Searching Feasible Design Space by Solving Quantified Constraint Satisfaction Problems

Jie Hu
The State Key Laboratory of Advanced Design and Manufacturing for Vehicle Body,
Hunan University, Changsha,
Hunan 410082, China
e-mail: hu_jie@hnu.edu.cn

Masoumeh Aminzadeh
Multiscale Systems Engineering Research Group,
George W. Woodruff School of Mechanical Engineering,
Georgia Institute of Technology,
Atlanta, GA 30332

Yan Wang[1]
Multiscale Systems Engineering Research Group,
George W. Woodruff School of Mechanical Engineering,
Georgia Institute of Technology,
Atlanta, GA 30332
e-mail: yan.wang@me.gatech.edu

In complex systems design, multidisciplinary constraints are imposed by stakeholders. Engineers need to search feasible design space for a given problem before searching for the optimum design solution. Searching feasible design space can be modeled as a constraint satisfaction problem (CSP). By introducing logical quantifiers, CSP is extended to quantified constraint satisfaction problem (QCSP) so that more semantics and design intent can be captured. This paper presents a new approach to formulate searching design problems as QCSPs in a continuous design space based on generalized interval, and to numerically solve them for feasible solution sets, where the lower and upper bounds of design variables are specified. The approach includes two major components. One is a semantic analysis which evaluates the logic relationship of variables in generalized interval constraints based on Kaucher arithmetic, and the other is a branch-and-prune algorithm that takes advantage of the logic interpretation. The new approach is generic and can be applied to the case when variables occur multiple times, which is not available in other QCSP solving methods. A hybrid stratified Monte Carlo method that combines interval arithmetic with Monte Carlo sampling is also developed to verify the correctness of the QCSP solution sets obtained by the branch-and-prune algorithm. [DOI: 10.1115/1.4026027]

Keywords: quantified constraint satisfaction problem, feasible solution set, generalized interval, chassis design

1 Introduction

In the process of engineering design, designers face various requirements and specifications for products, such as functionalities, material properties, subsystems behaviors, manufacturing capabilities, financial budgets, and others. Constraint-based approaches thus can be employed to formulate a design problem as a CSP. A CSP is a system of constraints where the variables are within certain domains. The solution of a CSP is a set of feasible values for the design variables that satisfy all of the design constraints. With its concise and formal representation, the CSP formulation has been applied in floor plan design [1,2], geometric modeling [3], conceptual design [4,5], embodiment design [6], collaborative design [7–9], design space searching [10–12], and other design problems. The CSP formulation can help engineers to explore the design space and find feasible solutions that satisfy the given constraints. This is different from the optimization problem, which is to find the optimum among the feasible solutions. CSPs thus provide an overview of possible choices for design engineers before they perform optimization.

However, the traditional CSP formulation can only express limited semantics. All of the variables in CSPs are existentially quantified ($\exists$) in the sense of first-order logic and predicate. Using a simple example to illustrate, we have constraints $y = z$ and $x \neq z$ with variables in discrete domains $x \in \{1,2,3\}$, $y \in \{1,2\}$, and $z \in \{1,2\}$. The possible solutions for (x,y,z) are (1,2,2), (2,1,1), (3,2,2), and (3,1,1) based on the existentially quantified variables with the semantics of $\exists x \in \{1,2,3\}$, $\exists y \in \{1,2\}$, and $\exists z \in \{1,2\}$, as in the traditional CSPs. Suppose that the variable x is controllable or adjustable, whereas y and z are not. We want to ensure that given any y and z within the respective domains, the satisfaction of constraints is guaranteed. Then the universal quantifier ($\forall$) can be applied to y and z with the semantics of $\exists x \in \{1,2,3\}$, $\forall y \in \{1,2\}$, and $\forall z \in \{1,2\}$. In this case, the solutions will be (3,1,1) and (3,2,2). Therefore, it is necessary to have a more general formulation in which both universally and existentially quantified variables can be included to express more semantics. QCSP is an extension of CSP and allows both universal and existential quantifiers to be associated with variables [13]. QCSP is a general problem with its solutions that satisfy all constraints in the form of both mathematical and logic expressions.

In engineering design, QCSP can be used to integrate design intent of engineers into calculations by assigning quantifiers to variables. Two types of variables are differentiated. One is design parameter, and the other is target performance. The design parameters which are not controllable by engineers are associated with quantifier $\forall$. They usually correspond to the disturbances of a system or a model. The parameters which can be controlled and modified within prescribed domains are associated with quantifier $\exists$. If every value in the target performance domain should be realized, it should be associated with $\forall$. On the other hand, if the values only need to be included in the domain, it is associated with $\exists$. All values in the domains of universally quantified variables have to satisfy the constraints, whereas at least one value in each of the domains of existentially quantified variables has to satisfy the constraints.

By incorporating quantifiers, QCSP formulation can capture semantics related to material properties and process sequences. However, when all variables in a constraint are universally quantified, or constraints are overly restrictive on variable domains, we may not be able to find feasible solutions. From the perspective of semantics, CSP can be regarded as a special case of QCSP, in which all variables are existentially quantified. In this paper, we

[1]Corresponding author.

Contributed by the Design Automation Committee of ASME for publication in the Journal of Mechanical Design. Manuscript received June 28, 2012; final manuscript received November 6, 2013; published online December 31, 2013. Assoc. Editor: Karthik Ramani.

only consider QCSPs that have both universal and existential quantifiers involved. Different quantifier assignments to variables in QCSPs will result in different solution spaces.

With the advantage of quantified variables, QCSPs have recently been applied in solving mechanical design problems [14–18]. Additionally they have been applied in the areas of control [19–23], scheduling [24,25], and planning [26,27]. Both discrete and continuous values can be used in QCSP formulation. For the continuous case, interval is typically used as a set of values. As an extension of the classical set-based interval, generalized interval [28,29] has been proposed to incorporate logic quantifiers with the concept of modality. It has more semantics than the classical interval. A generalized interval is defined as a pair of numbers, instead of a set of numbers. The pair is no longer restricted to be ordered. For example, both [1,2] (which is called proper) and [2,1] (which is called improper) are valid generalized intervals. Suppose **a** is a design parameter as given, **x** is an unknown design variable, and **b** is the target performance. Their respective interval values are $\mathbf{a}=[1,4]$, $\mathbf{x}=[2,3]$, $\mathbf{a}+\mathbf{x}=\mathbf{b}=[3,7]$. The relationship can be interpreted as $\forall a\in[\![1,4]\!]$, $\forall x\in[\![2,3]\!]$, $\exists b\in[\![3,7]\!]$, such that $a+x=b$. This scenario corresponds to the case where **a** and **x** are uncontrollable and are associated with $\forall$, whereas the resulted range of **b** is guaranteed to enclose all possible values of $a+x$. In another scenario, $\mathbf{a}=[1,4]$, $\mathbf{x}=[3,2]$, $\mathbf{a}+\mathbf{x}=\mathbf{b}=[4,6]$. The interpretation is $\forall a\in[\![1,4]\!]$, $\exists x\in[\![2,3]\!]$, $\exists b\in[\![4,6]\!]$, such that $a+x=b$. It implies that **x** has become a controllable design variable. The modality of generalized interval provides the convenience for numerical calculations and logic interpretations in QCSP formulation.

Generalized interval with logic interpretations provides the semantic capability of capturing design intent. For instance, in product family design [30,31], the products share a platform but have specific features and functionalities required by different customer groups. There are two sets of design parameters, which are associated with different design intent. One is called common parameter, which embraces the commonality and modularization among different variants. The other is called scaled parameter to "scale" or "stretch" the platform to satisfy a range of performance requirements. Here, we use an example of electric motors [31,32] to illustrate how generalized interval with logic quantifier is applied in product family design. The motors need to satisfy a range of torque requirements (T). The design variables, including armature wire area (A_{wa}), field wire area (A_{wf}), the number of turns of field wire (N_s), and the stack length of the motor (L), are selected as common parameters, whereas the radius of the motor (r), thickness (t), number of turns of the armature wire (N_c), and current (I) are scaled parameters. The common parameters are fixed in the platform but with an allowance of 10% variation. Based on the values of common parameters, appropriate values of the scaled parameters should be selected to satisfy the target requirement of torque. At the early design stage, the requirement is vague and can have a wide range of values. Thus, the desirable interpretation for the QCSP solution is ($\forall A_{wa}$, $\forall A_{wf}$, $\forall N_s$, $\exists T$, $\exists r$, $\exists t$, $\exists N_c$, $\exists I$, $\exists L$) (all constraints in Ref. [31] should be satisfied) (case I). When more information about the target requirements for T becomes available, all values in the updated domain of T have to be fulfilled. The desirable interpretation becomes ($\forall A_{wa}$, $\forall A_{wf}$, $\forall N_s$, $\forall \boldsymbol{L}$, $\forall T$, $\exists r$, $\exists t$, $\exists N_c$, $\exists I$) (all the constraints should be satisfied) (case II). In a yet different case, we may find that not all the requirements of T can be fulfilled for all possible value combinations of scaled parameters. The solution should then be described by ($\forall A_{wa}$, $\forall A_{wf}$, $\forall N_s$, $\forall r$, $\forall t$, $\forall N_c$, $\forall I$, $\forall L$, $\exists T$) (all the constraints should be satisfied) (case III). Under different circumstances, the semantic differentiation for feasible solution sets by logic quantifiers provides engineers more information about the nature of the problem. Such design intent is captured by the QCSP formulation based on generalized interval.

In this paper, we present a new approach to formulate and solve feasible design problems as QCSPs. As a result, the modeling and solving processes are much simplified compared to CSP formulations. Generalized interval is used to represent the quantified variables in the QCSP formulation. Semantic analysis is applied to provide logic interpretation for the QCSP. Based on the logic interpretation, a set of consistency tests are developed to identify the solutions. A branch-and-prune algorithm is developed to provide the inner estimation of the solution to the defined QCSP. A hybrid stratified Monte Carlo approach is proposed to verify the solutions obtained from the branch-and-prune algorithm, which is much more efficient than the traditional Monte Carlo method. Our formulation is general without the restriction on the number of occurrences for variables and can be used in different forms of constraints, linear or nonlinear, analytical or reduced-order. The only requirement is that the constraints can be expressed analytically and logic interpretation is available.

In the remainder of the paper, the background of solving CSPs and QCSPs as well as generalized interval is introduced in Sec. 2. The developed semantics-based branch-and-prune algorithm to obtain the inner estimation of solutions for QCSPs is presented in Sec. 3. In Sec. 4, the proposed interval-based hybrid stratified Monte Carlo method is described and used to verify the feasible solutions provided by the new approach in Sec. 3. In Sec. 5, a numerical example of vehicle chassis design is presented to demonstrate the proposed formulation and numerical algorithms.

2 Background

In this section, the methods of solving CSPs and QCSPs are reviewed. Generalized interval and its algebraic and logic properties are introduced.

2.1 Solving CSPs.

The commonly used methods to solve CSPs with variables in discrete domains include arc-consistency techniques [34] that eliminate inconsistent values of variables in binary relationships, and search systematically through the possible assignments of values. The backtracking [35], forward checking [36], and maintaining arc consistency [37] are three classical search algorithms with different treatments of value assignment. For CSPs with variables in continuous domains, constraint propagation [38] iteratively reduces either constraints or domains to simpler ones. The hull-consistency method [39,40] approximates the solution set by decomposing constraints into primitive ones that contain only one operation and computing the interval result with fixed-point iterations. The box-consistency method [39,40] evaluates the lower and upper bounds of variables by searching the zeros of interval constraints via the interval Newton method. The (3,2)-relational consistency method [12] approximates the regions of feasible solutions in the form of 2^k-tree with two variables. The solution of constraints is found by intersection operation or composition of individual constraint solutions. A polytope-based representation was also proposed to describe and visualize the solution space [10].

2.2 Solving QCSPs.

Solving a QCSP is to find all possible values of variables that satisfy the quantified constraints. The early methods to solve QCSPs with discrete variables were extended from those for CSPs, which mainly focus on binary constraints [41–44]. Later, some methods were developed specifically for QCSPs. The quantifier elimination approach transforms quantified constraints to the equivalent quantifier-free conjunction/disjunction constraints [45–47]. More recently, QCSPs are formulated with interval-valued variables to represent continuous domains and the solving algorithms are based on interval arithmetic. An algebraic approach to compute the inner estimation of solution sets for linear interval constraints was developed [48–50]. An inner estimation of a solution set is a set of values all of which are guaranteed to be the solutions. The basic idea is searching the algebraic solution such that constraints are satisfied. The value y is called an *algebraic solution* to the constraint $f(x)=d$ if $f(y)=d$. A series of numerical estimation algorithms for different cases of

QCSPs were developed, including constraints where all universal quantifiers precede existential ones [51,52], constraints with shared existential variables [53], linear and nonlinear constraints [54,55], and constraints where existential quantifiers precede all universal ones [56]. Quantified set inversion method [21,22] provides three computing rules to solve an interval QCSP by combining the set inversion technique and modal interval [28].

Despite the effectiveness of existing algorithms to solve QCSPs, there are still some unresolved issues. For instance, the typical assumption is that existential quantified variables occur only once in all constraints. Some approaches such as quantifier elimination and hull consistency methods rely on the constraint decomposition procedure, which significantly increases the number of constraints. Moreover, the solving process by using consistency techniques may become complex. For example, arc-consistency [41,43] is effective for binary constraints, whereas its generalization to nonbinary constraints is far more challenging. Hull-consistency has no guarantee of optimality when variables have multiple occurrences [40]. Box-consistency will be inefficient in the case that the search space is large, although it can handle the constraints with multiple variable occurrences.

2.3 Generalized Interval. Generalized interval [28,29] with the Kaucher arithmetic [33] is an algebraic and semantic extension of the classical interval [57]. The classical interval is defined as a set of real numbers with lower and upper bounds, i.e., $[\![\underline{x},\overline{x}]\!] := \{x \in \mathbb{R} | \underline{x} \le x \le \overline{x}\}$. In contrast, a generalized interval $\mathbf{x} := [\underline{x},\overline{x}] \in \mathbb{KR}$, defined by a pair of numbers, is no longer restricted to the ordered bounds as $\underline{x} \le \overline{x}$. $\mathbb{KR}$ denotes the set of generalized intervals. An operator Δ maps a generalized interval $\mathbf{x}$ to a classical interval, defined as

$$[\underline{x},\overline{x}]^{\Delta} := \begin{cases} [\![\underline{x},\overline{x}]\!], & \text{if} \quad \underline{x} \le \overline{x} \\ [\![\overline{x},\underline{x}]\!], & \text{if} \quad \underline{x} \ge \overline{x} \end{cases} \tag{2.1}$$

For example, $[-2,1]^{\Delta} = [\![-2,1]\!]$, and $[2,-1]^{\Delta} = [\![-1,2]\!]$.

$\mathbf{x}$ is *proper* when $\underline{x} \le \overline{x}$ and denoted as $\mathbf{x} \in \mathbb{IR}$. $\mathbf{x}$ is *improper* when $\underline{x} \ge \overline{x}$ and denoted as $\mathbf{x} \in \overline{\mathbb{IR}}$. Pointwise interval $\mathbf{x}$ can be either proper or improper when $\underline{x} = \overline{x}$. Operators pro and imp return proper and improper intervals respectively and are defined as

$$\text{pro}[\underline{x},\overline{x}] := [\min(\underline{x},\overline{x}), \quad \max(\underline{x},\overline{x})] \tag{2.2}$$

$$\text{imp}[\underline{x},\overline{x}] := [\max(\underline{x},\overline{x}), \quad \min(\underline{x},\overline{x})] \tag{2.3}$$

The relationship between proper and improper intervals is established by an operator *dual*, defined as $\text{dual}[\underline{x},\overline{x}] := [\overline{x},\underline{x}]$. The inclusion relationship $\subseteq$ between two generalized intervals $\mathbf{x} = [\underline{x},\overline{x}]$ and $\mathbf{y} = [\underline{y},\overline{y}]$ is defined as

$$[\underline{x},\overline{x}] \subseteq [\underline{y},\overline{y}] \Leftrightarrow \underline{y} \le \underline{x} \wedge \overline{x} \le \overline{y} \tag{2.4}$$

The inclusion isotonicity is a fundamental property of the Kaucher arithmetic, which is expressed as

$$\mathbf{y}_1 \subseteq \mathbf{x}_1, \quad \mathbf{y}_2 \subseteq \mathbf{x}_2 \Rightarrow \mathbf{y}_1 \circ \mathbf{x}_1 \subseteq \mathbf{y}_2 \circ \mathbf{x}_2 \tag{2.5}$$

where $\circ \in \{+,-,\times,\div\}$, and $\mathbf{x}_1$, $\mathbf{x}_2$, $\mathbf{y}_1$, $\mathbf{y}_2$ are generalized intervals.

The intersection between $\mathbf{x}$ and $\mathbf{y}$ is generally defined as

$$[\underline{x},\overline{x}] \cap [\underline{y},\overline{y}] := [\max(\underline{x},\underline{y}), \min(\overline{x},\overline{y})] \tag{2.6}$$

which also holds for the intersection of classical intervals, specifically defined as

$$[[\underline{x},\overline{x}]] \cap [[\underline{y},\overline{y}]] := \left\{ r | r \in [[\underline{x},\overline{x}]] \wedge r \in [[\underline{y},\overline{y}]] \right\} \tag{2.7}$$

For example, $[\![-2,1]\!] \cap [\![2,4]\!] = \phi$. However, $[-2,1] \cap [2,4] = [2,1]$. The width of a generalized interval $\mathbf{x} = [\underline{x},\overline{x}]$ is defined by $\text{wid}\mathbf{x} := |\overline{x} - \underline{x}|$. The center is found by $\text{mid}\mathbf{x} := (\overline{x} + \underline{x})/2$.

In contrast to the semigroup of classical interval, generalized interval forms a group with invertibility, which is maintained by the dual operation. For example, $[2,3] - \text{dual}[2,3] = 0$, compared to $[2,3] - [2,3] = [-1,1] \ne 0$ as in the classical interval arithmetic. For the equation $[2,3] + \mathbf{x}_2 = [6.9]$, the algebraic solution is obtained as $\mathbf{x}_2 = [6,9] - \text{dual}[2,3] = [4,6]$. In contrast, the direct subtraction gives $x_2' = [6,9] - [2,3] = [3,7]$ as in the classical interval arithmetic. Here, x_2' is an overestimated value of x_2 instead of an algebraic solution. The original value of summation is not obtained if x_2' is plugged back into the original equation $\mathbf{z} = \mathbf{x}_1 + \mathbf{x}_2$, as $[2,3] + [3,7] = [5,10] \ne [6,9]$.

Generalized interval has more semantic power than classical interval, because each $\mathbf{x} \in \mathbb{KR}$ has an associated logical quantifier $Q_x \in \{\forall,\exists\}$, either universal ($\forall$) or existential ($\exists$). If $z = f(x)$ where $z \in \mathbb{R}$ and $x = (x_1,\ldots,x_n) \in \mathbb{R}^n$ is extended to its interval extension $\mathbf{z} = F(\mathbf{x})$, $\mathbf{z}$ is a collection of the values of f over the interval $\mathbf{x}$. If F has a semantic relationship among the interval variables as

$$(Q_{x_1} x_1 \in \mathbf{x}_1^{\Delta}) \cdots (Q_{x_n} x_n \in \mathbf{x}_n^{\Delta})(Q_z z \in \mathbf{z}^{\Delta})(z = f(x)) \tag{2.8}$$

Then, we say $\mathbf{z} = F(\mathbf{x})$ is interpretable. In this paper, only the solution sets in which all universal quantifiers precedes the existential ones in the logic interpretation are considered. For example, $\mathbf{z} = \mathbf{x}_1 + \mathbf{x}_2$, where $\mathbf{x}_1 = [2,3]$, $\mathbf{x}_2 = [4,6]$, and $\mathbf{z} = [6.9]$, is interpreted as $(\forall x_1 \in [2,3]^{\Delta})(\forall x_2 \in [4,6]^{\Delta})(\exists z \in [6,9]^{\Delta})(z = x_1 + x_2)$.

Three types of solutions for the same mathematical problem can be differentiated by introducing quantifiers [49,50]. They are *united solution*, *tolerable solution*, and *controllable solution*. Specifically, for constraint $f(a,x) = b$, where a and b are known variables and x is unknown, its united solution set is defined as $\Sigma_{\exists\exists} := \{x | (\exists a \in [\![\underline{a},\overline{a}]\!])(\exists b \in [\![\underline{b},\overline{b}]\!]) f(a,x) = b)\}$, the tolerable solution set is $\Sigma_{\forall\exists} := \{x | (\forall a \in [\![\underline{a},\overline{a}]\!])(\exists b \in [\![\underline{b},\overline{b}]\!]) f(a,x) = b)\}$, and the controllable solution set is $\Sigma_{\exists\forall} := \{x | (\forall b \in [\![\underline{b},\overline{b}]\!])(\exists a \in [\![\underline{a},\overline{a}]\!]) f(a,x) = b)\}$, where $[\![\underline{a},\overline{a}]\!]$ and $[\![\underline{b},\overline{b}]\!]$ are the possible ranges of the continuous values. The three different solution sets can arise at different design stages. The united one provides a broad set of feasible solutions in the very beginning of design process when many options are available such as in set based design [58] and Case I of the product family design example in Sec. 1. Controllable and tolerable solution sets can be applied when more information or requirement is available as the design process proceeds to the detailed design stage, which correspond to cases II and III of the product family design example in Sec. 1, respectively. The collection or union of feasible solutions that satisfy all constraints forms the feasible design space.

3 The Proposed Semantic Approach for Searching Feasible Solution Sets

The proposed approach for solving QCSPs includes two components. One is the semantic analysis, and the other is a branch-and-prune algorithm. The semantic analysis makes use of the logic semantics embedded in generalized intervals to interpret the defined QCSP for searching the inner estimations of its feasible solution sets. The branch-and-prune algorithm based on Kaucher arithmetic is to find the inner estimation of the feasible subspace by searching through the given design space in the form of continuous domains or intervals.

The general process of formulating and solving a QCSP for feasible design solutions starts from a numerical CSP, which is specified by mathematical constraints, variables, and their respective domains. According to the assumed design scenarios or desirable interpretation, quantifiers are assigned to the variables. Based on the interpretation rules described in Sec. 3.1, constraints are interpreted manually. Therefore, the numerical CSP is reformulated as a QCSP with logical quantified variables. Then, the problem is solved by the proposed branch-and-prune algorithm, in which Theorems 1–3 in Sec. 3.2 are used as solution identification tests, and Theorem 4 is applied in the pruning test. Iteratively the inner estimations of possible solutions are found. The union of them is the final result for feasible design space. The details of these steps will be described in the following subsections.

3.1 Semantic Analysis. The constraint system of a QCSP is written as

$$F(\mathbf{a}, \mathbf{x}) = \mathbf{b} \tag{3.1}$$

where F: $\mathbb{KR}^l \times \mathbb{KR}^n \rightarrow \mathbb{KR}^m$. The elements of $\mathbf{a} \in \mathbb{KR}^l$ are generalized intervals. With the notation ρ for proper and ι for improper, a proper interval vector $\mathbf{a}_\rho \in \mathbb{IR}^l$ with its ith element $(\mathbf{a}_\rho)_i$ and an improper interval vector $\mathbf{a}_\iota \in \overline{\mathbb{IR}}^l$ with its ith element $(\mathbf{a}_\iota)_i$ are defined as

$$(\mathbf{a}_\rho)_i := \begin{cases} \mathbf{a}_i, & \text{if} \quad \underline{a}_i \le \bar{a}_i \\ 0, & \text{otherwise} \end{cases}, \quad (\mathbf{a}_\iota)_i := \begin{cases} \mathbf{a}_i, & \text{if} \quad \underline{a}_i \ge \bar{a}_i \\ 0, & \text{otherwise} \end{cases} \tag{3.2}$$

such that $\mathbf{a} = \mathbf{a}_\rho + \mathbf{a}_\iota$. Similarly, we denote $\mathbf{b} = \mathbf{b}_\rho + \mathbf{b}_\iota$ and $\mathbf{x} = \mathbf{x}_\rho + \mathbf{x}_\iota$. The solution set of Eq. (3.1) denoted by Σ is interpreted either as

$$\begin{aligned} &(\forall x_\rho \in \mathbf{x}_\rho^\Delta)(\forall a_\rho \in \mathbf{a}_\rho^\Delta)(\forall b_\iota \in \mathbf{b}_\iota^\Delta)(\exists x_\iota \in \mathbf{x}_\iota^\Delta) \\ &(\exists a_\iota \in \mathbf{a}_\iota^\Delta)(\exists b_\rho \in \mathbf{b}_\rho^\Delta) f(a_\rho + a_\iota, x_\rho + x_\iota) = b_\rho + b_\iota \end{aligned} \tag{3.3}$$

or as

$$\begin{aligned} &(\forall x_\iota \in \mathbf{x}_\iota^\Delta)(\forall a_\iota \in \mathbf{a}_\iota^\Delta)(\forall b_\rho \in \mathbf{b}_\rho^\Delta)(\exists x_\rho \in \mathbf{x}_\rho^\Delta) \\ &(\exists a_\rho \in \mathbf{a}_\rho^\Delta)(\exists b_\iota \in \mathbf{b}_\iota^\Delta) f(a_\rho + a_\iota, x_\rho + x_\iota) = b_\rho + b_\iota \end{aligned} \tag{3.4}$$

A simple example of liner equations is used to illustrate the interpretations in Eqs. (3.3) and (3.4). Suppose $\mathbf{a}_1\mathbf{x}_1 = \mathbf{b}_1$ and $\mathbf{a}_2\mathbf{x}_1 + \mathbf{a}_1\mathbf{x}_2 = \mathbf{b}_2$, in which $\mathbf{x}_1, \mathbf{a}_1, \mathbf{b}_1, \mathbf{b}_2 \in \mathbb{IR}$, $\mathbf{x}_2, \mathbf{a}_2 \in \overline{\mathbb{IR}}$. Thus, we have $\mathbf{a} = (\mathbf{a}_1, \mathbf{a}_2) = \mathbf{a}_\rho + \mathbf{a}_\iota$ in which $\mathbf{a}_\rho = (\mathbf{a}_1, 0)$ and $\mathbf{a}_\iota = (0, \mathbf{a}_2)$. Its solution set can be interpreted as $(\forall x_1 \in \mathbf{x}_1^\Delta)(\forall a_1 \in \mathbf{a}_1^\Delta)(\exists x_2 \in \mathbf{x}_2^\Delta)(\exists a_2 \in \mathbf{a}_2^\Delta)(\exists b_1 \in \mathbf{b}_1^\Delta)(\exists b_2 \in \mathbf{b}_2^\Delta) a_1x_1 = b_1$ and $a_2x_1 + a_1x_2 = b_2$.

The interpretations of $F(\mathbf{a},\mathbf{x}) = \mathbf{b}$ are determined by the modality (proper or improper) of a generalized interval and on which side (left or right) it appears in the mathematical relation. Our convention of interpretation following the interpretation theorems of modal interval [28] is summarized as the following six rules.

R1: For variables that are located on the left side of the relation, $\forall$ is associated with proper intervals, and $\exists$ is associated with improper ones.

R2: For variables that are located on the right side of the relation, $\exists$ is associated with the proper intervals, and $\forall$ with the improper ones.

R3: A pointwise interval (e.g., [2,2] that is equivalent to a real number) is both proper and improper and can be associated with either quantifier as suited.

For variables which have multiple occurrences in a QCSP, the following rules apply.

R4: For a variable that appears multiple times all with the same quantifier $\forall$, the final one should preserve the quantifier $\forall$;

R5: For a variable that appears multiple times, all occurrences are associated with $\forall$ except for only one with $\exists$, the final interpretation of the variable should preserve the quantifier $\exists$;

R6: Variables that appear more than once with same quantifier $\exists$, the final interpretation cannot be decided. Therefore, multiple occurrences of the same existential variable should be avoided by applying the dual operation to all occurrences except for one.

Note that combining two statements with an existentially quantified variable appearing twice does not always lead to one interpretable statement. Based on the above interpretation rules, constraints will be modified, if necessary, during the semantic analysis step such that the numerical results are interpretable. The following two simple examples illustrate how the above rules are applied to ensure interpretability.

Example 1. Given $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{IR}$, $\mathbf{y} = \mathbf{x}_1(\mathbf{x}_1 + \mathbf{x}_2)$. $\mathbf{y}$ should be proper according to the Kaucher arithmetic. Because $\mathbf{x}_1$ appears twice and is associated with $\forall$, based on Rules R1, R2, and R4, the interpretation of the equation is $(\forall x_1 \in \mathbf{x}_1^\Delta)(\forall x_2 \in \mathbf{x}_2^\Delta)(\exists y \in \mathbf{y}^\Delta) y = x_1(x_1 + x_2)$.

Example 2. Given $\mathbf{x}_1 \in \overline{\mathbb{IR}}$, $\mathbf{x}_2 \in \mathbb{IR}$, $\mathbf{y} = \mathbf{x}_1(\mathbf{x}_1 + \mathbf{x}_2)$ in which $\mathbf{x}_1$ appears twice with $\exists$. Based on the Rule R6, dual is applied to the second one as $\mathbf{y} = \mathbf{x}_1(\text{dual}\mathbf{x}_1 + \mathbf{x}_2)$ so that only one $\mathbf{x}_1$ is associated with $\exists$. The multiplication of an improper interval and a proper interval can be either proper or improper, depending their specific values. For example, $\mathbf{y} = \mathbf{x}_1(\text{dual}\mathbf{x}_1 + \mathbf{x}_2) = [2,4]$ is proper if $\mathbf{x}_1 = [2,1]$ and $\mathbf{x}_2 = [1,2]$. $\mathbf{y} = [4,3]$ is improper if $\mathbf{x}_1 = [2,1]$ and $\mathbf{x}_2 = [1,1]$. Therefore, based on the Rules R1, R2, and R5, it can be interpreted either as $(\forall x_2 \in \mathbf{x}_2^\Delta)(\exists y \in \mathbf{y}^\Delta)(\exists x_1 \in \mathbf{x}_1^\Delta) y = x_1(x_1 + x_2)$ when $\mathbf{y}$ is proper, or as $(\forall x_2 \in \mathbf{x}_2^\Delta)(\forall y \in \mathbf{y}^\Delta)(\exists x_1 \in \mathbf{x}_1^\Delta) y = x_1(x_1 + x_2)$ when $\mathbf{y}$ is improper.

As a result of the semantic analysis, design constraints are modified as necessary so that they become interpretable. At the same time, different modalities can be assigned to variables to capture the intended semantics based on the desirable interpretation. Note that interpreting the multiple-occurrence variables associated with existential quantifiers requires more attentions than the ones associated with universal quantifiers.

3.2 The Branch-and-Prune Algorithm. The branch-and-prune algorithm consists of three steps: consistency test, pruning operation, and branching operation. Given a domain of variable $\mathbf{x}^*$, which can be considered as a box that is the Cartesian product of intervals, and a set of constraints $C(x)$ as the mathematical relations among variables $\mathbf{x} \in \mathbb{R}^n$, the consistency test is to check which one of the following three cases $\mathbf{x}^*$ belongs to.

Case 1: $\mathbf{x}^*$ is a *solution box* and contained in the solution set S:

$$(\forall x^* \in \mathbf{x}^{*\Delta})(C(x) \text{ is satisfied}) \Leftrightarrow \mathbf{x}^* \subseteq S$$

Case 2: $\mathbf{x}^*$ is an *infeasible box* that contains no solution and excluded from the solution set S:

$$(\forall x^* \in \mathbf{x}^{*\Delta})(C(x) \text{ is satisfied}) \Leftrightarrow (\mathbf{x}^* \cap S) = \phi$$

Case 3: $\mathbf{x}^*$ contains at least a solution:

$$(\forall x^* \in \mathbf{x}^{*\Delta})(C(x) \text{ is satisfied}) \Leftrightarrow (\mathbf{x}^* \cap S) \ne \phi$$

In case 1, $\mathbf{x}^*$ is declared as a subset of the solution set. In case 2, $\mathbf{x}^*$ is discarded by the pruning operation. In case 3, $\mathbf{x}^*$ is branched into smaller boxes, and each of those boxes is saved to be processed in the later iterations. Numerically, the exact solution set can only be approximated by a union of solution boxes that belong to case 1. The union is an inner estimation of the solution set. If we can find as many of these solution boxes as possible, the union of them will be close to the exact solution set. The inner estimation provides a sound estimation of the solution set. An estimation is called *sound* if all solutions that it contains are guaranteed to be feasible. In contrast, an estimation is called *complete* if all feasible solutions of the constraint satisfaction problem are guaranteed to be included in the estimation, which is also called an outer estimation. An outer estimation does not underestimate the true solution set, whereas an inner estimation does not overestimate.

The inclusion isotonicity [59] is fundamental for the consistency test, which states that if $F = F(\mathbf{x}_1, ..., \mathbf{x}_n)$ is an interval extension of function f and $\text{pro}\mathbf{y}_i \subseteq \text{pro}\mathbf{x}_i$ for $i = 1,\ldots,n$, then $F(\text{pro}\mathbf{y}_1, ..., \text{pro}\mathbf{y}_n) \subseteq F(\text{pro}\mathbf{x}_1, ..., \text{pro}\mathbf{x}_n)$.

The consistency test is based on the following four theorems. Consider QCSPs with constraints in the form of Eq. (3.1) and an interval box $\mathbf{x}^* \in \mathbb{KR}^n$.

THEOREM 1. *Given* $\mathbf{x}^*$, *if* $\mathbf{b}^* = F(\mathbf{a}, \mathbf{x}^*)$ *and it can be interpreted as*

$$(\forall x^* \in \mathbf{x}^{*\Delta})(\forall a \in \mathbf{a}^{\Delta})(\exists b^* \in \mathbf{b}^{*\Delta})f(a, x^*) = b^*$$

and $\text{pro}\mathbf{b}^* \subseteq \text{pro}\mathbf{b}$, then $\mathbf{x}^{*\Delta}$ is contained in the tolerable solution set $\sum_{\forall\exists}$ of Eq. (3.1).

Proof. $\text{pro}\mathbf{b}^* \subseteq \text{pro}\mathbf{b} \Rightarrow (\forall b^* \in \mathbf{b}^{*\Delta}), b^* \in \mathbf{b}^{\Delta}$. If $\mathbf{b}^* = F(\mathbf{a}, \mathbf{x}^*)$ and it can be interpreted as $(\forall x^* \in \mathbf{x}^{*\Delta})(\forall a \in \mathbf{a}^{\Delta})(\exists b^* \in \mathbf{b}^{*\Delta})$ $f(a, x^*) = b^*$, then $(\forall x^* \in \mathbf{x}^{*\Delta})(\forall a \in \mathbf{a}^{\Delta})(\exists b \in \mathbf{b}^{\Delta})f(a, x^*) = b$ holds. Therefore, $\mathbf{x}^{*\Delta} \subseteq \sum_{\forall\exists}$. □

Example 3. From a liner system $\mathbf{ax} = \mathbf{b}$ [50]

$$\begin{pmatrix} [2,4] & [-2,1] \\ [-1,2] & [2,4] \end{pmatrix} \mathbf{x} = \begin{pmatrix} [-2,2] \\ [-2.2] \end{pmatrix} \tag{3.5}$$

with given $\mathbf{x}^* = ([-0.2, 0.2],[-0.2, 0.2])$, we receive $\mathbf{b}^* = \mathbf{ax}^* = ([-1.2,1.2], [-1.2,1.2])$ for which $\text{pro}\mathbf{b}^* \subseteq \text{pro}\mathbf{b}$ holds. As in the proof of Theorem 1, $\mathbf{x}^*$ belongs to the tolerable solution set of Eq. (3.5).

THEOREM 2. *Given* $\mathbf{x}^*$, *if* $\mathbf{b}^* = F(\mathbf{a}, \mathbf{x}^*)$ *and it can be interpreted as*

$$(\forall x^* \in \mathbf{x}^{*\Delta})(\exists a \in \mathbf{a}^{\Delta})(\exists b^* \in \mathbf{b}^{*\Delta})f(a, x^*) = b^*$$

and $\text{pro}\mathbf{b}^* \subseteq \text{pro}\,\mathbf{b}$, then $\mathbf{x}^{*\Delta}$ is contained in the united solution set $\sum_{\exists\exists}$.

Proof. $\text{pro}\mathbf{b}^* \subseteq \text{pro}\mathbf{b} \Rightarrow (\forall b^* \in \mathbf{b}^{*\Delta}), b^* \in \mathbf{b}^{\Delta}$, if $\mathbf{b}^* = F(\mathbf{a},\mathbf{x}^*)$ and it can be interpreted as $(\forall x^* \in \mathbf{x}^{*\Delta})(\exists a \in \mathbf{a}^{\Delta})$ $(\exists b^* \in \mathbf{b}^{*\Delta})f(a, x^*) = b^*$, then $(\forall x^* \in \mathbf{x}^{*\Delta})(\exists a \in \mathbf{a}^{\Delta})(\exists b \in \mathbf{b}^{\Delta})$ $f(a, x^*) = b$ holds. Therefore, $\mathbf{x}^{*\Delta} \subseteq \sum_{\exists\exists}$. □

In order to obtain a united solution in Example 3, $\mathbf{a}$ should be associated with $\exists$, which implies that $\mathbf{a}$ must be improper according to Rule R1. Thus, imp $\mathbf{a}$ (defined in Eq. (2.3)) is applied. With the same given $\mathbf{x}^*$, $\mathbf{b}^* = (\text{imp }\mathbf{a})\mathbf{x}^* = ([-0.4, 0.4], [-0.4, 0.4])$ for which $\text{pro}\mathbf{b}^* \subseteq \text{pro}\mathbf{b}$ holds. From Theorem 2, $\mathbf{x}^*$ is a united solution set of Eq. (3.5).

THEOREM 3. *Given* $\mathbf{x}^*$, *if* $\mathbf{b}^* = F(\mathbf{a}, \mathbf{x}^*)$ *and it can be interpreted as*

$$(\forall x^* \in \mathbf{x}^{*\Delta})(\forall b^* \in \mathbf{b}^{*\Delta})(\exists a \in \mathbf{a}^{\Delta})f(a, x^*) = b^*$$

and $\text{pro}\mathbf{b}^* \supseteq \text{pro}\mathbf{b}$, then $\mathbf{x}^{*\Delta}$ is contained in the controllable solution set $\sum_{\exists\forall}$.

Proof. Since $\text{pro}\mathbf{b}^* \supseteq \text{pro}\mathbf{b} \Rightarrow (\forall b \in \mathbf{b}^{\Delta}), b \in \mathbf{b}^{*\Delta}$, if $\mathbf{b}^* = F(\mathbf{a}, \mathbf{x}^*)$ and it can be interpreted as $(\forall x^* \in \mathbf{x}^{*\Delta})(\forall b \in \mathbf{b}^{*\Delta})$ $(\exists a \in \mathbf{a}^{\Delta})f(a, x^*) = b^*$, then $(\forall x^* \in \mathbf{x}^{*\Delta})(\forall b \in \mathbf{b}^{\Delta})(\exists a \in \mathbf{a}^{\Delta})$ $f(a, x^*) = b$ holds. Therefore, $\mathbf{x}^{*\Delta} \subseteq \sum_{\exists\forall}$. □

Given $\mathbf{x}^*$ as in Example 3, $\mathbf{b}^* = (\text{imp }\mathbf{a})\mathbf{x}^* = ([-0.4, 0.4], [-0.4, 0.4])$ for which $\text{pro}\mathbf{b}^* \supseteq \text{pro}\mathbf{b}$ does not hold. Therefore, $\mathbf{x}^*$ is not a controllable solution of Eq. (3.5), according to Theorem 3.

THEOREM 4. *Given the constraint in Eq.* (3.1) *and a function* G*:* $\mathbb{KR}^l \times \mathbb{KR}^m \rightarrow \mathbb{KR}^n$ *such that there is a* $\mathbf{x} = G(\mathbf{a}, \mathbf{b})$ *which can be interpreted as*

$$(\forall a \in \mathbf{a}^{\Delta})(\forall b \in \mathbf{b}^{\Delta})(\exists x \in \mathbf{y}^{\Delta})f(a, x) = b \tag{3.6}$$

for any interval box $\mathbf{x}^*$ with $\mathbf{x}^{*\Delta} \cap \mathbf{x}^{\Delta} = \phi$, $\mathbf{x}^{*\Delta}$ does not contain any solution that is in either one of the tolerable, controllable, or united solution set.

Table 1 The branch-and-prune algorithm

```
INPUT: $C(x, F, \mathbf{a}, \mathbf{b})$, $\mathbf{x}^0 \in \mathbb{KR}^n$, $\varepsilon$, 'Semantics'; OUTPUT: $S$, $B$;

1     Initialization: $\mathcal{M} = \mathbf{x}^0$; $S = \Phi$;
2     WHILE $\mathcal{M}$ is not empty DO
3         $\mathbf{x} \leftarrow$ extract($\mathcal{M}$);
4        IF Solution Identification($C$, $\mathbf{x}$, 'Semantics') = True
5           $S = S \cup \mathbf{x}$;
6        ELSEIF Pruning Test($C$, $\mathbf{x}$) = True
7           $\mathbf{x}$ is pruned;
8        OTHERWISE
9        IF wid($\mathbf{x}$) $\geq \varepsilon$
10          Branch $\mathbf{x}$ to $\mathbf{x}1$ and $\mathbf{x}2$;
11          Store $\mathbf{x}1$ and $\mathbf{x}2$ in $\mathcal{M}$;
12    ELSE
13          Store $\mathbf{x}$ in $B$;
14    END WHILE;
15    RETURN $S$ and $B$.
```

Proof. For any $\mathbf{x}$ that can be interpreted as Eq. (3.6), $\mathbf{x}^{\Delta}$ is a complete solution to the constraints of F. So for a solution $\mathbf{y}$ that can be interpreted as either $(\forall x \in \mathbf{y}^{\Delta})(\forall a \in \mathbf{a}^{\Delta})(\exists b \in \mathbf{b}^{\Delta})$ $f(a, x) = b$, $(\forall x \in \mathbf{y}^{\Delta})(\forall b \in \mathbf{b}^{\Delta})(\exists a \in \mathbf{a}^{\Delta})f(a, x) = b$, or $(\forall x \in \mathbf{y}^{\Delta})(\exists a \in \mathbf{a}^{\Delta})(\exists b \in \mathbf{b}^{\Delta})f(a, x) = b$, $\text{pro}\mathbf{y} \subseteq \text{pro}\mathbf{x}$ should be true. Therefore, if $\mathbf{x}^{*\Delta} \cap \mathbf{x}^{\Delta} = \phi$, then $\mathbf{x}^{*\Delta}$ does not contain any solution that is included in one of the three solution sets. □

Remark. Generally, $\mathbf{x}$ in Theorem 4 can be calculated by deriving individual constraints of G associated with each variable of $\mathbf{x} = (\mathbf{x}_1,\ldots,\mathbf{x}_n)$ in a fixed-point fashion from the original constraints of F. For example, there may exist three individual constraints G_j's for $j = 1, 2, 3$, where $\mathbf{x}_i$ can be derived from the constraints in F. Then, $\mathbf{x}_i^j = G_j(\mathbf{a}, \mathbf{b}; \mathbf{x}_1, ..., \mathbf{x}_{i-1}, \mathbf{x}_{i+1}, ..., \mathbf{x}_n)$, where $\mathbf{x}_i^j$ is the ith element in $\mathbf{x} \in \mathbb{KR}^n$, and solved by the jth constraint that involves $\mathbf{x}_i$. If any $\mathbf{x}_i^{j\Delta} \cap \mathbf{x}_i^{\Delta} = \phi$, then $\mathbf{x}^{\Delta}$ does not contain any solution that is included in one of the three solution sets. For the first equation in Eq. (3.5), $\mathbf{x}_1 = ([-2,2] - [-2,1]\mathbf{x}_2)/[2,4] = [-1.2,1.2]$ for which $[-1.2,1.2]^{\Delta} \cap [-0.2,0.2]^{\Delta} \neq \phi$ holds. So does for the second equation of Eq. (3.5). Therefore, we can conclude that $\mathbf{x}_1^*$ contains the solution of the equations based on Theorem 4. $\mathbf{x}_1^*$ indeed is the solution.

In the proposed branch-and-prune algorithm listed in Table 1, Theorems 1–3 are used during the solution identification test for the three types of solution sets. Theorem 4 is applied in the pruning test. In each step, consistency test searches and identifies an inner estimation of the feasible solution set. Iteratively, the collection of inner estimations is accumulated, and the union of them is the final approximation of the solution. Note that the cost of the proposed branch-and-prune algorithm depends on the number of design variables n, the smallest variable domain size d, and the required precision ε. The number of iterations is bounded by $\mathcal{O}((d/\varepsilon)^n)$.

The boundary boxes enclose the solution set. The branching step subdivides the boundary boxes to smaller ones until certain level of precision is reached. Bisection along one dimension of $\mathbf{x}$ is usually applied in the branching step such that two sub-boxes $\mathbf{x}_1$ and $\mathbf{x}_2$ with the same volume are generated. The two sub-boxes do not overlap with each other. That is, $\mathbf{x} = \mathbf{x}_1^{\Delta} \cup \mathbf{x}_2^{\Delta}$. So the solution S is a collection of nonoverlapped sub-boxes that are branched from $\mathbf{x}^0$.

4 Hybrid Stratified Monte Carlo Method

A hybrid stratified Monte Carlo sampling method is proposed to verify the solutions obtained from the branch-and-prune algorithm in Sec. 3. It can be used both to verify if the computed feasible solution subspace is indeed an inner estimation of the true solution set, and to verify if the subspace that has been pruned does not include any solution.

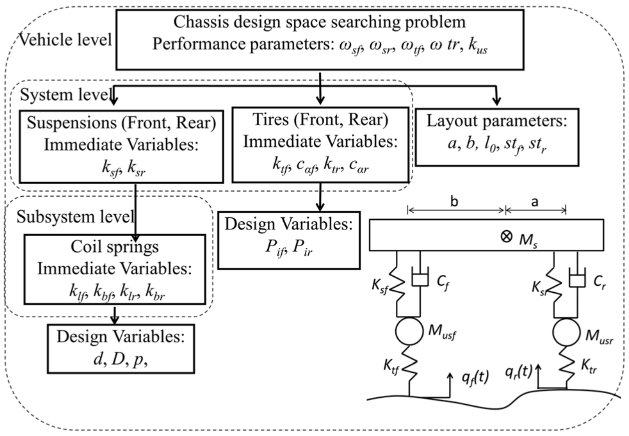

Vehicle level
Chassis design space searching problem
Performance parameters: ω_{sf}, ω_{sr}, ω_{tf}, ω tr, k_{us}
System level
Suspensions (Front, Rear) Immediate Variables: k_{sf}, k_{sr}
Tires (Front, Rear) Immediate Variables: k_{tf}, $c_{\alpha f}$, k_{tr}, $c_{\alpha r}$
Layout parameters: a, b, l_0, st_f, st_r
Subsystem level
Coil springs Immediate Variables: k_{lf}, k_{bf}, k_{lr}, k_{br}
Design Variables: P_{if}, P_{ir}
Design Variables: d, D, p,
b
a
M_s
K_{sf}
C_f
K_{sr}
C_r
M_{usf}
M_{usr}
K_{tf}
K_{tr}
$q_f(t)$
$q_r(t)$

Fig. 1 Variables in QCSP for chassis design

The conventional Monte Carlo method can be used to evaluate the solution set of Eq. (3.1) by uniformly sampling within the domains of parameters **a, b**, and variables **x**. Compared to the conventional Monte Carlo method where all parameters and variables are sampled from their respective interval domains, the proposed hybrid stratified Monte Carlo method is of much lower computational load because it only samples variables within their domains and leaves all parameters as intervals. To apply this method to verify the feasible or infeasible subspace, values of $x \in \mathbb{R}^n$ within the computed subspace $\mathbf{x} \in \mathbb{KR}^n$ are sampled based on uniform distributions. For each of sampled value, the consistency test is applied to check if the value indeed belongs to the solution.

Because the solution subspace computed in Sec. 3 is a set of boxes, we deterministically choose every box and sample uniformly within each box. The number of samples drawn for each box is proportional to the volume of the box. The hybrid stratified Monte Carlo approach improves the sampling efficiency while ensuring that each box is verified. Additionally, proper and improper interval values can be assigned to parameters so that all three types of solution sets for the problem in Eq. (3.1) can be verified, whereas the conventional Monte Carlo method can only be applied to verify tolerable solution sets. Moreover, for the case of tolerable solution sets, a large number of real-valued samples for parameters **a** are needed before the conclusion about $(\forall a \in \mathbf{a}^\Delta)$ is drawn in a statistical sense instead of the guaranteed assessment for all values of a. In contrast, the hybrid stratified Monte Carlo method samples **x** and keeps parameters **a** as intervals so that calculation based on intervals can guarantee all values of a satisfy the constraints. The rigorous verification can be performed efficiently by interval arithmetic. The consistency tests for detecting a feasible or infeasible subspace for each of the three solution sets are described in details as follows.

We define $\mathbf{b}_{\text{in}}^\Delta$ and $\mathbf{b}_{\text{out}}^\Delta$ as the inner and outer estimations of the exact range of b for $b = f(a, x)$, respectively, where a and x can take any value within their given boxes **a** and **x**. The exact range of $f(a, x)$ over interval boxes **a** and **x** is denoted as $R(F(\mathbf{a},\mathbf{x}))$. $R(F(\mathbf{a},\mathbf{x}))$ is both the sound and complete solution for $b = f(a, x)$, where $a \in \mathbf{a}^\Delta$ and $x \in \mathbf{x}^\Delta$. Thus, we have $\mathbf{b}_{\text{in}}^\Delta \subseteq R(F(\mathbf{a},\mathbf{x})) \subseteq \mathbf{b}_{\text{out}}^\Delta$. Because of the overestimation involved in interval arithmetic, $\mathbf{b}_{\text{out}}^\triangle$ can be obtained using natural inclusion function and interval arithmetic with $f(a,x)$. The inner estimation of $F(\mathbf{a},\mathbf{x})$, $\mathbf{b}_{\text{in}}^\Delta$, can be computed from $\mathbf{b}_{\text{in}} = F(\text{imp}\mathbf{a}, \text{pro}\mathbf{x})$ provided that the resulted $\mathbf{b}_{\text{in}}$ is improper, which is always the case when **x** is pointwise. At each verification, a real value x_0 is sampled from the boxes within the solution set, then $\mathbf{b}_{\text{in}}^\Delta$ and $\mathbf{b}_{\text{out}}^\Delta$ are calculated over **a** and pointwise interval x_0.

For the tolerable solution set where the existential quantifier is associated with **b,** and the universal ones are associated with **a** and **x**, if **x** is a solution box, $R(F(\mathbf{a}, \mathbf{x}))$ must be a subset of $\mathbf{b}^\Delta$. Therefore, $x_0 \in \sum_{\forall\exists} \Leftrightarrow R(F(\mathbf{a},x_0)) \subseteq \mathbf{b}^\Delta$. Since obtaining the exact range is not possible in many cases, the outer and inner estimations are used in the consistency test. The consistency test for the tolerable solution set is as follows. If $\mathbf{b}_{\text{out}}^\Delta \subseteq \mathbf{b}^\Delta$, then $x_0 \in \sum_{\forall\exists}$; if $\mathbf{b}_{\text{in}}^\Delta \not\subseteq \mathbf{b}$, then $x_0 \notin \sum_{\forall\exists}$; if $\mathbf{b}_{\text{out}}^\Delta \not\subseteq \mathbf{b}^\Delta$ and $\mathbf{b}_{\text{in}}^\Delta \subseteq \mathbf{b}$, then no decision of feasible or infeasible solution can be made. If the inner estimation is computationally more expensive than the outer estimation, then the alternative consistency test is as follows. If $\mathbf{b}_{\text{out}}^\Delta \subseteq \mathbf{b}^\Delta$, then $x_0 \in \sum_{\forall\exists}$; if $\mathbf{b}_{\text{out}}^\Delta \cap \mathbf{b}^\Delta = \Phi$, then $x_0 \notin \sum_{\forall\exists}$; if $\mathbf{b}_{\text{out}}^\Delta \not\subseteq \mathbf{b}^\Delta$ and $\mathbf{b}_{\text{out}}^\Delta \cap \mathbf{b}^\Delta \neq \Phi$, then no decision can be made.

For the united solution set, if **x** belongs to the solution, $R(F(\mathbf{a}, \mathbf{x}))$ must have at least one value in common with **b.** Therefore, $x_0 \in \sum_{\exists\exists} \Leftrightarrow R(F(\mathbf{a},x_0)) \cap \mathbf{b}^\Delta \neq \Phi$. The consistency test for the united solution set is as follows. If $\mathbf{b}_{\text{in}}^\Delta \cap \mathbf{b}^\Delta \neq \Phi$, then $x_0 \in \sum_{\exists\exists}$; if $\mathbf{b}_{\text{out}}^\Delta \cap \mathbf{b}^\Delta = \Phi$, then $x_0 \notin \sum_{\exists\exists}$; if $\mathbf{b}_{\text{in}}^\Delta \cap \mathbf{b}^\Delta = \Phi$ and $\mathbf{b}_{\text{out}}^\Delta \cap \mathbf{b}^\Delta \neq \Phi$, then no decision can be made.

For the controllable solution set, if **x** belongs to the solution, $R(F(\mathbf{a}, \mathbf{x}))$ must be a superset of **b**. Therefore, $x_0 \in \sum_{\exists\forall} \Leftrightarrow R(F(\mathbf{a},x_0)) \supseteq \mathbf{b}^\Delta$. The consistency test for the controllable solution set is as follows. If $\mathbf{b}_{\text{in}}^\Delta \supseteq \mathbf{b}^\Delta$, then $x_0 \in \sum_{\exists\forall}$; if $\mathbf{b}_{\text{out}}^\Delta \subseteq \mathbf{b}^\Delta$ then $x_0 \notin \sum_{\exists\forall}$; if $\mathbf{b}_{\text{in}}^\Delta \not\subseteq \mathbf{b}^\Delta$ and $\mathbf{b}_{\text{out}}^\Delta \supseteq \mathbf{b}^\Delta$, then no decision can be made.

The proposed stratified Monte Carlo method can be applied to the general case of $f(a, x) = b$ to verify the solution sets with no limitation over the complexity of function f. In the following section, a numerical example is presented to demonstrate the proposed formulation and numerical algorithms.

5 Numerical Example

The proposed approach is applied to a vehicle chassis design problem. This example is adopted from Kim et al. [60] where the original one is formulated as a three-level multidisciplinary optimization problem. Here, the problem is modified as searching feasible design space in the form of QCSP without considering optimum objective functions. Searching feasible design space is to find all possible solutions from the initial design space with all constraints satisfied. The three types of solution sets are calculated by assuming three different design scenarios for the example.

The structure of the problem with all variables is shown in Fig. 1. The variables are divided into four groups, which are design variables, layout parameters, performance parameters, and immediate variables for gravity. The design space is consisted with all the possible values of design variables, which satisfy all of the constraints.

There are five constraints at the vehicle-level, such as

$$\omega_{sf} = \frac{1}{2\pi}\sqrt{\frac{K_{sf}}{M_{sf}}}(a), \omega_{sr} = \frac{1}{2\pi}\sqrt{\frac{K_{sr}}{M_{sr}}}(b), \omega_{tf} = \frac{1}{2\pi}\sqrt{\frac{K_{tf}}{M_{usf}}}(c),$$
$$\omega_{tr} = \frac{1}{2\pi}\sqrt{\frac{K_{tr}}{M_{usr}}}(d), k_{us} = \frac{Mb}{LC_{\alpha f}} - \frac{Ma}{LC_{\alpha r}}(e) \tag{5.1}$$

where ω_{sf} and ω_{sr} are first natural frequencies of front and rear suspensions, ω_{tf} and ω_{tr} are second natural frequencies of front and rear suspensions, k_{us} is understeer gradient, a and b are the distances of gravity center to the axles, $M_{sf} = M_s b/2(a+b)$, $M_{sr} = M_s a/2(a+b)$, $M_{usf(r)} = M_{us}/4$, $L = a + b$, and $M = M_s + M_{us}$ in which sprung mass $M_s = 2282$ kg, unsprung mass $M_{us} = 228$ kg.

Performance requirements, ω_{sf}, ω_{sr}, ω_{tf}, ω_{tr}, k_{us}, are defined as a vector B. a and b are layout parameters. $K = (K_{sf}, K_{sr}, K_{tf}, K_{tr}, C_{\alpha f}, C_{\alpha r})$ is the vector of immediate variables in which K_{sf}, K_{sr}, K_{tf}, K_{tr} are the stiffness of suspensions (with subscript s) and tires (with subscript t). $C_{\alpha f}$ and $C_{\alpha r}$ are damping coefficients of front (with subscript f) and rear (with subscript r) tire systems.

The suspension systems at the system level are formulated by regression models

$$K_{sf} = 17 + 0.067K_{lf} - 0.0034L_{of} + 0.000278K_{Bf} + 35.6st_f \tag{5.2}$$

and

$$K_{sr} = 17 + 0.067K_{lr} - 0.0034L_{or} + 0.000278K_{Br} + 35.6st_r \tag{5.3}$$

respectively, where the suspension stiffness K_s is inflected by linear spring stiffness K_l, spring bending stiffness K_B, spring free length L_o, and spring travel distance st.

The design constraints for the vertical and cornering tire stiffness [61] are described as

$$K_{tf} = 0.9((0.1839P_{if} - 9.2605)F_m + 110119) \tag{5.4}$$

$$K_{tr} = 0.9((0.1839P_{ir} - 9.2605)F_m + 110119) \tag{5.5}$$

$$C_{\alpha f} = 180F_m(-2.668 \times 10^{-6}P_{if}^2 + 1.605 \times 10^{-3}P_{if} - 3.86 \times 10^{-2})/\pi \tag{5.6}$$

$$C_{\alpha r} = 180F_m(-2.668 \times 10^{-6}P_{ir}^2 + 1.605 \times 10^{-3}P_{ir} - 3.86 \times 10^{-2})/\pi \tag{5.7}$$

where $F_m = 9.81Mb/(a+b)$, P_{if}, and P_{ir} are tire inflation pressures.

At the subsystem level, coil spring is with the design variables of wire diameter d, coil diameter D, and the pith p. Free length L_o is a layout parameter. The front and rear coil springs are assumed to be the same type. The detailed constraints are given as

$$K_{lf} = \frac{Gd^4}{8D^3(L_o - 3d)/p}, K_{Bf} = \frac{EGd^4}{16D(2G+E)} \bullet \frac{\pi}{180} \tag{5.8}$$

where rigidity modulus G of spring material is usually 8×10^4 N/mm^2 for the steel without preload, and Young's modulus E is usually 2.01×10^5 N/mm^2 for steel.

In summary, the QCSP has design variables $x = (d, D, p, P_{if}, P_{ir})$ in the respective given interval domains $\mathbf{x} = (\mathbf{d}, \mathbf{D}, \mathbf{p}, \mathbf{P}_{if}, \mathbf{P}_{ir})$, layout parameters $A = (a, b, L_o, st_f, st_r)$ in the respective given domains $\mathbf{A} = (\mathbf{a}, \mathbf{b}, \mathbf{L}_o, \mathbf{st}_f, \mathbf{st}_r)$, and performance parameters $B = (\omega_{sf}, \omega_{sr}, \omega_{tf}, \omega_{tr}, k_{us})$ in the respective given domains $\mathbf{B} = (\boldsymbol{\omega}_{sf}, \boldsymbol{\omega}_{sr}, \boldsymbol{\omega}_{tf}, \boldsymbol{\omega}_{tr}, \mathbf{k}_{us})$.

5.1 Quantified Constraints With Logic Interpretation.

In the case of set-based design at the early design stage, the feasible solution set for the chassis design problem can be described as

$$\sum(\mathbf{A}, \mathbf{B}, x) = \{x \in \mathbf{X} | (\exists A \in \mathbf{A}^\Delta)(\exists B \in \mathbf{B}^\Delta) f(A, x) = B\} \tag{5.9}$$

which means that we aim to find all feasible values for each variable such that there exists at least one combination of layout and performance parameters that satisfies the performance requirements. It implies that the performance requirements can be realized by current design and manufacturing conditions. Eq. (5.9) is in accordance with the definition of united solution set. Therefore, based on the interpretation rules R1–R2 in Sec. 3, interval boxes for layout parameters **A** should be improper, performance parameters **B** should be proper, and design variables **X** should be proper. The logic interpretation for united solution set is given in Eqs. (5.10)–(5.19).

Layout parameters a and b are multioccurrences variables, to which the dual operation should be applied based on Rule R6. Thus, dual is applied to all occurrences of a and b in the first four sub-equations of Eq. (5.1). Immediate variables K_{sf}, K_{sr}, K_{tf}, K_{tr}, $C_{\alpha f}$, $C_{\alpha r}$ are assigned to be proper in Eq. (5.1) because of Kaucher arithmetic. For example,ω_{tr} is required to be proper interval, M_{usf} is a real number in Eq.(5.1*d*). The numerical result of the calculation between a proper interval and a real number is proper. Thus, K_{tr} should be proper too. Then, based on Rules R1, R3, and R5, the logic interpretations for the sub-equations in Eq. (5.1) are

$$(\forall K_{sf} \in \mathbf{K}_{sf}^\Delta)(\forall a \in \mathbf{a}^\Delta)(\forall b \in \mathbf{b}^\Delta)(\exists \omega_{sf} \in \omega_{sf}^\Delta)(\text{Eq.}(5.1a) \text{ is satisfied}) \tag{5.10}$$

$$(\forall K_{sr} \in \mathbf{K}_{sr}^\Delta)(\forall a \in \mathbf{a}^\Delta)(\forall b \in \mathbf{b}^\Delta)(\exists \omega_{sr} \in \omega_{sr}^\Delta)(\text{Eq.}(5.1b) \text{ is satisfied}) \tag{5.11}$$

$$(\forall K_{tf} \in \mathbf{K}_{tf}^\Delta)(\forall m_{usf} \in \mathbf{M}_{usf}^\Delta)(\exists \omega_{tf} \in \omega_{tf}^\Delta)(\text{Eq.}(5.1c) \text{ is satisfied}) \tag{5.12}$$

$$(\forall K_{tr} \in \mathbf{K}_{tr}^\Delta)(\exists m_{usr} \in \mathbf{M}_{usr}^\Delta)(\exists \omega_{tr} \in \omega_{tr}^\Delta)(\text{Eq.}(5.1d) \text{ is satisfied}) \tag{5.13}$$

$$(\forall C_{\alpha f} \in \mathbf{C}_{\alpha f}^\Delta)(\forall C_{\alpha r} \in \mathbf{C}_{\alpha r}^\Delta)(\exists a \in \mathbf{a}^\Delta)(\exists b \in \mathbf{b}^\Delta)(\exists k_{us} \in \mathbf{k}_{us}^\Delta)(\text{Eq.}(5.1e) \text{ is satisfied}) \tag{5.14}$$

A real number defined as a pointwise interval can be interpreted as either universal or existential as needed, such as M_{usf} and M_{usr} in Eqs. (5.12) and (5.13). For most of the time, there is no need to interpret real numbers, because M_{usf} and M_{usr} can be either proper or improper based on Rule R3, which also have no effect on the modality of final interval result. Therefore, the interpretation for real numbers are omitted in the remainder of this section. The rear suspension/tire systems are very similar to the front ones.

Layout parameters L_o and st_f, which are elements of **A**, are assigned to be improper in this scenario. In order to preserve K_{sf}, which is shared by Eqs. (5.2) and Eq. (5.1*a*), and proper in Eq. (5.1*a*), K_{lf} and K_{Bf} should be proper too based on Kaucher arithmetic. Therefore, Eq. (5.2) is interpreted as

$$(\forall K_{lf} \in \mathbf{K}_{lf}^\Delta)(\forall K_{Bf} \in \mathbf{K}_{Bf}^\Delta)(\exists L_o \in \mathbf{L}_o^\Delta)(\exists st_f \in \mathbf{st}_f^\Delta)(\exists K_{sf} \in \mathbf{K}_{sf}^\Delta)(\text{Eq.}(5.2) \text{ is satisfied}) \tag{5.15}$$

Equations (5.4) and (5.6) involve multioccurrence parameters a and b. Their interpretations have been preserved as existential as in Eq. (5.1). Dual operation is applied here so that Eqs. (5.4) and (5.6) can be combined with the interpretations above based on Rules R5–R6. Therefore, Eqs. (5.4) and (5.6) are interpreted as

$$(\forall P_{if} \in \mathbf{P}_{if}^\Delta)(\forall a \in \mathbf{a}^\Delta)(\forall b \in \mathbf{b}^\Delta)(\exists K_{tf} \in \mathbf{K}_{tf}^\Delta)(\text{Eq.}(5.4) \text{ is satisfied}) \tag{5.16}$$

and

$$(\forall P_{if} \in \mathbf{P}_{if}^\Delta)(\forall a \in \mathbf{a}^\Delta)(\forall b \in \mathbf{b}^\Delta)(\exists C_{\alpha f} \in \mathbf{C}_{\alpha f}^\Delta)(\text{Eq.}(5.6) \text{ is satisfied}) \tag{5.17}$$

respectively. Because coil spring is a subsystem of suspension, the constraints in Eq. (5.2) for the suspension stiffness and Eq. (5.8) for the coil spring are coupled. Therefore, when Eq. (5.8) is interpreted, the interpretation in Eq. (5.15) for Eq. (5.2) and the interpretation in Eq. (5.10) for Eq. (5.1*a*) at the vehicle level should be taken into consideration. Design variables d, D, and p are assigned to be proper in Eq. (5.9) in this scenario. L_o is improper in Eq. (5.15), which should be applied by a dual operation based on Rule R6. K_{lf} and K_{Bf}, which are proper in Eq. (5.15), should still be proper too. The values of K_{lf} and K_{Bf} used in Eq. (5.2) are coming from Eq. (5.8), for which the interpretations should be self-consistent. Based on Rule R1, Eq. (5.8) is interpreted as

Table 2 The given values

Performance parameters		Immediate variables	
ω_{sf} (Hz)	[1.1,1.5]	$\mathbf{K}_{sf}$ (N/mm)	[13.13,56.25]
ω_{sr} (Hz)	[1.1,1.5]	$\mathbf{K}_{sr}$ (N/mm)	[25.7, 60]
ω_{tf} (Hz)	[11,14]	$\mathbf{K}_{tf}$ (N/mm)	[142, 494]
ω_{tr} (Hz)	[11,14]	$\mathbf{K}_{tr}$ (N/mm)	[126, 355]
$\mathbf{k}_{us}$ (rad/m/s^2)	[0.0015, 0.0055]	$\mathbf{K}_{lf(r)}$ (N/mm)	[120,180]
		$\mathbf{K}_{Bf(r)}$ (Nmm/deg)	[75,000,85,000]
		$\mathbf{C}_{\alpha f}$ (N/rad/10^{-4})	[7.08,19.6]
		$\mathbf{C}_{\alpha r}$ (N/rad/10^{-4})	[4.28,12.32]
Layout parameters		Design variables	
a (m)	[1.25,1.39]	$\mathbf{P}_{ir}$ (KPa)	[83,330]
b (m)	[2.31,2.45]	$\mathbf{P}_{if}$ (KPa)	[83,330]
$\mathbf{L}_{0f(r)}$ (mm)	[350,420]	**d** (mm)	[5,30]
$\mathbf{st}_{f(r)}$ (m)	[0.05,0.1]	**D**(mm)	[50,200]
		p (mm)	[50,100]

$$(\forall d \in \mathbf{d}^{\Delta})(\forall D \in \mathbf{D}^{\Delta})(\forall p \in \mathbf{p}^{\Delta})(\forall L_o \in \mathbf{L}_o^{\Delta})(\exists K_{lf} \in \mathbf{K}_{lf}^{\Delta})$$
$$(\exists K_{Bf} \in \mathbf{K}_{Bf}^{\Delta})(\text{Eq.(5.8) is satisfied}) \quad (5.18)$$

Based on Rules R4–R5, the final logic interpretation of the united solution set for this chassis design problem is obtained by combining Eqs. (5.10)–(5.18) as

$$(\forall d \in \mathbf{d}^{\Delta})(\forall D \in \mathbf{D}^{\Delta})(\forall p \in \mathbf{p}^{\Delta})(\forall P_{if} \in \mathbf{P}_{if}^{\Delta})(\forall P_{ir} \in \mathbf{P}_{ir}^{\Delta})(\exists a \in \mathbf{a}^{\Delta})$$
$$(\exists b \in \mathbf{b}^{\Delta})(\exists L_o \in \mathbf{L}_o^{\Delta})(\exists st_f \in \mathbf{st}_f^{\Delta})(\exists st_r \in \mathbf{st}_r^{\Delta})(\exists \omega_{sf} \in \omega_{sf}^{\Delta})$$
$$(\exists \omega_{sr} \in \omega_{sr}^{\Delta})(\exists \omega_{tf} \in \omega_{tf}^{\Delta})(\exists \omega_{tr} \in \omega_{tr}^{\Delta})(\exists k_{us} \in \mathbf{k}_{us}^{\Delta})$$
$$(\text{Eqs.}(5.1)-(5.8) \text{ are satisfied}) \quad (5.19)$$

In addition to the united solution set, different kinds of solutions can be used based on the desired design intent. For instance, in the scenario of product family design as discussed in Sec. 1, products can be generated by combining different modules with one platform. In a vehicle family, the suspension and tire systems can be taken as modules to generate the design variations, whose parameters are scaled ones. They usually come with the chassis. A chassis as an assembly of the platform should not only have the ability to accommodate different suspension and tire modules in the vehicle family but also satisfy the performance requirements of the vehicle family. The accommodating ability of a chassis should not be over designed by considering the design and manufacturing cost. It means that all the layout parameters on the chassis should be used in the family. In that case, the feasible solution set for the chassis design problem can be described as

$$\sum(\mathbf{A},\mathbf{B},x) = \{x \in \mathbf{X} | (\forall A \in \mathbf{A}^{\Delta})(\exists B \in \mathbf{B}^{\Delta}) f(A,x) = B\} \quad (5.20)$$

In other words, we aim to find all feasible values for each design variable x such that for all the values of layout parameters a, b, L_0, st_f, and st_r, there exist acceptable values for performance parameters in the given ranges that satisfy the performances requirements. These layout parameters vary slightly within the given domains for different products. The goal is to accommodate the differences in the common platform of chassis. Therefore, the interval boxes **A**, **B**, and **X** are assigned to be proper. Similar to the previous interpretation in the scenario of untied solution set, the logic interpretation of the tolerable solution set for the chassis design problem based on Rules R1, R2, R4, and R5 is

$$(\forall d \in \mathbf{d}^{\Delta})(\forall D \in \mathbf{D}^{\Delta})(\forall p \in \mathbf{p}^{\Delta})(\forall P_{if} \in \mathbf{P}_{if}^{\Delta})(\forall P_{ir} \in \mathbf{P}_{ir}^{\Delta})$$
$$(\forall a \in \mathbf{a}^{\Delta})(\forall b \in \mathbf{b}^{\Delta})(\forall L_o \in \mathbf{L}_o^{\Delta})(\forall st_f \in \mathbf{st}_f^{\Delta})$$
$$(\forall st_r \in \mathbf{st}_r^{\Delta})(\exists \omega_{sf} \in \omega_{sf}^{\Delta})(\exists \omega_{sr} \in \omega_{sr}^{\Delta}) \quad (5.21)$$
$$(\exists \omega_{tf} \in \omega_{tf}^{\Delta})(\exists \omega_{tr} \in \omega_{tr}^{\Delta})(\exists k_{us} \in \mathbf{k}_{us}^{\Delta})$$
$$(\text{Eqs.}(5.1)-(5.8) \text{ are satisfied})$$

In a different scenario, engineers have specific performance targets to satisfy customer requirements. They may find that the new product family can be developed from an existing platform. It indicates that all the targets can be satisfied by selecting proper values for layout parameters of the existing platform and integrating new modules. The ability to accommodate a current platform can support the development of the new production family. In that case, the feasible solution set can be described as

$$\sum(\mathbf{A},\mathbf{B},x) = \{x \in \mathbf{X} | (\forall B \in \mathbf{B}^{\Delta})(\exists A \in \mathbf{A}^{\Delta}) f(A,x) = B\} \quad (5.22)$$

which is a controllable solution set. We need to find all feasible values of design variables such that all values in the range of performance requirements can be achieved by the layout parameters. The interval boxes **A** and **B** are assigned to be improper and **X** to be proper. The desired logic interpretation is

$$(\forall d \in \mathbf{d}^{\Delta})(\forall D \in \mathbf{D}^{\Delta})(\forall p \in \mathbf{p}^{\Delta})(\forall P_{if} \in \mathbf{P}_{if}^{\Delta})(\forall P_{ir} \in \mathbf{P}_{ir}^{\Delta})$$
$$(\forall \omega_{sf} \in \omega_{sf}^{\Delta})(\forall \omega_{sr} \in \omega_{sr}^{\Delta})(\forall \omega_{tf} \in \omega_{tf}^{\Delta})$$
$$(\forall \omega_{tr} \in \omega_{tr}^{\Delta})(\forall k_{us} \in \mathbf{k}_{us}^{\Delta})(\exists a \in \mathbf{a}^{\Delta})(\exists b \in \mathbf{b}^{\Delta})$$
$$(\exists L_o \in \mathbf{L}_o^{\Delta})(\exists st_f \in \mathbf{st}_f^{\Delta})(\exists st_r \in \mathbf{st}_r^{\Delta})$$
$$(\text{Eqs.}(5.1)-(5.8) \text{ satisfied}) \quad (5.23)$$

During the QCSP formulation for a design problem, the time consumption depends on the complexity of the problem, i.e., the numbers of variables, constraints, and the type of QCSP. It takes more time when a large number of existential variables are in the constraints, because Rule R6 in Sec. 3.1 needs to be applied.

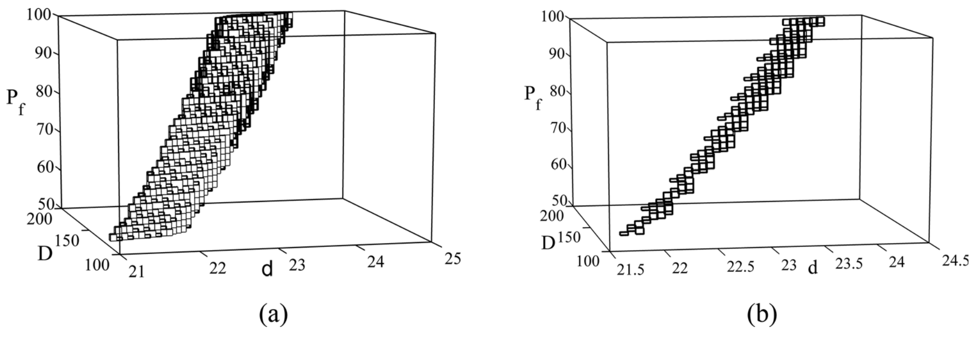
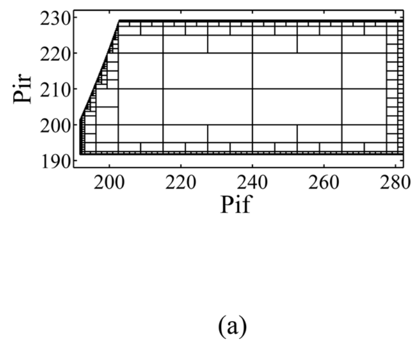
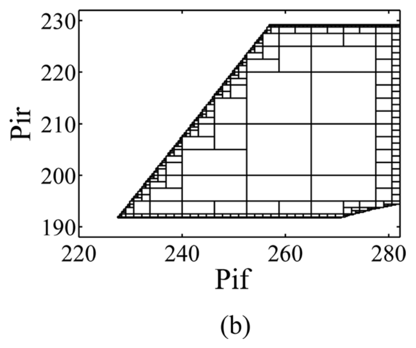


Fig. 2 Design space (*d*, *D*, *p*) (mm) for suspension system, (*a*) united solution set and (*b*) tolerable solution set





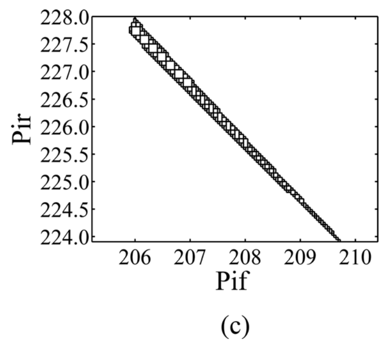


Fig. 3 Design space (P_{if}, P_{ir}) (Kpa) for tire system (*a*) united solution set, (*b*) tolerable solution set, and (*c*) controllable solution set

Table 3 The values of variables for controllable solution set

Performance parameters		Immediate variables	
ω_{tf} (Hz)	[11.74,11.75]	K_{tf} (N/mm)	[143, 493.8]
ω_{tr} (Hz)	[12,12.01]	K_{tr} (N/mm)	[125.6, 350]
$\mathbf{k}_{us}$ (rad/m/s^2)	[0.0046, 0.0047]	$\mathbf{C}_{\alpha f}$ (N/rad/10^{-4})	[13.08,19.6]
		$\mathbf{C}_{\alpha r}$ (N/rad/10^{-4})	[12.28,18.6]
Layout parameters		Design variables	
a (m)	[1.25,1.39]	$\mathbf{P}_{ir}$ (KPa)	[80,330]
b (m)	[2.31,2.45]	$\mathbf{P}_{if}$ (KPa)	[80,330]

Therefore, the tolerable solution set takes the least time to compute, whereas the controllable and united ones take more.

5.2 Results and Verifications. During the calculation, the design variables are separated into two groups. d, D, and p are in the suspension systems, whereas P_{if} and P_{ir} are in the tire system. Thus, the chassis design problem is divided in two sub-problems. Some of the layout parameters are shared between these two systems. To ensure that the combined interpretation is available, the shared parameters that are existentially quantified and occur multiple times should be modified by the dual operation based on Rule R6.

With the initial values of variables in Table 2, the inner estimation of feasible solution set for the chassis design problem based on our branch-and-prune algorithm is shown in Figs. 2 and 3. Figure 2 shows the ranges of variables d, D, and p for the suspension system for the united solution set and tolerable solution set in a 3D view, with the precision of 0.05. The design range for wire diameter d has shrunk from $[\![5,30]\!]$ to $[\![21.9,24.22]\!]$ and $[\![21.72,24.06]\!]$, respectively. Coil diameter D was reduced from $[\![5,200]\!]$ to $[\![126.6,200]\!]$ and $[\![131.8,193.8]\!]$, respectively. The spring pitch p does not change and has the value of $[\![50,100]\!]$. In Kim et al. [60], d, D, and p are formulated as design variables, and their optimum values are 23 mm, 180 mm, and 80 mm, respectively, all of which are enclosed by the feasible solutions from our calculation.

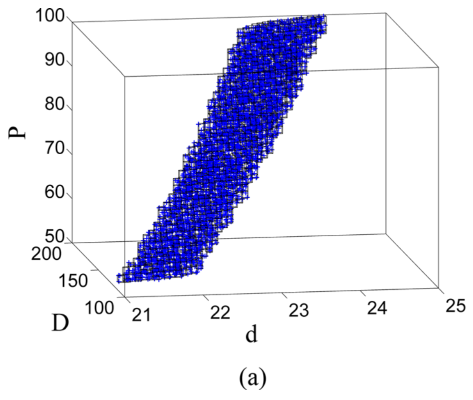
Figures 3(*a*) and 3(*b*) show the ranges of variables P_{if} and P_{ir} for the tire system for the united and tolerable solution sets, respectively, with the precision of 0.05. The range of P_{if} is reduced from $[\![80,330]\!]$ to $[\![191.8,282.2]\!]$ and the range of P_{ir} shrinks to $[\![191.6,229.2]\!]$ for the united solution set. The ranges of P_{if} and P_{ir} are reduced to $[\![227.5,282.2]\!]$ and $[\![191.7,229.2]\!]$ for the tolerable one. In Kim et al. [60], P_{if} and P_{ir} are linked variables, and their optimum values were not given. Again, different assignments of quantifiers to variables result in different solution spaces.

No controllable solution set can be found for this chassis design problem with the given values in Table 3. We pick the tire system design as a sub-problem to verify the algorithm for the controllable solution set. The constraints for this sub-problem include Eqs. 5.1(*c–e*) and Eqs. (5.4)–(5.7) with the final logic interpretation

$$(\forall P_{if} \in \mathbf{P}_{if}^{\Delta})(\forall P_{ir} \in \mathbf{P}_{ir}^{\Delta})(\forall \omega_{tf} \in \omega_{tf}^{\Delta})(\forall \omega_{tr} \in \omega_{tr}^{\Delta})(\forall k_{us} \in \mathbf{k}_{us}^{\Delta})$$
$$(\exists a \in \mathbf{a}^{\Delta})(\exists b \in \mathbf{b}^{\Delta})$$
$$(\text{Eqs.}5.1(c-e), \text{Eqs.}(5.4)-(5.7) \text{ are satisfied})$$

The domains of performance parameters listed in Table 3 are used instead of Table 2. Figure 3(*c*) shows the feasible design space of the controllable solution set for the tire inflation pressures P_{if} and P_{ir} with the precision of 0.001. The ranges of P_{if} and P_{ir} are reduced to $[\![205.9,209.7]\!]$ and $[\![223.9,227.9]\!]$.

The proposed hybrid stratified Monte Carlo method is used to verify the solutions obtained in Figs. 2 and 3. A total number of



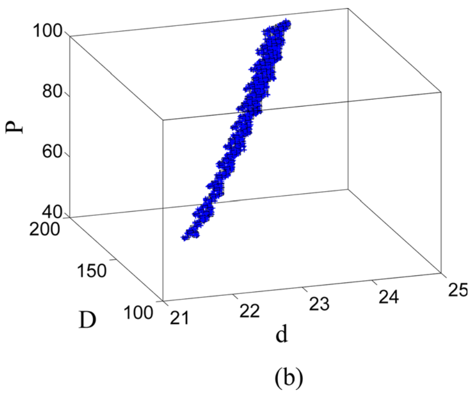


Fig. 4 Verification for the design space of suspension system (*a*) united solution set and (*b*) tolerable solution set

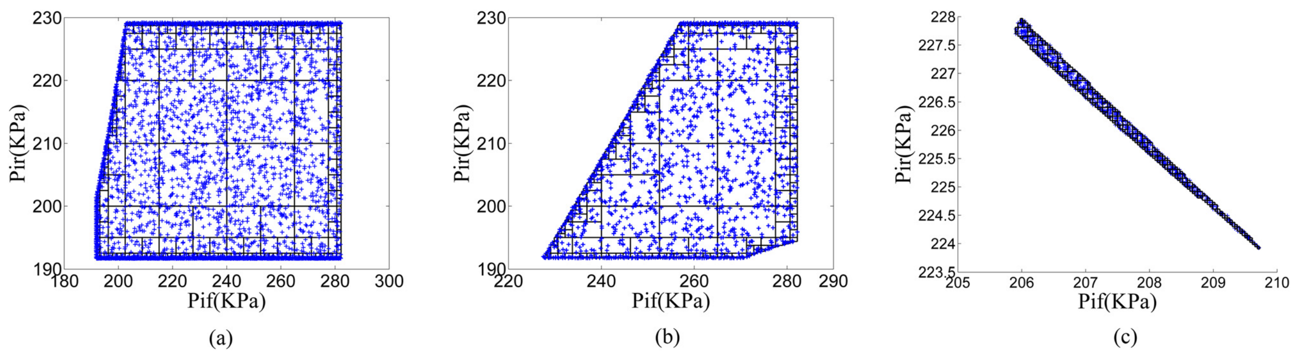


Fig. 5 Verification for the design space of tire system (*a*) united solution set, (*b*) tolerable solution set, and (*c*) controllable solution set

1500 samples were chosen for the tolerable solution set and distributed among the boxes of x proportionally to their volumes. For the united and controllable solution sets, a total number of 3000 samples were chosen. The results are presented in Figs. 4 and 5. In each figure, the boxes indicate the branch-and-prune algorithm results, and "$+$" are the sampled values in the verification.

It is seen that both the branch-and-prune method and hybrid stratified Monte Carlo method can be used to solve QCSPs. However, the stratified sampling requires more computational time to find a good estimation of solution than the branch-and-prune method, although the efficiency of the stratified sampling method has been improved significantly from the classical Monte Carlo sampling by using intervals instead of real values. Therefore, the sampling based method is not recommended to solve QCSPs.

United solution set is the largest one among the three quantified solution sets and encloses the other two. It thus gives us an upper limit of all feasible solutions. Therefore, the results of united solution set can be taken as the initial input to calculate the tolerable and controllable ones to improve computational efficiency.

6 Conclusions

A new QCSP formulation and solving approach have been developed to search feasible design solutions, which can incorporate more design intent than the traditional CSP. The different types of feasible solution sets (united, tolerable, and controllable) can be obtained by solving different logically quantified problems. The proposed branch-and-prune algorithm to solve QCSPs is based on the logic interpretation of generalized interval. There is no restriction on the number of occurrences for the variables. A hybrid stratified Monte Carlo method is also developed to verify the results obtained by the proposed algorithm. With the extra information from the interpretation, engineers can make better decisions at the different design stages. The complete knowledge of feasible design space is useful for complex problems of multidisciplinary design optimization.

Note that our approach is generic and can be applied to any constraint that is in the generic functional form of $f(a,x) = b$. The limitation of our approach is that it cannot solve over-constrained problems, for instance, when every variable is associated with $\forall$ and statements are overly quantified. In future work, the limitation needs to be addressed. Other branching methods, in addition to the bisection used in our algorithm, can also be investigated to improve the computational efficiency.

Acknowledgment

This research is supported in part by China Scholarship Council with a scholarship (No. 2011613059) to the first author.

References

[1] Eastman, C. M., 1973, "Automated Space Planning," Artif. Intell., **4**(1), pp. 41–64.
[2] Medjdoub, B., and Yannou, B., 2000, "Separating Topology and Geometry in Space Planning," Comput.-Aided Des., **32**(1), pp. 39–61.
[3] Dohmen, M., 1995, "A Survey of Constraint Satisfaction Techniques for Geometric Modeling," Comput. Graphics, **19**(6), pp. 831–845.
[4] Yannou, B., Moreno, F., Thevenot, H. J., and Simpson, T. W.,2005, "Faster Generation of Feasible Design Points," Proceedings of International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC/CIE2005), pp. 355–363.
[5] Titus, N., and Ramani, K., 2005, "Design Space Exploration Using Constraint Satisfaction," Papers From the Configuration Workshop at the 19th International Joint Conference on Artificial Intelligence (IJCAI'05), pp. 31–36.
[6] Sébastian, P., Chenouard, R., Nadeau, J. P., and Fischer, X., 2007, "The Embodiment Design Constraint Satisfaction Problem of the BOOTSTRAP Facing Interval Analysis and Genetic Algorithm Based Decision Support Tools," Int. J. Interact. Des. Manuf., **1**, pp. 99–106.
[7] Panchal, J. H., Gero Fernández, M., Paredis, C. J. J., Allen, J. K., and Mistree, F., 2007, "An Interval-Based Constraint Satisfaction (IBCS) Method for Decentralized, Collaborative Multifunctional Design," Concurr. Eng. Res. Appl., **15**(3), pp. 309–323.
[8] Yvars, P. A., 2009, "A CSP Approach for the Network of Product Lifecycle Constraints Consistency in a Collaborative Design Context," Eng. Applic. Artif. Intell., **22**(6), pp. 961–970.
[9] Lottaz, C., Sam-Haroud, D., Faltings, B., and Smith, I.,1998, "Constraint Techniques for Collaborative Design," Proceedings of IEEE International Conference on Tools With Artificial Intelligence, pp. 34–41.
[10] Devanathan, S., and Ramani, K., 2010, "Creating Polytope Representations of Design Spaces for Visual Exploration Using Consistency Techniques," ASME J. Mech. Des., **132**(8), p. 081011.
[11] Yan, X.-T., and Sawada, H., 2006, "A Framework for Supporting Multidisciplinary Engineering Design Exploration and Life-Cycle Design Using Underconstrained Problem Solving," Artif. Intell. Eng. Des. Anal. Manuf., **20**(4), pp. 329–350.
[12] Sam-Haroud, D., and Faltings, B., 1996, "Consistency Techniques for Continuous Constraints," Constraints, **1**(1–2), pp. 85–118.
[13] Börner, F., Bulatov, A., Jeavons, P., and Krokhin, A., 2003, "Quantified Constraints: Algorithms and Complexity," Comp. Sci. Logic, **2803**, pp. 58–70.
[14] Dantan, Y. J., 2005, "Tolerance Synthesis: Quantifier Notion and Virtual Boundary," Comput.-Aided Des., **37**, pp. 231–240.
[15] Qureshi, A. J., Dantan, J. Y., Bruyere, J., and Bigot, R., 2010, "Set Based Robust Design of Mechanical Systems Using the Quantifier Constraint Satisfaction Algorithm," Eng. Applic. Artif. Intell., **23**(7), pp. 1173–1186.
[16] Wang, Y., 2008, "Interpretable Interval Constraint Solvers in Semantic Tolerance Analysis," Comput.-Aided Des. Appl., **5**, pp. 654–666.
[17] Wang, Y., 2008, "Closed-Loop Analysis in Semantic Tolerance Modeling," ASME J. Mech. Des., **130**(6), p. 061701.
[18] Dantan, J. Y., and Qureshi, A. J., 2009, "Worst-Case and Statistical Tolerance Analysis Based on Quantified Constraint Satisfaction Problems and Monte Carlo Simulation," Comput.-Aided Des., **41**(1), pp. 1–12.
[19] Benhamou, F., Goualard, F., Languenou, E., and Cheristie, M., 2004, "Interval Constraint Solving for Camera Control and Motion Planning," ACM Trans. Comput. Logic, **5**(4), pp. 732–767.
[20] Jirstrand, M., 1997, "Nonlinear Control System Design by Quantifier Elimination," J. Symb. Comput., **24**, pp. 137–152.
[21] Herrero, P., Sainz, M. A., Vehi, J., and Jaulin, L., 2005, "Quantified Set Inversion Algorithm With Applications to Control," Reliab. Comput., **11**(5), pp. 369–382.
[22] Herrero, P., Sainz, M. Á., Vehí, J., and Jaulin, L.,2004, "Quantified Set Inversion With Applications to Control," Proceedings of IEEE International Symposium on Computer Aided Control Systems Design, pp. 179–183.
[23] Ratschan, S., and Vehı, J.,2003, "Robust Pole Clustering of Parametric Uncertain Systems Using Interval Methods," Proceedings of the 4th IFAC Symposium on Robust Control Design, S. Bittanti, and P. Colaneri, eds., pp. 323–328.
[24] Benedetti, M., Lallouet, A., and Vautard, J., 2008, "Modeling Adversary Scheduling With QCSP+," Proceedings of the 23rd Annual ACM Symposium on Applied Computing, ACM Press, pp. 151–155.

[25] Benedetti, M., Lallouet, A., and Vautard, J.,2007, "QCSP made Practical by Virtue of Restricted Quantification," Proceedings of the 20th International Joint Conference on Artiffcial Intelligence (IJCAI 2007), pp. 38–43.
[26] Sachenbacher, M., and Maier, P., 2008, "Test Strategy Generation Using Quantified CSPs," Proceedings of the 14th International Conference on Pingciples and Prractice of Active of Constraint Programming (CP-08), P. J. Stuckey, ed., Springer, New York, pp. 566–570.
[27] Sachenbacher, M., and Schwoon, S., 2008, "Model-Based Testing Using Quantified CSPs: A Map," Papers From the Workshop at the ECAI 2008 on Model-Based Systems, pp. 37–41.
[28] Gardeñes, E., Sainz, M. Á., Jorba, L., Calm, R., Estela, R., Mielgo, H., and Trepat, A., 2001, "Modal Intervals," Reliab. Comput., **7**(2), pp. 77–111.
[29] Dimitrova, N. S., Markov, S. M., and Popova, E. D., 1992, "Extended Interval Arithmetics: New Results and Applications," L. Atanassova and J. Herzberger, eds., Comp. Arith. Enclosure Methods, Elsevier Sci. Publishers, pp. 225–232.
[30] Simpson, T. W., Siddique, Z., and Jiao, J., 2006, *Product Platform and Product Family Design: Methods and Applications*, Springer-Verlag, New York.
[31] Nayak, R. U., Chen, W., and Simpson, T. W., 2002, "A Variation-Based Method for Product Family Design," Eng. Optimiz., **34**(1), pp. 65–81.
[32] Messac, A., Martinez, M. P., and Simpson, T. W., 2002, "Effective Product Family Design Using Physical Programming," Eng. Optimiz., **34**(3), pp. 245–261.
[33] Kaucher, E., 1980, "Interval Analysis in the Extended Interval Space IR," Comput. Suppl., **2**, pp. 33–49.
[34] Mackworth, A. K., and Eugnene, C. F., 1985, "The Complexity of Some Polynomial Network Consistency Algorithms for Constraint Satisfaction Problems," Artif. Intell., **25**(1), pp. 65–73.
[35] Golomb, S. W., and Baumert, L. D., 1965, "Backtrack Programming," J. ACM, **12**(4), pp. 516–524.
[36] Haralick, R., and Elliott, G., 1980, "Increasing Tree Search Efficiency for Constraint Satisfaction Problems," Artif. Intell., **14**, pp. 263–313.
[37] Sabin, D., and Freuder, E. C.,1994, "Contradicting Conventional Wisdom in Constraint Satisfaction," Proceedings of the 11th European Conference on Artifcial Intelligence (ECAI-94), A. G. Cohn, ed., John Wiley and Sons, UK, pp. 125–129.
[38] Apt, K. R., 1999, "The Essence of Constraint Propagation," Theor. Comput. Sci., **221**(1–2), pp. 179–210.
[39] Benhamou, F., Goualard, F., Granvilliers, L., and Puget, J.-F.,1999, "Revising Hull and Box Consistency," Proceedings of the 16th Iinternational Conference on Logic Programming, MIT Press, pp. 230–244.
[40] Benhamou, F., and Granvilliers, L., 2006, "Continuous and Interval Constraints," *Handbook of Constraint Programming*, F. Rossi, P. Van Beek, and T. Walsh, eds., Elsevier, UK, pp. 574–604.
[41] Mamoulis, N., and Stergiou, K.,2004, "Algorithms for Quantified Constraint Satisfaction Problems," Proceedings of the 10th International Conference on Pingciples and Prractice of Active of Constraint Programming (CP-04), M. Wallace, ed., Springer, New York, pp. 752–756.
[42] Gent, I. P., Nightingale, P., and Stergiou, K.,2005, "QCSP-Solve: A Solver for Quantified Constraint Satisfaction Problems," Proceedings of the 19th International Joint Conference on Artificial Intelligence, Lawrence Erlbaum Asspciates LTD., pp. 138–143.
[43] Gent, I. P., Nightingale, P., Rowley, A., and Stergiou, K., 2008, "Solving Quantified Constraint Satisfaction Problems," Artif. Intell., **172**(6–7), pp. 738–771.
[44] Bordeaux, L., Cadoli, M., and Mancini, T.,2005, "CSP Properties for Quantified Constraints: Definitions and Complexity," Proceedings of the 20th National Conference on Artificial Intelligence (AAAI-05), AAAI Press, pp. 360–366.
[45] Collins, G. E.,1975, "Quantifier Elimination for Real Closed Fields by Cylindrical Algebraic Decomposition," Proceedings of the 2nd GI Conference on Automata Theory and Formal Languages, pp. 134–183.
[46] Leonid, L.,2000, "Variable Independence, Quantifier Elimination, and Constraint Representations," Proceedings of the 27th International Colloquium Automata, Languages and Programming, U. Montanari, J. D. P. Rolim, and E. Welzl, eds., Springer-Verlag, pp. 260–271.
[47] John, K. A., and Chakraborty, S., 2011, "A Quantifier Elimination Algorithm for Linear Modular Equations and Disequations," Comput. Aided Verification, **6806**, pp. 486–503.
[48] Kupriyanova, L., 1995, "Inner Estimation of the United Solution Set of Interval Linear Algebraic System," Reliab. Comput., **1**(1), pp. 15–31.
[49] Shary, S. P., 1996, "Algebraic Approach to the Interval Linear Static Identification, Tolerance, and Control Problems, or One More Application of Kaucher Arithmetic," Reliab. Comput., **2**(1), pp. 3–33.
[50] Shary, S. P., 2002, "A New Technique in Systems Analysis Under Interval Uncertainty and Ambiguity," Reliab. Comput., **8**(5), pp. 321–418.
[51] Goldsztejn, A., 2005, "A Right-Preconditioning Process for the Formal–Algebraic Approach to Inner and Outer Estimation of AE-Solution Sets," Reliab. Comput., **11**(6), pp. 443–478.
[52] Grandon, C., and Goldsztejn, A., 2006, "Inner Approximation of Distance Constraints With Existential Quantification of Parameters," ACM Symposium on Applied Computing, pp. 1660–1661.
[53] Goldsztejn, A., and Jaulin, L., 2006, "Inner and Outer Approximations of Existentially Quantified Equality Constraints," Proceedings of the 16th international Conference on Principles and Practice of Constraint Programming, F. Benhamou, ed., Springer, NewYork, pp. 189–202.
[54] Goldsztejn, A.,2006, "A Branch and Prune Algorithm for the Approximation of Non-Linear AE-Solution Sets," Proceedings the 21th ACM Symposium on Applied computing (SAC-06), pp. 1650–1654.
[55] Goldsztejn, A., and Chabert, G., 2007, "A Generalized Interval LU Decomposition for the Solution of Interval Linear Systems," Numer. Methods Appl., **4310**, pp. 312–319.
[56] Goldsztejn, A., Michel, C., and Rueher, M., 2008, "Efficient Handling of Universally Quantified Inequalities," Constraint, **14**(1), pp. 117–135.
[57] Moore, R. E., 1966, *Interval Analysis*, Pretince Hall, New Jersey.
[58] Sobek, D. K., Ward, A. C., and Liker, J. K., 1999, "Toyota's Principles of Set-Based Concurrent Engineering," Sloan Manage. Rev., **40**(2), pp. 67–84.
[59] Moore, R. E., Kearfott, R. B., and Cloud, M. J., 2009, *Introduction to Interval Analysis*, Society for Industrial Mathematics, Philadelphia, PA.
[60] Kim, H. M., Rideout, D. G., Papalambros, P. Y., and Stein, J. L., 2003, "Analytical Target Cascading in Automotive Vehicle Design," ASME J. Mech. Des., **125**(3), pp. 481–489.
[61] Wong, J. Y., 2001, *Theory of Ground Vehicles*, Wiley-Interscience, New York.