Systems & Control Transactions

Original Research Article
Peer Reviewed Conference Proceeding
ESCAPE 36 - European Symposium on Computer Aided Process Engineering
Sheffield, UK. 21-24 June 2026

PSE PRESS

Superstructure Framework for Feasibility and Flexibility Analysis Methods in Modular Plant Design

Julian Pamperin[a], Jonathan Mädler[b], Amy Koch[a], Isabel Viedth[a], and Leon Urbas[c]

[a] Dresden University of Technology, Process-to-Order Group, Dresden, Saxony, Germany
[b] FLEXCELERATE Solutions GmbH, Dresden, Saxony, Germany
[c] Dresden University of Technology, Chair of Process Control Systems, Dresden, Saxony, Germany
* Corresponding Author: julian.pamperin@mailbox.tu-dresden.de.

ABSTRACT

Modular plant design requires assessing whether independently characterized process requirements and module capabilities are compatible—a challenge that established methods address incompletely. Feasibility and flexibility analysis, as well as Quality by Design, typically assume integrated single-domain models where all variables belong to one coherent description, yet modular design involves domains that originate from different sources, evolve independently, and connect through interface variables. This work proposes Quantified Constraint Satisfaction Problems (QCSPs) as a formulation for interface-level suitability assessment: universal quantification encodes properties that must hold across their entire admissible range (e.g., physical properties, uncertain or environment-dependent characteristics requiring robustness), while existential quantification encodes variables where at least one feasible value must exist (e.g., critical process parameters, control inputs, configuration options). By operating over domain interfaces rather than requiring integrated system models, the formulation accommodates heterogeneous information sources within a common problem specification, with uncertainty handled through probabilistic intervals governed by a global confidence threshold. The framework supports both finite-domain solving on pre-evaluated datasets and numeric solving through on-demand model evaluation. A bioreactor suitability case study provides proof-of-concept, illustrating how the framework accommodates realistic problem complexity and offering initial observations on method selection trade-offs.

Keywords: Process Design, Design Under Uncertainty, Information Management, Interdisciplinary, Modelling and Simulations, Optimization

INTRODUCTION

Modular plant design has emerged as a paradigm for flexible, reconfigurable production systems in process industries facing volatile markets or rapid scale-up needs [1, 2]. By composing standardized, pre-engineered equipment units rather than designing monolithic facilities [3, 4], modular approaches have the potential to reduce engineering effort and support faster time-to-market [5]. However, this introduces a fundamental challenge: assessing whether a given module is suitable for a specific process before committing to detailed engineering [6, 7].

Two established methods offer relevant foundations. Feasibility and flexibility analysis evaluates whether process designs can maintain feasible operation under uncertainty [8, 9]. A key aspect within Quality by Design is the design space: the region of input variables and process parameters within which product quality is reliably met [10–12]. Despite their complementary strengths, neither directly addresses the module selection problem: determining whether independently characterized process requirements and module capabilities are compatible. Both typically assume integrated single-domain models, yet modular design involves domains that originate from different sources.

Bridging this gap requires confronting several challenges. First, process requirements and module capabilities may be characterized using different model types, information sources, and uncertainty representations.

Second, the interface between domains involves shared variables where semantic alignment introduces uncertainty. Third, practical suitability assessment must accommodate varying levels of detail: coarse screening across large module libraries demands different methods than detailed verification of selected candidates.

This work proposes a framework that addresses these challenges by reformulating suitability assessment as a Quantified Constraint Satisfaction Problem (QCSP) operating over domain interfaces. The framework provides a unified problem description capable of integrating diverse information types, systematic handling of uncertainties through interval specifications, and scalable assessment procedures. The following sections describe the theoretical foundations, present solution procedures, and demonstrate the framework through a case study in bioprocess equipment selection.

CONCEPTUAL FOUNDATIONS

Assessing module suitability requires clarity on what is being compared and at what level of abstraction. This section establishes the conceptual structure: domains as bodies of knowledge, their diverse characterizations, and approaches for comparison.

Domains and Abstractions

A domain denotes a bounded, abstract description space in which variables, constraints, and relations are defined—a consistent viewpoint independent of physical realization. In module-based plant design, a separation is typically made between the (equipment-) module domain and the process domain [13, 14].

Each domain may be characterized through diverse means—mechanistic models, experimental data, datasheets, simulation results, or expert knowledge—varying in fidelity, formalism, and the phenomena they capture. Crucially, characterizations from different domains are often not directly comparable: a detailed dynamic model of a bioreactor and a Quality-by-Design design space for a cultivation process do not share a common language.

Two strategies exist for enabling evaluation across domains. The first connects domain characterizations directly—for instance, coupling dynamic process and equipment models—and derives abstractions that evaluate the joint system's suitability for a given goal. This integrated approach permits analytical depth but introduces complexity: proprietary interfaces must propagate uncertainties across subsystems, and metrics must address the combined system. The second strategy, emphasized here, derives abstractions from individual domain characterizations independently, then assesses compatibility at the interface level. From process characterizations, requirements are derived (e.g., operating ranges, quality specifications); from module characterizations, capability metrics are extracted (e.g., operational envelopes). These transformations require analytical effort but yield quantities that are comparable without coupling the underlying characterizations, sacrificing analytical depth for manageability and accepting conservatism in exchange for modularity.

In practice, process and equipment characterizations are rarely created independently: experimental data emerges from specific equipment configurations, while equipment specifications are refined based on target processes. The framework accommodates such entanglement through uncertainty specifications on interface variables, allowing imperfect separation to be represented explicitly rather than assumed away.

Assessment Scope

Adopted from the concept of intra-modular safety in modular safety engineering [15, 16], this work introduces *intra-modular suitability*: the suitability of an individual module for a given process, assessed via boundary variables that characterize its interface with the environment. A module is considered intra-modularly suitable if its capability envelope covers the corresponding process requirements across all relevant boundary variables. If intra-modular suitability is established for all modules within a configuration, system-wide suitability can be inferred without further analysis.

CSP FRAMEWORK FOR SUITABILITY ASSESSMENT

The preceding conceptual foundations established the domains and the scope of the suitability assessment considered in this work. This section introduces a formal framework that operationalizes these concepts using QCSPs [17–19]. The following section relates the framework to feasibility analysis and Quality by Design, presents the advantages of the CSP-based formulation, provides the formal QCSP definition with uncertainty handling, and defines satisfaction semantics.

Relation to Feasibility Analysis and Quality by Design

The suitability assessment problem shares foundational objectives with feasibility and flexibility analysis as well as Quality by Design: ensuring constraint satisfaction under uncertainty through appropriate use of degrees of freedom. Classical feasibility and flexibility analysis evaluates whether feasible operation can be maintained for all uncertain parameters θ within a specified set T through appropriate adjustment of control variables z given constraints $g(d, z, x, \theta) \leq 0$ [8, 20, 21]. Quality by Design similarly distinguishes between material attributes and process parameters that define a design space within which product quality is assured [10, 11]. Both frameworks implicitly encode a quantifier structure analogous to that of QCSPs: in flexibility analysis, uncertain parameters θ correspond to universal quantification $\forall$ (feasibility must hold across their entire range), while control variables z correspond to existential quantification $\exists$ (finding any feasible adjustment suffices). In Quality by Design, Critical Material Attributes (CMAs) typically relate to universal quantification as properties that must be tolerated across their specified ranges, while Quality Target Product Profiles (QTPPs), Critical Quality Attributes (CQAs), and Critical Process Parameters (CPPs) relate to existential quantification as targets where achieving any value within acceptable bounds suffices.

However, neither method directly addresses the matching problem in module-based plant design. Both typically require an integrated model relating all variables within one coherent description. Performing a feasibility test for a given process-module combination therefore requires semantic alignment: the meaning encoded in design space characterizations must be transferred into a problem structure compatible with flexibility analysis. This translation is non-trivial and may need to be repeated for each combination, as these approaches are commonly implemented ad hoc rather than integrated into information model standards. While recently established standards such as VDI/VDE/NAMUR 2658 (Module Type Package) [22] or DEXPI [23] provide structured representations of equipment and process data, they do not prescribe a formal problem structure for suitability assessment. An interface language is therefore needed that encodes the semantics of feasibility and flexibility analysis and Quality by Design into a common problem specification.

Advantages of CSP-Based Formulation

QCSPs address this by providing a *common interface language* for independently characterized domain constraints. Each domain translates its characterization into QCSP-compliant interval specifications; matching then operates on these specifications without requiring knowledge of underlying models.

This interface-language property derives from several characteristics of the CSP formalism. First, CSP formulations treat all variables uniformly without requiring explicit differentiation between inputs, outputs, or other dependency structures. This uniformity shifts modeling complexity into the domain descriptions themselves, decoupling problem specification from solution strategy. The framework thus accommodates heterogeneous information sources—mechanistic models, empirical correlations, experimental datasets, manufacturer specifications—without prescribing how these sources internally relate variables.

Second, while the quantifier structure underlying feasibility and flexibility analysis and Quality by Design remains implicit in their problem formulations, the QCSP formalism makes this structure explicit and systematic. Variables are declared with their quantifiers as part of the problem specification, enabling direct expression of which conditions must be tolerated versus which represent degrees of freedom. This explicitness facilitates communication across domain boundaries and supports automated reasoning about suitability.

Third, the formulation accommodates expressiveness beyond classical approaches. Quantifier semantics may depend on assessment context: a critical process parameter such as cultivation temperature is existentially quantified for single-product assessment (one feasible setpoint suffices) but universally quantified for multi-product flexibility (all temperatures in the range must be achievable). The QCSP formulation handles such context shifts through quantifier specification rather than problem reformulation. Furthermore, discrete variables representing categorical choices (equipment variants, operating modes) integrate naturally alongside continuous variables—a capability that flexibility analysis handles less directly.

Fourth, integrating uncertainty directly into constraint specifications enables cleaner problem descriptions and more generic uncertainty handling than approaches common in flexibility analysis or Quality by Design. The formulation presented below embeds probability distributions as first-class constraint elements, with a global confidence threshold governing the overall satisfaction guarantee.

The approach localizes transformation effort to domain boundaries: while defining the translation from domain-specific characterizations to QCSP specifications requires non-trivial effort, such effort is inherent whenever knowledge from different sources must be integrated. The formalization provides a uniform interface that, once established, simplifies subsequent usage across different assessment contexts. Beyond binary satisfaction, CSP-based analysis can yield information analogous to flexibility analysis, including identification of binding constraints, feasible region characterization, and flexibility quantification through systematic constraint bound variation [24].

Quantified Constraint Satisfaction Problem Formulation

The QCSP is defined as a tuple $\langle X, D, C\rangle$ where:

- $X = \{X_1, \dots, X_n\}$ is a set of variables
- $D = \{D_1, \dots, D_n\}$ where each D_i is the domain for variable X_i, which may be continuous ($D_i \subseteq \mathbb{R}$) or discrete ($D_i = \{v_1, \dots, v_k\}$)
- $C = \{C_I, C_F, C_D, C_\alpha\}$ is a structured set of constraints

Each constraint type serves a distinct role in the problem specification: The set C_I contains interval constraints that bound individual variables:

$$C_I = \{(Q_i, X_i, B_i) : i \in \mathcal{I}\} \quad (1)$$

where $Q_i \in \{\exists, \forall\}$ denotes the quantifier specifying whether satisfaction is required for *some* (∃) or *all* (∀) values of the variable, $X_i \in X$ is the constrained variable, and B_i specifies the constraint bound. The bound B_i may take the form of either a *deterministic interval* $[l_i, u_i]$ with $l_i, u_i \in \mathbb{R}$, or a *probabilistic specification* P_i, i.e., a probability distribution from which operational bounds are derived during solving. Deterministic intervals directly restrict a variable to a known range. Probabilistic specifications, by contrast, encode uncertain knowledge about the constraint—for example, a process viscosity requirement subject to measurement or modeling uncertainty, or an equipment limit known only within confidence bounds. The translation from a probability distribution to an effective interval depends on the global confidence threshold and the solving procedure, allowing the same problem specification to induce different operational constraints under different confidence requirements.

The set C_D restricts variables to specific discrete values:

$$C_D = \{(X_k, V_k) : k \in \mathcal{K}\} \quad (2)$$

where $V_k \subset D_k$ is a finite set of allowed values for variable X_k. These constraints represent categorical choices such as equipment variants, operating modes, or material selections.

The set C_F captures relational constraints among multiple variables:

$$C_F = \{(t_j, R_j) : j \in \mathcal{J}\} \quad (3)$$

where $t_j \subseteq X$ denotes the scope of the constraint, i.e., the subset of variables involved, and $R_j \subseteq \prod_{X_k \in t_j} D_k$ defines the relation specifying the allowed combinations of values over that scope. The relation R_j may be specified explicitly (e.g., as a lookup table or pre-evaluated dataset) or implicitly (e.g., through algebraic equations, differential equations, or trained surrogate models). This flexibility accommodates the diverse information types encountered in modular plant design.

For any variable $X_i \in X$, an optional associated distribution P_{X_i} may be specified, indicating that values of X_i are known only probabilistically. The impact of such variable-level uncertainty on constraint satisfaction is governed by the global confidence threshold.

The constraint C_α specifies the required confidence level for the overall problem:

$$C_\alpha = \alpha_{\text{global}} \in [0, 1) \quad (4)$$

This threshold defines an upper bound on the probability with which a declared solution may violate one or more constraints. A high confidence requirement (e.g., $\alpha_{\text{global}} = 0.01$) demands that uncertainties be treated conservatively, yielding solutions that are highly trustworthy but potentially restrictive. Larger values of α_{global} permit more optimistic interpretations of uncertain constraints, expanding the solution space. The global confidence constraint affects all probabilistic elements in the problem: interval constraints specified as distributions ($B_i = P_i$) and variables with associated uncertainty (P_{X_i}). Deterministic constraints remain unaffected. The specific mechanism by which α_{global} propagates to individual uncertain elements—for instance, through Bonferroni correction (as described in the case study) or more sophisticated allocation schemes—is a solver concern rather than a problem specification concern. This separation ensures that the problem formulation remains independent of the solution method.

Satisfaction Semantics

The QCSP is satisfied if a strategy σ exists for existentially quantified variables such that, for any assignment to universally quantified variables, all constraints are satisfied with significance at least α_{global}.

Formally, let $X_\forall = \{X_i : \exists (Q_i, X_i, B_i) \in C_I \text{ with } Q_i = \forall\}$ denote the universally quantified variables, and $X_\exists = X \backslash X_\forall$ the existentially quantified variables. The QCSP is satisfied if and only if:

$$\mathbb{P}\left[\exists \sigma : \prod_{X_j \in X_\exists} D_j \text{ s.t. } \forall \mathbf{x}_\forall \in \prod_{X_i \in X_\forall} \tilde{B}_i : (\mathbf{x}_\forall, \sigma(\mathbf{x}_\forall)) \models C\right] \geq 1 - \alpha_{\text{global}} \quad (5)$$

where $\tilde{B}_i$ denotes the effective interval for variable X_i (either the deterministic interval $[l_i, u_i]$ or an interval derived from P_i consistent with α_{global}), and $\models C$ indicates satisfaction of all constraints in C. When $\alpha_{\text{global}} = 0$ and all constraints are deterministic, this reduces to classical QCSP semantics. When only existential quantifiers are present ($X_\forall = \varnothing$), the problem simplifies to a standard CSP.

CASE STUDY

This section demonstrates the framework through a proof-of-concept assessment of bioreactor suitability for cell cultivation processes. First, the solving procedures are presented; then, these procedures are applied to four representative scenarios.

Solving Procedures

The QCSP formulation admits various solving strategies; the methods employed here prioritize simplicity to illustrate the connection between the problem formulation and established techniques. More sophisticated solvers from flexibility analysis or constraint programming can be substituted as problem complexity increases, while the fundamental structure of QCSP solving remains unchanged.

As noted above, constraint relations may be specified through pre-evaluated datasets (finite-domain solving via direct lookup) or model structures requiring computation (numeric solving via on-demand evaluation). This distinction is decisive: these information sources entail fundamentally different solver strategies and different potentials for characterizing domain properties.

For uncertainty handling, a spectrum of approaches exists between repeated sampling from probability distributions followed by solving CSP instances, and conservative treatment through worst-case bounds on confidence intervals. The latter approach, employed here via Bonferroni correction, offers computational efficiency at the cost of conservatism. When multiple uncertain constraints must be satisfied jointly, the union bound controls overall failure probability: if α_{global} is the acceptable global significance level and N constraints carry uncertainty, allocating a budget α_i to each constraint such that $\sum_{i=1}^{N} \alpha_i = \alpha_{\text{global}}$ guarantees that the global failure probability remains below α_{global}. The simplest allocation divides α_{global} uniformly; more sophisticated schemes can assign stricter thresholds to critical constraints, enabling optimized prioritization. For a constraint with probabilistic specification (e.g., Gaussian with mean μ_i and standard deviation σ_i), the effective interval becomes $[\mu_i - z_{\alpha_i/2}\sigma_i, \mu_i + z_{\alpha_i/2}\sigma_i]$, where $z_{\alpha_i/2}$ is the critical value corresponding to the allocated local significance level. This transformation converts probabilistic constraints into deterministic intervals suitable for standard constraint checking, while preserving statistical guarantees on the overall solution.

For standard CSPs (only existential quantifiers), satisfaction requires finding at least one assignment satisfying all constraints. The solving procedure applies a search strategy appropriate to the information type: for finite-domain problems, constraint propagation successively filters candidate points, or backtracking examines candidates sequentially with early termination upon finding a valid assignment; for numeric problems, sampling-based search (e.g., random or structured sampling) generates input configurations for on-demand model evaluation. In both cases, uncertain constraints are checked against intervals derived from their associated distributions using the allocated local significance thresholds.

For QCSPs containing universal quantifiers ∀, direct verification would require checking infinitely many value combinations within ∀-intervals—an intractable task. The approach adopted here decomposes universal constraints into boundary evaluations: each ∀-constraint on interval $[l_i, u_i]$ generates sub-problems at the boundary values $\{l_i, u_i\}$, yielding $2^{|X_\forall|}$ standard CSP sub-problems. Under a continuity assumption—that feasible regions form convex sets such that boundary feasibility implies interior feasibility—satisfaction of all boundary sub-problems guarantee satisfaction of the original QCSP. This assumption holds for many physical systems but may yield false positives when solution spaces exhibit discontinuities or non-convexities; such cases require explicit mode enumeration or conservative interior sampling.

Algorithm 1 is a unified solving procedure. The algorithm accepts a QCSP specification and returns a satisfaction status. The SEARCH subroutine encapsulates information-type-specific strategies: for finite-domain problems, it implements constraint propagation or backtracking over pre-evaluated datasets; for numeric problems, it implements sampling-based search with on-demand model evaluation. The DECOMPOSE subroutine generates boundary sub-problems from universal constraints.

Algorithm 1: QCSP Solver

Input: QCSP $\langle X, D, C \rangle$ with $C = \{C_I, C_F, C_D, C_\alpha\}$
Output: Satisfaction status
$X_\forall \quad \{X_i : (Q_i, X_i, B_i) \in C_I \land Q_i = \forall\}$
$\mathcal{S} \leftarrow$ Decompose$(C_I, X_\forall)$ `// ` $2^{|X_\forall|}$ `sub-problems (1 if` $X_\forall = \emptyset$`)`
Compute α_{local} from α_{global} $|\mathcal{S}|$, and uncertain constraint count;
Derive effective intervals $\tilde{B}_i$ from probabilistic specifications using α_{local}
foreach *sub-problem* $s \in \mathcal{S}$ **do**
    if *SEARCH*$(X, D, C, \tilde{B}^{(s)}) = \mathit{unsatisfied}$ **then**
        return `unsatisfied`
return `satisfied`

Bioreactor Suitability Assessment

The suitability assessment integrates a process domain characterized by Quality-by-Design-style cultivation requirements with a module domain described by a steady-state simulation model. Four representative scenarios, selected from a broader set of 18 assessment cases, illustrate the trade-offs between finite-domain and numeric solving while validating the framework's capability to handle problems of typical complexity in this domain. The remaining cases are omitted for brevity.

The bioreactor model represents a 10–200 L stirred-tank reactor typical of pilot-scale cell culture systems. The model accepts 12 input variables spanning thermal properties, equipment parameters, operational settings, and medium physical properties. Each output incorporates Gaussian uncertainty distributions with mechanistically derived mean values and standard deviations, enabling uncertainty propagation through the assessment. Despite the 12-dimensional input space, gradient-based principal component analysis revealed an effective dimensionality of approximately 5.4 dimensions, indicating significant redundancy due to variable correlations—a

Table 1: QCSP formulation for Case 1: CHO high cell density cultivation (20–30 × 10^6 cells/mL) with high tempering capability of the bioreactor.

Variable	Unit	Interval	Quant.
Process Requirements			
Medium Temperature	°C	[36.5, 37.0]	∀
Medium Volume	m^3	[0.15, 0.17]	∀
Dynamic Viscosity	Pa·s	[0.001, 0.008]	∀
Medium Density	kg/m^3	[1010, 1040]	∀
Specific Heat Capacity	J/(kg·K)	[3800, 3900]	∀
Thermal Conductivity	W/(m·K)	[0.58, 0.62]	∀
Eff. Shear Stress	Pa	[0.0, 1.4]	∃
Mixing Time	s	[0.0, 30]	∃
Min. Spec. Heat Flow	W/kg	[−732, −2.25]	∃
Equipment Specification			
Min. Jacket Temp.	°C	[15, 25]	∃
Max. Jacket Temp.	°C	[50, 70]	∃
Wall Therm. Resistance	m^2·K/W	[0.0002, 0.0006]	∃
Jacket Heat Trans. Coeff.	W/(m^2·K)	[400, 800]	∃
Fouling Resistance	m^2·K/W	[0.00015, 0.0005]	∃
Max. Spec. Heat Flow	W/kg	[−203, 557]	∃
Agitation Speed	rpm	[20, 400]	∃

characteristic that influences data requirements for explicit capability characterization.

The cultivation scenarios were selected to represent diverse assessment outcomes: Case 1 with requirements well within equipment capabilities; Case 2 requiring significantly more data due to proximity to capability boundaries; Case 3 where requirements approach the equipment limits; and Case 4 representing genuine incompatibility. Table 1 shows the complete QCSP formulation for Case 1 as an exemplary problem specification, showing how cultivation requirements and equipment specifications are transformed into interval constraints with appropriate quantifiers. The six universally quantified variables—representing process conditions that must be satisfied across their entire specified range—yield $2^6 = 64$ subproblems per assessment. A global confidence threshold of $\alpha_{\text{global}} = 0.05$ governed uncertainty propagation via Bonferroni correction.

For finite-domain solving, a dataset of 5×10^6 uniformly random samples were generated across the 12-dimensional input domain. The solver employed hybrid constraint propagation and backtracking with early termination. The numeric solver implemented uniform random search with on-demand model evaluation. Both solvers were implemented in Julia and executed single-threaded on a Windows system with 64 GB RAM and Intel vPRO EVO Edition CPU.

The assessment outcomes are summarized in Table 2. Both methods correctly identified Case 1 and Case 2 as satisfiable and Case 4 as infeasible. However, for Case 3, the finite-domain approach produced a false negative: it failed to find satisfying assignments despite the numeric solver confirming that feasible operational configurations exist. This discrepancy arises because the requirements in Case 3 approach capability boundaries where the feasible region becomes too narrow for the pre-computed dataset to capture. The finite-domain method cannot detect feasible regions between sparse datapoints, regardless of total dataset size—a fundamental precision limitation when requirements lie near equipment limits.

The performance comparison for Cases 1 and 2 (see Table 3) reveals the trade-off structure between methods. Case 1, with requirements well within capabilities, achieved satisfaction with 5×10^5 datapoints in 5.2 s, while the numeric solver required only 64 model evaluations. Case 2, despite both methods succeeding, demanded 5×10^6 datapoints and 36.9 s for finite-domain solving—a tenfold increase in data requirements due to proximity to capability boundaries—whereas numeric solving required only 69 evaluations. The breakeven simulation time, at which both methods require equal computational effort, shifts from 0.081 s per evaluation (Case 1) to 0.535 s (Case 2). For models with evaluation times above these thresholds, pre-computed datasets offer superior performance for repeated assessments; below them, on-demand evaluation becomes more efficient. This breakeven analysis, combined with the precision limitations observed in Case 3, suggests a practical strategy: finite-domain methods provide efficient initial

Table 2: Suitability assessment results for four representative scenarios comparing finite-domain (pre-evaluated dataset) and numeric (on-demand evaluation) solving approaches. Temp. indicates the bioreactor's tempering capability configuration (high: extended jacket temperature range; low: restricted range).

Case	Cultivation	Temp.	Numeric	Finite-D.
1	CHO high cell dens.	high	satisfied	satisfied
2	E. coli low-temp expr.	high	satisfied	satisfied
3	E. coli low-temp expr.	low	satisfied	unsatisfied
4	P. pastoris methanol	low	unsatisfied	unsatisfied

Table 3: Computational performance comparison for cases where both methods found satisfaction. Datapoints: minimum dataset size required for finite-domain satisfaction; FD Time: finite-domain solving duration; Num. Evals: model evaluations required by numeric solver; Breakeven: simulation time per evaluation at which both methods require equal total computation time.

Case	Datapoints	FD Time (s)	Num. Evals	Breakeven (s)
1	5×10^5	5.20	64	0.081
2	5×10^6	36.92	69	0.535

screening when requirements lie safely within anticipated capabilities, while numeric methods become essential for boundary exploration. The framework's unified problem formulation enables seamless transition between these approaches as assessment demands evolve.

CONCLUSION

This work addressed a methodological gap in modular plant design: while feasibility and flexibility analysis provide rigorous frameworks for evaluating process robustness, and Quality by Design establishes systematic design space characterization, both traditions typically assume integrated single-domain models. Modular plant design, however, requires assessing compatibility between process requirements and module capabilities that originate from different sources, evolve independently, and connect through shared interface variables.

The proposed QCSP formulation addresses this gap primarily as a *formal interface language*. Rather than requiring integrated models that connect diverse domains manually—which demands detailed knowledge of both domain-specific model structures—the QCSP provides a common specification through which independently characterized domains can express their constraints. Each domain translates its characterization into QCSP-compliant interval specifications; the matching process then operates on these specifications without requiring knowledge of underlying model structures.

The framework accommodates heterogeneous information sources—mechanistic models, empirical datasets, expert-specified bounds—through uniform variable treatment, and supports both finite-domain solving on pre-evaluated datasets and numeric solving through on-demand model evaluation. The bioreactor case study provided proof-of-concept that the formulation can represent realistic suitability problems and that established solving techniques remain applicable.

Several limitations warrant emphasis. The case study employed an unvalidated first-principles model; industrial validation remains necessary. The solving procedures presented are illustrative rather than contributory—more sophisticated solvers from flexibility analysis or constraint programming literature can and should be substituted when problem complexity demands. Furthermore, while the framework provides mechanisms for uncertainty handling, systematic study of how uncertainty specifications affect assessment outcomes in practice was out of scope.

The breakeven thresholds (0.08–0.54 s per evaluation) illustrate trade-offs critical when assessing large module libraries. Separating problem specification from solution method enables staged workflows where inexpensive screening precedes costly verification. Capability characterizations generated during assessment can be stored and reused across future queries, amortizing effort and improving assessment efficiency over time.

OUTLOOK

The primary need is empirical validation: applying the framework to industrial module selection problems would test whether the formalization provides practical value beyond conceptual clarity. Integration with industrial information model standards could enable automated translation from existing domain characterizations to QCSP specifications, reducing the manual effort currently required; relevant standards include the Module Type Package (VDI/VDE/NAMUR 2658) for modular automation [22], DEXPI (IEC 62424) for process equipment data [23], IEC 61512 for batch control models [25], and ICH Q8–Q12 for pharmaceutical design spaces [10]. Development of accessible tooling would facilitate transition from methodological contribution to practical engineering workflow. Finally, comparative studies against established flexibility analysis tools would help quantify whether the interface-language benefits outweigh the translation overhead in realistic engineering contexts.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge the Process2Order Lab at TU Dresden for institutional support. This work is based on a diploma thesis funded and supported by FLEXCELERATE Solutions, whose input significantly contributed to its development.

REFERENCES

1. Baldea M, Edgar TF, Stanley BL, Kiss AA. Modular manufacturing processes: status, challenges, and opportunities. AIChE Journal 63:4262-4272 (2017). https://doi.org/10.1002/aic.15872
2. Becker T, Lier S, Werners B. Value of modular production concepts in future chemical industry production networks. European Journal of Operational Research 276:957-970 (2019). https://doi.org/10.1016/j.ejor.2019.01.066
3. Hohmann L, Kössl K, Kockmann N, Schembecker G, Bramsiepe C. Modules in process industry ? a life cycle definition. Chemical Engineering and Processing: Process Intensification 111:115-126 (2017). https://doi.org/10.1016/j.cep.2016.09.017
4. Lier S, Paul S, Ferdinand D, Grünewald M. Modulare verfahrenstechnik: apparateentwicklung für wandlungsfähige produktionssysteme. Chemie Ingenieur Technik 88:1444-1454 (2016). https://doi.org/10.1002/cite.201600015
5. O'Connor JT, Kowall CP, Haapala KR, Agrawal NV, Paul BK. Specialty chemicals production case study: economic analysis of modular chemical process intensification versus conventional stick?built approaches. J Adv Manuf & Process 3: (2021). https://doi.org/10.1002/amp2.10102
6. Schindel , Polyakova , Harding , Weinhold , Stenger , Grünewald , Bramsiepe . General approach for technology and process equipment assembly (PEA) selection in process design. Chemical Engineering and Processing - Process Intensification 159:108223 (2021). https://doi.org/10.1016/j.cep.2020.108223
7. Harding D, Polyakova M, Nowara D, Rech S, Grünewald M, Bramsiepe C. New process function-based selection and configuration methodology for process equipment assemblies (peas) exemplified on the unit operation distillation. Chemical Engineering and Processing - Process Intensification 168:108531 (2021). https://doi.org/10.1016/j.cep.2021.108531
8. Swaney RE, Grossmann IE. An index for operational flexibility in chemical process design. part I: formulation and theory. AIChE Journal 31:621-630 (2004). https://doi.org/10.1002/aic.690310412
9. Bansal V, Perkins JD, Pistikopoulos EN. Flexibility analysis and design using a parametric programming framework. AIChE Journal 48:2851-2868 (2004). https://doi.org/10.1002/aic.690481213
10. International Council for Harmonisation, "ICH Quality Guidelines Q8–Q12, " 2005–2019. Available: https://www.ich.org
11. Yu LX, Amidon G, Khan MA, Hoag SW, Polli J, Raju GK, Woodcock J. Understanding pharmaceutical quality by design. AAPS J 16:771-783 (2014). https://doi.org/10.1208/s12248-014-9598-3
12. Holden JM. Many lives, many masters, by brian L. weiss. J Near-Death Stud 11:57-61 (1992). https://doi.org/10.1007/bf01082738
13. Maedler, J., I. Viedt, J. Lorenz, and L. Urbas, "Requirements to a digital twin-centered concept for smart manufacturing in modular plants considering distributed knowledge, " in Computer Aided Chemical Engineering, vol. 49, Elsevier, 2022, pp. 1507–1512. doi: 10.1016/B978-0-323-85159-6.50251-7
14. Koch A, Hamedi N, Furtner L, Kock T, Klose A, Mädler J. Standards for information models considering knowledge distribution in modular plants. 2023 IEEE 21st International Conference on Industrial Informatics (INDIN) :1-7 (2023). https://doi.org/10.1109/indin51400.2023.10218218
15. Klose A, Merkelbach S, Menschner A, Hensel S, Heinze S, Bittorf L, Kockmann N, Schäfer C, Szmais S, Eckert M, Rüde T, Scherwietes T, da Silva Santos P, Stenger F, Holm T, Welscher W, Krink N, Schenk T, Stutz A, Maurmaier M, Stark K, Hoernicke M, Unland S, Erben S, Kessler F, Apitz F, Urbas L. Orchestration requirements for modular process plants in chemical and pharmaceutical industries. Chem Eng & Technol 42:2282-2291 (2019). https://doi.org/10.1002/ceat.201900298
16. VDI 2776-3:2024, Process engineering plants – Modular plants – Safety of modular plants, Jan. 2024.

17. Gent, I. P., P. Nightingale, and K. Stergiou, "QCSP-Solve: A Solver for Quantified Constraint Satisfaction Problems, " 2005.
18. Mamoulis N, Stergiou K. Algorithms for quantified constraint satisfaction problems. Lecture Notes in Computer Science :752-756 (2004). https://doi.org/10.1007/978-3-540-30201-8_60
19. Qureshi AJ, Dantan JY, Bruyere J, Bigot R. Set based robust design of mechanical systems using the quantifier constraint satisfaction algorithm. Engineering Applications of Artificial Intelligence 23:1173-1186 (2010). https://doi.org/10.1016/j.engappai.2010.02.003
20. Grossmann IE, Floudas CA. Active constraint strategy for flexibility analysis in chemical processes. Computers & Chemical Engineering 11:675-693 (1987). https://doi.org/10.1016/0098-1354(87)87011-4
21. Floudas CA, Gümü? ZH, Marianthi G Ierapetritou . Global optimization in design under uncertainty: feasibility test and flexibility index problems. Ind. Eng. Chem. Res. 40:4267-4282 (2001). https://doi.org/10.1021/ie001014g
22. VDI/VDE/NAMUR 2658-4:2022, Automation engineering of modular systems in the process industry – Modellierung von Moduldiensten, Oct. 2022.
23. Tolksdorf G, Cameron DB, Theißen M. DEXPI 2.0: synergistic integration of PFD and p&id in a unified digital model. Chemie Ingenieur Technik 97:1065-1069 (2025). https://doi.org/10.1002/cite.70009
24. Hu J, Aminzadeh M, Wang Y. Searching feasible design space by solving quantified constraint satisfaction problems. Journal of Mechanical Design 136: (2013). https://doi.org/10.1115/1.4026027
25. IEC 61512-1, Batch control – Part 1: Models and terminology, 1997

© 2026 by the authors. Licensed to PSEcommunity.org and PSE Press. This is an open access article under the creative commons CC-BY-SA licensing terms. Credit must be given to creator and adaptations must be shared under the same terms. See https://creativecommons.org/licenses/by-sa/4.0/