	9,789.41 ± 256.51

Abbreviation: DR, decision rules.

realization, q^0, the robust feasible design requires slightly lower flow rates, and correspondingly a lower second-stage cost. However, in expectation, these flow rates will be higher, as reported in Table 7. It is also interesting to note that DR seem to help alleviate some of the conservativeness of the static approximation solution, as they strictly improve in terms of EAC, both nominal and expected.

The total number of iterations of the GRCS algorithm, its total CPU time and the fraction of total time spent on various portions of the algorithm are shown in Table 8. The number of iterations and CPU times are relatively small, due to the small scale and good tractability of the deterministic optimization problem. As in the previous example, the majority of the GRCS algorithm's time is spent solving separation problems. The algorithmic overhead tasks also consumed a significant fraction of the total CPU time, noting though that the very small total CPU times do not encourage general insights.

In Figure 6, we show the progression of feasibility over each iteration of the GRCS for the case of the static approximation recourse policy. Similar plots for the cases of affine and quadratic DR policies are provided in Figures S3 and S4. In these plots, the designs from each iteration of the master problem are evaluated across a set $\mathcal{S}$ of 200 uniformly sampled realizations $q_s \in \mathcal{Q}$, $s \in \mathcal{S}$. The feasibility of a design under a realization q_s is defined against the performance constraint in Equation (S2n), as this was the only inequality constraint in the formulation that led to violations in the separation problems solved. In other words, for points shown in blue, the design leads to $x_A \geq 0.9$, while for points shown in red, $x_A < 0.9$. Figure 6a, which is a plot depicting the feasibility of the nominal deterministic design under various uncertain parameter realizations, clearly shows a horizontal division in the uncertainty set between a region of feasibility and a region of infeasibility. More specifically, while the nominal realization, $q^0 = \left(U^0, k_0^0\right)$, is included in the region of feasibility, most points with $k_0 < k_0^0$ render the design infeasible. Interestingly, the first violating parameter realization identified in separation, q^1, is a non-vertex point on the border of the otherwise polyhedral uncertainty set. The amount of infeasibility that persists in each iteration is quantified in Table S2.

5.3 | Case Study III: MEA-solvent CO_2 separation flow sheet

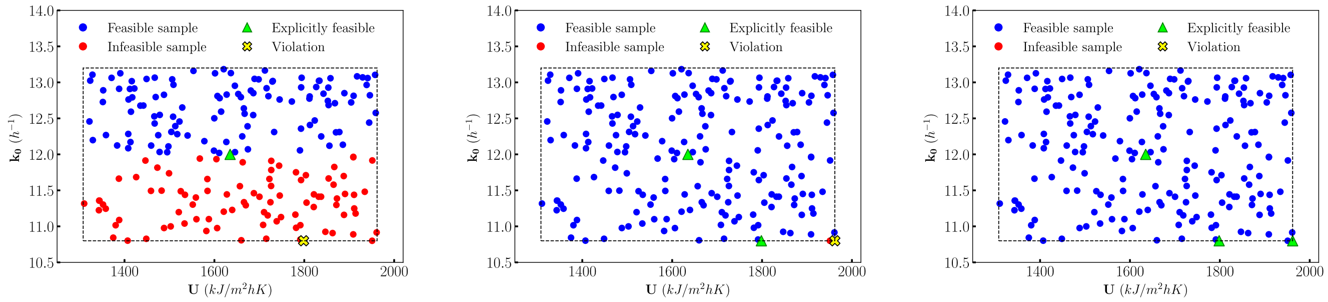
The final example studied here is a complete equation-oriented model for post-combustion CO_2 capture using a monoethanolamine (MEA) solvent. The complete flow sheet for the process is shown in Figure 7. The key units in this process include the absorber and regenerator (stripper) columns, the cross heat exchanger, and the reboiler and condenser. The electrolyte non-random two-liquid (e-NRTL) model[51] was used to represent the vapor–liquid equilibrium, while an enhancement factor[52] is used to characterize the effect of liquid phase reaction on the mass transfer rate between the liquid and gas phases. The latter was based on a two-film model, wherein mass transfer occurs via molecular diffusion through a stagnant film of a given thickness and the bulk phase is well mixed.[53] In addition, our process model includes rigorous submodels to describe gas–liquid reactions, surface tension, diffusivity, and heat transfer. The mass and energy balances in the column models are differential equations, as the concentrations of vapor and liquid components as well as the temperature varies along the length of the columns. Some representative model equations are

TABLE 8 Total number of iterations and CPU time spent within the GRCS algorithm when addressing the reactor–heater model

	Static approx.	Affine DR	Quadratic DR
# of GRCS iterations	3	3	3
Total CPU time (s)	1.3	2.9	4.0
% spent on master problems	0.5	0.3	0.4
% spent on separation	49.0	66.5	66.0
% overhead time	50.5	33.2	33.6

Note: The total time includes the time to execute the algorithm and subordinate solver calls. The percentage of time spent on master and separation problems only includes the total execution time for the respective subordinate solvers.

Abbreviations: DR, decision rules; GRCS, generalized robust cutting-set.



(a) Nominal design ($k = 0$); feasibility against constraint (S2n).

(b) Intermediate design ($k = 1$); feasibility against constraint (S2n).

(c) Final GRCS design ($k = 2$); fully robust feasible.

FIGURE 6 Evolution during the generalized robust cutting-set (GRCS) algorithm of the robust feasibility of the reactor–heater designs using the static approximation policy [Color figure can be viewed at wileyonlinelibrary.com]

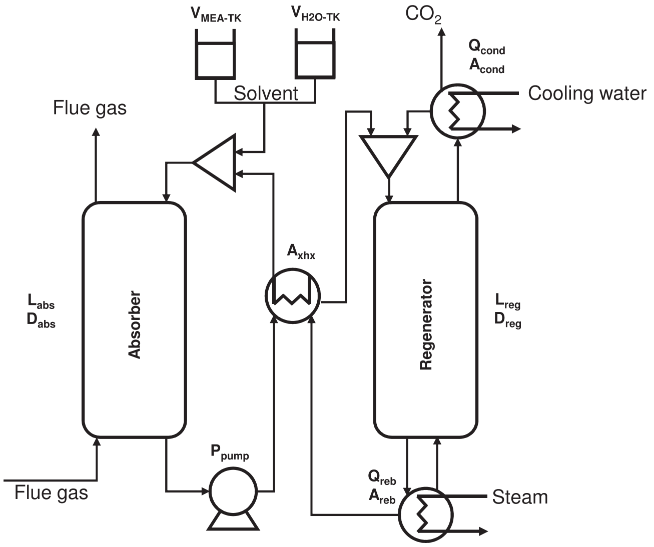


FIGURE 7 Flow sheet representing the MEA-based CO_2 capture flow sheet considered in Section 5.3, as adapted from Mores et al[46]

shown in Section S3.3 and have first been presented in Chinen et al[54] for a solvent-based carbon capture system. Notably, the simulation model of the flow sheet, featuring zero degrees of freedom and 30 finite elements to discretize the differential equations, is a system of 4,334 variables and 4,327 constraints.

In our study, we seek to design a process for treating 1,000 mol/s of flue gas, with the main performance constraint calling for the process to achieve 85% CO_2 capture. Therefore, we augmented the simulation model to include degrees of freedom for investments to install capacity for the various utilities and an overall economic objective to minimize the EAC. We adapted the straightforward algebraic formulation of Mores et al[46] and utilize simple capital cost correlations from Towler and Sinnott[55] and Seider et al[56] to demonstrate relative impact on EAC of our RO approach. Note that we do not account for uncertainty in costs and that the use of more rigorous costing methods are outside the scope of this paper. The full set of equations defining the EAC function can be found in Section S3.2.

After the above augmentations, the deterministic optimization model we used consists of 4,342 decision variables, 4,327 equality constraints, and 13 inequality constraints (excluding variable bounds). The main first-stage design variables correspond to the sizing of major equipment, namely the diameters of the columns, D_{abs} and D_{reb}, the heights of the columns, L_{abs} and L_{reg}), and the total surface area of the cross heat exchanger, A_{xhx}. There are also several first-stage variables introduced in the model to properly capture the investment costs for the installed capacities of necessary utilities. These include the power to pump CO_2-rich solvent from the absorber bottom to the inlet of the regenerator, P_{pump}, volumes of storage tanks for the hold-up of H_2O and MEA solvent to supply to the make-up mixer, V_{MEA-TK} and V_{H_2O-TK}, as well as the effective surface areas of the reboiler and condenser, A_{reb} and A_{con}, respectively. In summary, $x = (D_{abs}, D_{reg}, L_{abs}, L_{reg}, A_{xhx}, P_{pump}, V_{MEA-TK}, V_{H_2O-TK}, A_{reb}, A_{con})$. The second-stage control variables are the heat duties supplied to the reboiler and condenser, thus $z = (Q_{reb}, Q_{cond})$. All remaining variables are considered to be state variables y.

Potential sources of uncertainty have been studied in the literature,[54,57] while the optimization under uncertainty of the absorber column unit for MEA-based post-combustion carbon capture has been studied also in Reference 58. Here, we expand upon previous works by studying the entire process flow sheet, presenting solutions obtained via our novel algorithm. More specifically, we postulate an ellipsoidal uncertainty set representing the 95% confidence interval around two parameters, $q = (b_1, b_2)$, used to calculate equilibrium constants $K_{eq,r}$, $r \in \{1, 2\}$ via Equations (S3m) in Section S3.3. Here, the index r refers to two solvent-phase reactions considered in the kinetic model.[59] We stress that parameters b_1 and b_2 participate non-linearly in the set of equations defining the equilibrium constants, while the latter are further non-linearly related to the overall mass-transfer coefficient between the vapor and liquid phases. As we shall show later, this leads to interesting regions of feasibility of the deterministic design within the uncertainty set. Details regarding the uncertainty set used, including the mean and covariance matrix values, are provided in Section S3.4.

There are several inequality constraints in the model representing requirements on performance. These include a constraint on the fraction of CO_2 captured, that is, $\eta_{CO_2} \geq 0.85$, where $\eta_{CO_2} = \frac{y^{in}_{CO_2} - y^{out}_{CO_2}}{y^{in}_{CO_2}}$ and y_{CO_2} is the gas phase mole fraction of CO_2. In addition, explicit constraints on the superficial vapor velocities at the bottom of the absorber and the top of the regenerator are imposed, in order to prevent flooding. These constraints can be found in Section S3.3 (Equations S3n–S3r). Additional inequality constraints of interest are the bounds on the control variables, namely the reboiler and condenser duties. The latter include trivial "zero" bounds to limit their sign (Equations S3u and S3v) as well as induced bounds due to the availability of steam (Equation S3w) and cooling water (Equation S3x), respectively. In the case of affine DR, these bounds must be included in the set of performance constraints to be separated against. All other equations in the model represent state equations, in which the uncertain parameters participate nonlinearly and implicitly. Additionally, we note that the flue gas flow rate at the bottom of the absorber is fixed to 1,000 $\frac{\text{mol}}{\text{s}}$, while the solvent flow rate at the top of the absorber is fixed to 3.64 $\frac{\text{kmol}}{\text{s}}$. The solvent's temperature at this inlet is calculated as a state variable in the model. The solvent make-up is set to be 27.5% wt. MEA at the outlet of the make-up mixer.

5.3.1 | Case Study III results

For this case study, we only consider the static approximation and affine DR as our two recourse policies. Results under quadratic recourse are not presented due to the fact that the subordinate nonlinear optimization solver failed to converge while attempting to solve a subproblem generated at an intermediate iteration of the GRCS algorithm. This is an example of how the overall performance of the GRCS algorithm depends on the ability of the employed subordinate solvers to identify optimal solutions to subproblems at each and every iteration and in reasonable computational times.

The optimal values for first-stage variables and costs for the deterministic and robust feasible solutions are shown in Table 9. Our deterministic design is consistent with general trends established in optimal designs previously presented in the literature,[46,60] such as the fact that the absorber height is greater than the regenerator column, as well as a drum-like regenerator column wherein $L/D \approx 1$. One possible motivation for this is to drive down capital and utility costs associated with pumping the solvent to the top of a higher regenerator column, at the expense of increased reboiler duty. This may also be motivated by the presence of explicit constraints to limit the superficial vapor velocity within a fraction of the flooding velocity at the regenerator bottom, since an increase in diameter corresponds to a decrease in velocity for a given volumetric flow rate.

We begin by comparing the deterministic solution to the robust solutions. First, we notice that the robust solutions (static approximation and affine DR) feature decreased heights of both columns as well as an increased diameter of the regenerator column, as compared to the deterministic solution. As a result of the reboiler column height decrease in the robust solutions, there is also a corresponding decrease in the required pump power. Additionally, there is an increase in heat transfer surface area in the reboiler and condenser. This can be attributed to the increase in reboiler and condenser heat duties, contributing to achieving robust feasibility. This also relates to the decrease in the robust values of cross heat exchanger surface area, since more heat transfer load between streams can be taken on by the larger condenser and reboiler.

Next, we consider notable differences between the robust solutions determined via the static approximation and affine DR policy. First, we see that the optimal surface areas for the heat exchanger, reboiler and condenser are smaller in magnitude in the case of an affine DR policy, when compared to the static approximation. Because

TABLE 9 Optimal values of first-stage variables and costs for the CO_2 capture flow sheet model

	Deterministic	Static approx.	Affine DR
L_{abs} (m)	7.57	6.00	6.93
D_{abs} (m)	4.95	4.96	4.96
L_{reg} (m)	4.00	3.00	3.52
D_{reg} (m)	3.44	4.04	4.00
A_{xhx} (m^2)	4,734	3,928	3,764
P_{pump} (kW)	4.44	3.67	4.28
V_{MEA-TK} (m^3)	2.25	3.46	3.48
V_{H_2O-TK} (m^3)	11,541	13,300	13,486
A_{reb} (m^2)	179.4	813.4	797.4
A_{con} (m^2)	191.8	2,116.8	2,034.0
$f_1(x^*)$ (MM$/year)	2.06	2.07	2.09

Abbreviation: DR, decision rules.

the affine DR allow for more flexible heat transfer policies in the second-stage, there is less need to invest in larger equipment, such as heat transfer surface areas, in the first-stage. This decrease in heat transfer surface areas in the affine case leads to a corresponding increase in the regenerator column height, which is necessary for achieving sufficient CO_2 regeneration at the implied lower reboiler and condenser duties. In spite of the above differences, the first-stage capital costs appear to be very similar in all three solutions.

Table 10 presents information related to the second-stage variables and costs, under the scenario of nominal uncertainty realization. In this case, the robust reboiler heat duty, Q_{reb}, is much higher than the optimal value in the deterministic case, as is also the flow rate of solvent supplied to the regenerator. This is illustrating a trade-off wherein the shorter, more drum-like regenerator columns require higher reboiler duties to achieve the lower lean loading values. The decrease in column height also allows for larger flow rates of solvent due to a decreased pumping requirement. As expected, robustness comes at a noticeable increase in total cost, which is necessary for the design to possess enough flexibility to remain operational (feasible) under a wide range of scenarios (uncertainty set). It is important to note, however, that the use of affine DR leads to a robust feasible solution with a slightly lower nominal EAC, when compared to the static approximation policy, which demonstrates reduction of the two-stage adaptivity gap. In addition to the optimal deterministic second-stage variables and costs, Table 10 also shows the optimal values for several state variables of interest. These include the CO_2 capture, denoted as η_{CO_2}, the lean loading calculated as $\alpha = \frac{x_{CO_2}}{x_{MEA}}$, where x_{CO_2} and x_{MEA} are liquid phase mole fractions, as well as several key flow rates related to the pumping utility ($F^{\text{in}}_{\text{reg,liq}}$) and the solvent make-up supplied by the mixer at the top of the absorber ($F^{\text{in}}_{\text{mix,MEA}}$ and $F^{\text{in}}_{\text{mix,}H_2O}$). We see that the nominal capture for the robust solutions, in both the affine DR and static approximation cases, is larger than the deterministic case. To achieve feasibility against all uncertain parameter realizations, there must be a degree of "over-design" when considered in light of a subset of realizations in the uncertainty set. This means that, for some realizations, and in particular the nominal realization, the robust design will exceed the requirements of the performance constraints. Additionally, we see that lean loading values are lower in the robust solutions, when compared to the deterministic case, due to the increase in the robust reboiler duties.

The expected values and *SD*s for the second-stage costs and variables are shown in Table 11. We note that the expected second-stage costs are significantly lower than their values when the nominal realization prevails. This can be attributed to the fact that the expected values are determined without a particular DR policy in place, instead being calculated in light of unrestricted recourse potential. By simply allowing for more flexibility in second-stage recourse, there is a broader range of operating points available, which can decrease the second-stage costs. Unlike in the previous case studies, the affine DR solution does not result in lower overall expected costs when compared to the static approximation solution, albeit these costs show less variation. Note that the trade-offs between first-stage investment costs and second-stage operating costs are more complex in this system, and since the economic objective function only considers the nominal scenario, we do not get a guarantee that the designs under the more flexible recourse policy will be lower in expectation.

In terms of control variables, Table 11 reveals that the need for reboiler duty is generally more stable than the need for condenser duty. Indeed, as different scenarios were sampled, the latter ranged quite a bit more on a relative scale, leading to a more skewed distribution with an elongated tail. Overall, the range of values that these duties would have to attain under the various sampled scenarios is larger in the design stemming from the affine DR policy, which can be explained by the fact that this design was explicitly chosen to perform under a wider range of control variable values.

The iterations count and algorithm execution times for the CO_2 capture flow sheet study are shown in Table 12. When compared to the previous case studies, the total CPU times are much higher due to the larger model size and complexity. For reference, the deterministic version of the CO_2 capture flow sheet model already requires 31 s of

TABLE 10 Optimal values of second-stage control and other key variables, as well as total and second-stage costs, evaluated under the nominal realization of uncertainty, for the CO_2 capture flow sheet model

	Deterministic	Static approx.	Affine DR
Q_{reb} (MW)	18.14	41.10	39.05
Q_{con} (MW)	−4.54	−25.18	−22.89
$\eta_{CO_2}\left(\frac{\text{mol}\,CO_2}{\text{total mol}}\right)$	0.85	0.93	0.96
$\alpha\left(\frac{\text{mol}\,CO_2}{\text{mol MEA}}\right)$	0.22	0.17	0.17
$F^{\text{in}}_{\text{reg,liq}}$(kg/s)	84.84	92.10	91.22
$F^{\text{in}}_{\text{mix,MEA}}$(kg/s)	0.03	0.04	0.04
$F^{\text{in}}_{\text{mix,}H_2O}$(kg/s)	4.44	4.85	5.00
$f_2(z^*, y^*; q^0)$ (MM$/year)	5.19	8.83	8.67
ζ^* (MM$/year)	7.25	10.90	10.76

Abbreviation: DR, decision rules.

TABLE 11 Expected values and *SD*s of second-stage control and other key second-stage variables, and second-stage costs, for the CO_2 capture flow sheet model

	Static approx.	Affine DR
$\mathbb{E}[Q_{\text{reb}}] \pm \sigma[Q_{\text{reb}}]$ (MW)	18.57 ± 0.55	19.27 ± 4.42
$\mathbb{E}[Q_{\text{con}}] \pm \sigma[Q_{\text{con}}]$ (MW)	–0.94 ± 1.36	–1.86 ± 4.87
$\mathbb{E}[\eta_{\text{CO}_2}] \pm \sigma[\eta_{\text{CO}_2}]$	0.85 ± 0.00	0.86 ± 0.03
$\mathbb{E}[\alpha] \pm \sigma[\alpha] \left(\frac{\text{mol CO}_2}{\text{mol MEA}}\right)$	0.24 ± 0.03	0.24 ± 0.03
$\mathbb{E}\left[F^{\text{in}}_{\text{reg,liq}}\right] \pm \sigma\left[F^{\text{in}}_{\text{reg,liq}}\right] (\text{kg/s})$	83.63 ± 0.51	83.87 ± 1.52
$\mathbb{E}\left[F^{\text{in}}_{\text{mix,MEA}}\right] \pm \sigma\left[F^{\text{in}}_{\text{mix,MEA}}\right] (\text{kg/s})$	0.03 ± 0.00	0.03 ± 0.00
$\mathbb{E}\left[F^{\text{in}}_{\text{mix,H}_2\text{O}}\right] \pm \sigma\left[F^{\text{in}}_{\text{mix,H}_2\text{O}}\right] (\text{kg/s})$	5.89 ± 0.34	5.79 ± 0.27
$\mathbb{E}[f_2] \pm \sigma[f_2]$ (MM$/year)	5.51 ± 1.38	5.63 ± 0.78

Abbreviation: DR, decision rules.

TABLE 12 Total number of iterations and CPU time spent within the GRCS algorithm when addressing the CO_2 capture flow sheet model

	Static approx.	Affine DR
# of GRCS iterations	5	5
Total CPU time (s)	1,030.0	1,543.4
% spent on master problems	19.4	16.2
% spent on separation	48.6	51.2
% overhead time	32.0	32.6

Note: The total time includes the time to execute the algorithm and subordinate solver calls. The percentage of time spent on master and separation problems only includes the total execution time for the respective subordinate solvers.
Abbreviations: DR, decision rules; GRCS, generalized robust cutting-set.

CPU time to optimize locally. Table 12 also reveals that, despite the number of iterations being the same in both cases, the total CPU time for the affine DR is roughly 50% higher than the static approximation case, alluding to the fact that the complexity of the GRCS is directly related to the complexity of the DR relationship used. Additionally, we see the same trend as in previous case studies wherein the percentage of time spent solving separation problems is significantly larger than that spent solving the master problems. We also see that there is significant overhead time spent outside of subordinate solver optimization calls. In particular, due to the larger models involved in this case study, there are significantly more constraints and variables that need to be copied at each iteration as the master problem formulation evolves, leading to increased times spent toward Pyomo model building.

Figure 8 presents the progression of feasibility over each iteration of the GRCS under the static approximation policy. The progression under affine DR can be found in Figure S5. As before, designs are evaluated in light of 200 uniformly sampled realizations $q_s \in \mathcal{Q},\ s \in \mathcal{S}$. Here, feasibility of the designs primarily refers to the CO_2 capture constraint and the upper bound constraining the superficial vapor velocity in the absorber, as these were the only inequality constraints to lead to violations in the separation step. The nominal design performance shown in Figure 8a reveals that there exist two disjoint regions within the uncertainty set that render the nominal solution infeasible, which is indicative of the nonlinear manner in which the uncertain parameters affect design feasibility. Regardless, after a few iterations of the GRCS algorithm, the design becomes robust feasible across the totality of the uncertainty set. The amount of infeasibility that persists in each iteration is quantified in Table S3. It is also worth noting that, in this example, all of the violating parameter realizations added to master problems were points from the boundary of the ellipsoidal uncertainty set.

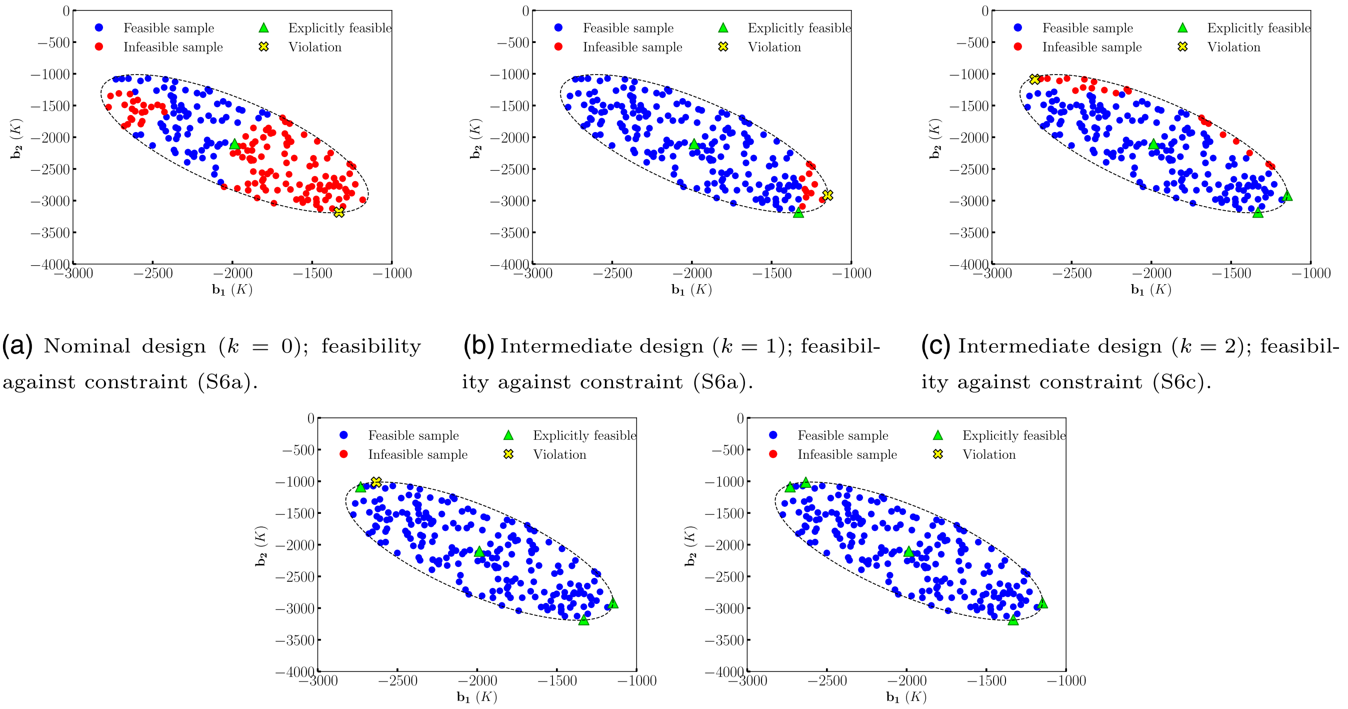
5.4 | Choosing form of recourse policy

In this section, we summarize some empirical findings and conclusions pertaining to the performance of the different recourse policies, hoping to inform a future modeler that wishes to address other process design models via our GRCS algorithm. First, we note that, in each of the examples investigated, the costs of designs determined via affine and quadratic DR are relatively close to each other, while they both differ more significantly from the deterministic and static approximation costs. From this perspective, our results clearly support the incentive to choose some kind of DR (i.e., affine or quadratic) over the static approximation recourse policy.

The choice to deviate from affine DR and invoke quadratic rules was less clearly motivated in our investigations. For example, the results for Case Study I showed a small, yet strict, improvement in expectation (i.e., $\mathbb{E}[\zeta]_{\text{ADR}} > \mathbb{E}[\zeta]_{\text{QDR}}$), while in Case Study II, the two equally-robust solutions where the same. For a general model, there is no a-priori indicator to predict whether the invocation of quadratic DR will or will not improve the results derived with affine DR. Certainly, as one considers hierarchies of DR (e.g., going from affine to quadratic to cubic, or even, to quartic polynomials), diminishing returns are expected due to the closure of the two-stage adaptivity gap, and the modeler has an incentive to empirically investigate that. This in fact strengthens the need for an algorithm like GRCS, which can consider different DR functions in a modular manner, enabling the modeler to perform such investigations.

One should also keep in mind that the GRCS algorithm also admits non-polynomial rules, which cannot be hierarchically compared to those tested here; those non-polynomial rules may prove to perform particularly well in certain models. We should also highlight that there generally exist opportunities to employ the more involved DR forms only for a carefully select subset of second-stage decisions, leaving the remaining ones to be decided based on simpler recourse policies (e.g., static approximation). Whereas at the interest of brevity such strategy was not explored in this work, anyone wishing to apply the GRCS algorithm presented here can readily do so.

Finally, any potential savings in terms of second-stage costs have to be viewed in light of the computational burden associated with choosing more involved DR functions, as the complexity of solving the resulting optimization subproblems is expected to increase, in general. For example, in cases when variables *z* and *y* as well as



(a) Nominal design ($k = 0$); feasibility against constraint (S6a).

(b) Intermediate design ($k = 1$); feasibility against constraint (S6a).

(c) Intermediate design ($k = 2$); feasibility against constraint (S6c).

(d) Intermediate design ($k = 3$); feasibility against constraint (S6c).

(e) Final GRCS design ($k = 4$); fully robust feasible.

FIGURE 8 Evolution during the generalized robust cutting-set (GRCS) algorithm of the robust feasibility of the MEA-based CO_2 capture flow sheet using the static approximation policy [Color figure can be viewed at wileyonlinelibrary.com]

parameters q participate linearly in the deterministic model, and the uncertainty set is polyhedral, quadratic DR may not be preferable, since they will change the class of the underlying separation problems from linear to nonlinear. In contrast, if the deterministic model is already nonlinear, as is often the case in the process systems engineering context, then quadratic DR are a plausible option. Whether or not the additional nonlinearities from the quadratic DR lead to exceedingly difficult numerical issues when paired with a particular model and solver would need to be explored in each case. Eventually, the practitioner must empirically determine the trade-off between solution quality and tractability, recognizing that employing higher order polynomial DR will come at a computational cost, in general.

6 | CONCLUSIONS

In this work, we proposed a GRCS algorithm for identifying robust feasible solutions to two-stage, non-convex process systems models under uncertainty. Starting from a given deterministic process optimization problem of interest, we formalize elements of the latter as first- and second-stage costs, first-stage design variables, second-stage control variables, irremovable state variables and state equations, uncertain parameters, as well as inequality constraints on system performance. With this information, and utilizing a suitable recourse policy, including affine or quadratic DR, to handle second-stage control variables, the GRCS algorithm was shown to be able to achieve robust feasible solutions. We demonstrated the overall algorithm via three case studies, including one very comprehensive, full-flow sheet model of a post-combustion carbon capture process. Through these case studies, we showed that flexibility in recourse is inversely related to total costs, notably demonstrating the use of nonlinear DR toward this. To conclude, our work builds upon existing literature by applying general, model-agnostic RO methodologies to process systems engineering models. With this capability, practitioners of process design under uncertainty can readily identify risk-averse solutions that are explicitly robust against user-defined uncertainty sets.

NOTATION

Sets

$\mathcal{I}$	set of performance constraints
$\mathcal{J}$	set of state equations
$\mathcal{K}$	set of algorithm iterations
$\mathcal{Q}$	uncertainty set

Variables

$x \in \mathcal{X} \subseteq \mathbb{R}^m$	first-stage design variables
$z \in \mathbb{R}^n$	second-stage control variables
$y \in \mathbb{R}^a$	state variables
$d \in \mathbb{R}^p$	decision rule variables

$q \in \mathcal{Q}$	uncertain parameters
$q^0 \in \mathbb{R}^w$	nominal uncertain parameter values
$\zeta \in \mathbb{R}$	objective function value

Functions

g	inequality constraints
h	equality constraints
f_1	first-stage costs
f_2	second-stage costs
v	decision rule function
$\mathbb{E}(\cdot)$	expectation of
$\sigma(\cdot)$	standard deviation of

ACKNOWLEDGMENTS

This research was conducted as part of the Institute for the Design of Advanced Energy Systems (IDAES) with funding from the Simulation-Based Engineering, Crosscutting Research Program of the U.S. Department of Energy's Office of Fossil Energy. Natalie Isenberg is also grateful for support from a U.S. Department of Energy, Office of Science Graduate Student Research award.

AUTHOR CONTRIBUTIONS

Natalie Isenberg: Data curation; formal analysis; investigation; methodology; software; visualization; writing-original draft. **Paul Akula:** Data curation; validation. **John Eslick:** Software. **Debangsu Bhattacharyya:** Validation; writing-review and editing. **David Miller:** Funding acquisition; writing-review and editing. **Chrysanthos Gounaris:** Conceptualization; funding acquisition; methodology; project administration; resources; supervision; writing-review and editing.

DISCLAIMER

This paper was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

CONFLICT OF INTEREST

None are to be reported.

ENDNOTES

[*] This unique mapping from (x, z, q) to y is a reasonable assumption for most process models. If this assumption does not hold, however, the offending elements of vector y should be regarded as flexible, second-stage variables and handled in the same way as the z variables.

[†] In the case of a quadratic decision rule function, it holds by construction that p:= 1 + w + w(w + 1)/2.

[‡] Since ζ^* is the minimal objective value of the master problem, requiring that the objective of the polishing problem attains a value no greater than ζ^* is equivalent to requiring that its solution corresponds to one of the equivalently optimal solutions of the master problem.

[§] We remark that the chosen scenarios may be samples from within the uncertainty set, that is, $q_s \in \mathcal{Q}$, or may be out-of-sample scenarios.

[¶] We remark that the first-stage costs would not vary depending on the scenario q_s, since the design variables, x^*, are kept fixed in this analysis.

[**] For example, the nominal realization q^0 is used as early as the first master problem to initialize the GRCS algorithm, and hence every master problem solution is explicitly robust against the nominal point.

[††] Note that we made use of the Underwood formula[50] to approximate ΔT_{ln}, which was found to be helpful in avoiding numerical issues with the NLP solvers.

DATA AVAILABILITY STATEMENT

The data that supports the findings of this study are available in the supplementary material of this article.

ORCID

Chrysanthos E. Gounaris https://orcid.org/0000-0001-5779-2510

REFERENCES

1. Grossmann IE, Sargent RWH. Optimum design of chemical plants with uncertain parameters. *AIChE J*. 1978;24(6):1021-1028.
2. Pistikopoulos EN, Marianthi G. Ierapetritou novel approach for optimal process design under uncertainty. *Comput Chem Eng*. 1995;19(10):1089-1110.
3. Acevedo J, Pistikopoulos EN. Stochastic optimization based algorithms for process synthesis under uncertainty. *Comput Chem Eng*. 1998;22(4–5):647-671.
4. Wendt M, Li P, Wozny G. Nonlinear chance-constrained process optimization under uncertainty. *Ind Eng Chem Res*. 2002;41(15):3621-3629.
5. 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.
6. Grossmann IE, Floudas CA. Active constraint strategy for flexibility analysis in chemical processes. *Comput Chem Eng*. 1987;11(6):675-693.
7. 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.
8. Li C, Grossmann IE. An improved l-shaped method for two-stage convex 0–1 mixed integer nonlinear stochastic programs. *Comput Chem Eng*. 2018;112:165-179.
9. Kelley MT, Baldick R, Baldea M. Demand response scheduling under uncertainty: chance-constrained framework and application to an air separation unit. *AIChE J*. 2020;66(9):e16273.
10. Wang Y, Biegler LT, Patel M, Wassick J. Robust optimization of solid-liquid batch reactors under parameter uncertainty. *Chem Eng Sci*. 2020;212:115170.
11. Ben-Tal A, Goryashko A, Guslitzer E, Nemirovski A. Adjustable robust solutions of uncertain linear programs. *Math Program*. 2004;99(2):351-376.
12. Zhang Q, Morari MF, Grossmann IE, Sundaramoorthy A, Pinto JM. An adjustable robust optimization approach to scheduling of continuous industrial processes providing interruptible load. *Comput Chem Eng*. 2016;86:106-119.

13. Shi H, You F. A computational framework and solution algorithms for two-stage adaptive robust scheduling of batch manufacturing processes under uncertainty. *AIChE J*. 2016;62(3):687-703.
14. Lappas NH, Gounaris CE. Multi-stage adjustable robust optimization for process scheduling under uncertainty. *AIChE J*. 2016;62(5):1646-1667.
15. Lappas NH, Gounaris CE. Theoretical and computational comparison of continuous-time process scheduling models for adjustable robust optimization. *AIChE J*. 2018;64(8):3055-3070.
16. Lappas NH, Ricardez-Sandoval LA, Fukasawa R, Gounaris CE. Adjustable robust optimization for multi-tasking scheduling with reprocessing due to imperfect tasks. *Optim Eng*. 2019;20(4):1117-1159.
17. Tejeda-Iglesias M, Lappas NH, Gounaris CE, Ricardez-Sandoval L. Explicit model predictive controller under uncertainty: an adjustable robust optimization approach. *J Process Control*. 2019;84:115-132.
18. Zhang X, Kamgarpour M, Georghiou A, Goulart P, Lygeros J. Robust optimal control with adjustable uncertainty sets. *Automatica*. 2017;75:249-259.
19. Subramanyam A, Mufalli F, Pinto JM, Lainez-Aguirre J, Gounaris CE. Robust multi-period vehicle routing under customer order uncertainty. *Optim Res*. 2020. https://doi.org/10.1287/opre.2020.2009.
20. Bruni ME, Pugliese LDP, Beraldi P, Guerriero F. An adjustable robust optimization model for the resource-constrained project scheduling problem with uncertain activity durations. *Omega*. 2017;71:66-84.
21. Liang Z, Ning C, You F. Operational optimization of industrial steam systems under uncertainty using data-driven adaptive robust optimization. *AIChE J*. 2019;65(7):e16500.
22. Matthews LR, Gounaris CE, Kevrekidis IG. Designing networks with resiliency to edge failures using two-stage robust optimization. *Eur J Oper Res*. 2019;279(3):704-720.
23. Zhang Q, Grossmann IE, Lima RM. On the relation between flexibility analysis and robust optimization for linear systems. *AIChE J*. 2016;62(9):3109-3123.
24. Bertsimas D, Nohadani O, Teo KM. Nonconvex robust optimization for problems with constraints. *INFORMS J Comput*. 2010;22(1):44-58.
25. Zhang Y. General robust-optimization formulation for nonlinear programming. *J Optim Theory Appl*. 2007;132(1):111-124.
26. Yuan Y, Li Z, Huang B. Nonlinear robust optimization for process design. *AIChE J*. 2018;64(2):481-494.
27. Wiebe J, Cecílio I, Misener R. Robust optimization for the pooling problem. *Ind Eng Chem Res*. 2019;58:12712–12722.
28. Kammammettu S, Li Z. Two-stage robust optimization of water treatment network design and operations under uncertainty. *Ind Eng Chem Res*. 2019;59:1218–1233.
29. Matthews LR, Guzman YA, Onel O, Niziolek AM, Floudas CA. Natural gas to liquid transportation fuels under uncertainty using robust optimization. *Ind Eng Chem Res*. 2018;57(32):11112-11129.
30. Gong J, You F. Resilient design and operations of process systems: nonlinear adaptive robust optimization model and algorithm for resilience analysis and enhancement. *Comput Chem Eng*. 2018;116:231-252.
31. Mutapcic A, Boyd S. Cutting-set methods for robust convex optimization with pessimizing oracles. *Optim Methods Softw*. 2009;24(3):381-406.
32. Kelley JE Jr. The cutting-plane method for solving convex programs. *J Soc Ind Appl Math*. 1960;8(4):703-712.
33. Rahal S, Li Z, Papageorgiou DJ. Proactive and reactive scheduling of the steelmaking and continuous casting process through adaptive robust optimization. *Comput Chem Eng*. 2019;106658.
34. Avraamidou S, Pistikopoulos EN. Adjustable robust optimization through multi-parametric programming. *Optim Lett*. 2020;14(4):873-887.
35. Bertsimas D, Iancu DA, Parrilo PA. A hierarchy of near-optimal policies for multistage adaptive optimization. *IEEE Trans Autom Control*. 2011;56(12):2809-2824.
36. Snyder LV, Daskin MS. Stochastic p-robust location problems. *IIE Trans*. 2006;38(11):971-985.
37. Hart WE, Watson J-P, Woodruff DL. Pyomo: modeling and solving mathematical programs in python. *Math Program Comput*. 2011;3(3):219-260.
38. Hart WE, Laird CD, Watson J-P, et al. *Pyomo–Optimization Modeling in Python*. Vol 67. 2nd ed. New York: Springer Science & Business Media; 2017.
39. Nicholson B, Siirola JD, Watson J-P, Zavala VM, Biegler LT. Pyomo.dae: a modeling and automatic discretization framework for optimization with differential and algebraic equations. *Math Program Comput*. 2018;10(2):187-223.
40. Miller DC, Siirola JD, Agarwal D, et al. Next generation multi-scale process systems engineering framework. *Comput Aided Chem Eng*. 2018;44:2209-2214.
41. Wächter A, Biegler LT. On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming. *Math Program*. 2006;106(1):25-57.
42. Duff IS, Reid JK. *MA27-A Set of Fortran Subroutines for Solving Sparse Symmetric Sets of Linear Equations*. London: UKAEA Atomic Energy Research Establishment; 1982.
43. Misener R, Floudas C. ANTIGONE: algorithms for coNTinuous/Integer global optimization of nonlinear equations. *J Glob Optim*. 2014;59(2):503-526.
44. Tawarmalani M, Sahinidis NV. A polyhedral branch-and-cut approach to global optimization. *Math Program*. 2005;103:225-249.
45. Rooney WC, Biegler LT. Design for model parameter uncertainty using nonlinear confidence regions. *AIChE J*. 2001;47(8):1794-1804.
46. Mores P, Rodríguez N, Scenna N, Mussati S. CO_2 capture in power plants: minimization of the investment and operating cost of the post-combustion process using mea aqueous solution. *Int J Greenhouse Gas Control*. 2012;10:148-163.
47. Halemane KP, Grossmann IE. Optimal process design under uncertainty. *AIChE J*. 1983;29(3):425-433.
48. Varvarezos DK, Biegler LT, Grossmann IE. Multiperiod design optimization with SQP decomposition. *Comput Chem Eng*. 1994;18(7):579-595.
49. Rooney WC, Biegler LT. Optimal process design with model parameter uncertainty and process variability. *AIChE J*. 2003;49(2):438-449.
50. Underwood AJV. Simple Formula to Calculate Mean Temperature Difference. *Chemical Engineering*; 1970;77(13):192.
51. Hessen ET, Haug-Warberg T, Svendsen HF. The refined e-nrtl model applied to co2–h2o–alkanolamine systems. *Chem Eng Sci*. 2010;65(11):3638-3648.
52. Weiland RH, Rawal M, Rice RG. Stripping of carbon dioxide from monoethanolamine solutions in a packed column. *AIChE J*. 1982;28(6):963-973.
53. Zhang Y, Chen H, Chen C-C, Plaza JM, Dugas R, Rochelle GT. Rate-based process modeling study of co2 capture with aqueous monoethanolamine solution. *Ind Eng Chem Res*. 2009;48(20):9233-9246.
54. Chinen AS, Morgan JC, Omell B, Bhattacharyya D, Tong C, Miller DC. Development of a rigorous modeling framework for solvent-based CO_2 capture. 1. Hydraulic and mass transfer models and their uncertainty quantification. *Ind Eng Chem Res*. 2018;57(31):10448-10463.
55. Towler G, Sinnott R. *Chemical Engineering Design: Principles, Practice and Economics of Plant and Process Design*. Oxford: Elsevier; 2012.
56. Seider WD, Seader JD, Lewin DR. *Product & Process Design Principles: Synthesis, Analysis and Evaluation*. New Jersey: John Wiley & Sons; 2009.

57. Morgan JC, Bhattacharyya D, Tong C, Miller DC. Uncertainty quantification of property models: methodology and its application to CO_2-loaded aqueous mea solutions. *AIChE J*. 2015;61(6):1822-1839.
58. Cerrillo-Briones IM, Ricardez-Sandoval LA. Robust optimization of a post-combustion CO_2 capture absorber column under process uncertainty. *Chem Eng Res Des*. 2019;144:386-396.
59. Morgan JC, Chinen AS, Omell B, Bhattacharyya D, Tong C, Miller DC. Thermodynamic modeling and uncertainty quantification of CO_2-loaded aqueous mea solutions. *Chem Eng Sci*. 2017;168: 309-324.
60. Fisher KS, Beitler C, Rueter C, Searcy K, Rochelle G, Jassim M. *Integrating MEA regeneration with CO_2 compression and peaking to reduce CO_2 capture costs*. : Technical report. Trimeric Corporation; 2005.

SUPPORTING INFORMATION

Additional supporting information may be found online in the Supporting Information section at the end of this article.

How to cite this article: Isenberg NM, Akula P, Eslick JC, Bhattacharyya D, Miller DC, Gounaris CE. A generalized cutting-set approach for nonlinear robust optimization in process systems engineering. *AIChE J*. 2021;67:e17175. https://doi.org/10.1002/aic.17175