Rigorous Global Optimization for Dynamic Systems Subject to Inequality Path Constraints

Yao Zhao and Mark A. Stadtherr[1]

Department of Chemical and Biomolecular Engineering
University of Notre Dame, Notre Dame, IN 46556, USA

May 6, 2011
(revised August 29, 2011)

[1]Author to whom all correspondence should be addressed. Phone: (574) 631-9318; Fax: (574) 631-8366; E-mail: markst@nd.edu

Abstract

A new approach is described for the rigorous global optimization of dynamic systems subject to inequality path constraints (IPCs). This method employs the sequential (control parameterization) approach and is based on techniques developed for the verified solution of parametric systems of ordinary differential equations. These techniques provide rigorous interval bounds on the state variables, and thus on the path constraints and objective function in the dynamic optimization problem. These techniques also provide explicit analytic representations (Taylor models) of these bounds in terms of the decision variables in the optimization problem. This facilitates the use of constraint propagation techniques that can greatly reduce the domain to be searched for the global optimum. Since IPCs are often related to safety concerns, we adopt a conservative, inner-approximation approach to constraint satisfaction. Through this approach, the search for the global optimum is restricted to a space in which continuous satisfaction of the IPCs is rigorously guaranteed, and an ϵ-global optimum within this space is determined. Examples are presented that demonstrate the potential and computational performance of this approach.

Keywords: Global optimization; Dynamic systems; Path constraints; Verified computing; Taylor models; Constraint propagation

1 Introduction

Optimization problems involving dynamic systems arise in many application areas in engineering and science. In practice it is often required that dynamic processes be operated to satisfy certain safety and/or quality control concerns over their entire operating time. Optimization problems in such systems are thus subject to appropriate inequality path constraints, which state that the operating trajectory may never enter regions of the state space that could cause process safety, product quality, or other problems. We will describe here a new approach for the rigorous global optimization of dynamic systems with inequality path constraints.

In optimization problems for dynamic systems, one or more of the decision variables may be a continuous function of time, resulting in essentially infinite-dimensional problems. To address this, there are in general two categories of methods. The classical *indirect* methods are based on solving Pontryagin's necessary conditions.[1,2] This often results in a two-point boundary value problem, and this may be as difficult to solve as the original optimization problem. In the other category are *direct* methods, which are based on problem formulations that use discretization to reduce the infinite-dimensional problem to a finite-dimensional one. There are two main approaches to the discretization, the *sequential* approach, in which only the decision variables (control parameters) are discretized,[3–8] and the *simultaneous* approach, in which both decision and state variables are discretized.[9–13] Direct methods enable easy incorporation of powerful numerical integration routines for systems of ordinary differential equations (ODEs) or differential-algebraic equations (DAEs) and of nonlinear programming (NLP) routines for optimization. Thus both the sequential and simultaneous approaches have been successfully applied to a wide range of problems, and both are in active use. Recently, for example, the optimization of periodic adsorption processes has been considered by Vetukuri et al.[14] using the sequential approach and by Agarwal et al.[15,16] using the simultaneous approach. Features of the two approaches can also be combined into a *quasi-sequential* approach.[17,18] In this paper, only the sequential approach will be considered.

The rigorous solution of dynamic optimization problems with inequality path constraints is challenging for two primary reasons. First, the presence of nonconvexities may lead to multiple local optima. This may become an issue for both constrained and unconstrained problems, and arises even in relatively simple problems.[19] Second, the inequality path constraints are continuously imposed; that is, they must be satisfied at all points in time, not just at discrete points. However, current methods (sequential or simultaneous) directly enforce the constraints only at the level of the

discretization used in the problem formulation or in the numerical routines used. Thus, rigorous, continuous satisfaction of the inequality path constraints remains an issue.

To handle the issue of multiple local optima, one approach is to use stochastic global optimization methods. For example, Banga et al.[20] have addressed constrained dynamic optimization problems using various stochastic global optimization methods, as well as a hybrid method that uses a stochastic approach globally with a gradient-based method locally, and this approach has been implemented in the MATLAB toolbox DOTcvp.[21] However, stochastic methods provide no theoretical guarantee of finding the global optimum. Thus there has also been significant recent interest, focused so far on unconstrained problems, in deterministic global optimization (DGO) methods designed to offer some such guarantee. An excellent review of DGO methods, covering many applications, has recently been provided by Floudas and Gounaris.[22] DGO methods are usually constructed on a branch-and-bound framework; thus, the rigor and efficiency (computational cost and bound tightness) of the bounding step has become a core issue, determining the overall performance of DGO methods. Convex relaxation methods for the bounding step have been described by Chachuat and Latifi,[23] Papamichail and Adjiman,[24,25] and Esposito and Floudas.[26,27] Both Chachuat and Latifi,[23] and Papamichail and Adjiman[24,25] obtain a theoretical guarantee of ϵ-global optimality, though at a very high computational cost. Singer and Barton[8] developed an approach, based on use of convex and concave relaxations to bound the state trajectories, that achieves ϵ-global optimality at significantly lower cost. Other recent ideas[28–31] for constructing convex and concave bounds of the states also may become important in the context of dynamic optimization. Lin and Stadtherr[32,33] developed a DGO algorithm based on using interval analysis and Taylor models to determine the state bounds.[34] Using this approach it is possible to obtain a rigorous guarantee of either exact global optimality or ϵ-global optimality.[32] The former requires the use of interval-Newton methods applied to the sensitivity equations. Either type of guarantee can be obtained at a computational cost that is often much less than that required by the approach of Singer and Barton.[8] The DGO methods discussed above are focused on unconstrained problems. In the work described below, we extend the DGO approach of Lin and Stadtherr to problems involving IPCs.

For addressing the satisfaction of IPCs in dynamic systems, most existing methods use certain types of transformations of the original problem. For example, a penalty function[35,36] may be introduced, either added directly to the objective function, or used to determine a set of end point constraints. Another approach is to introduce squared slack variables[37,38] to transform the

inequality constraints into equalities. The interior point approach transforms the continuous IPCs into a series of discrete point constraints.[6,39] Hybrid interior point and penalty function approaches have also been proposed.[40,41] Feehery and Barton[7,42] have developed a type of active set strategy, which detects dynamically whether the IPC is active or inactive, and solves either the appropriate unconstrained or equality constrained problem. With any of these approaches, there is no assurance that the IPC will be satisfied between discretization points, which may occur directly in the problem formulation (e.g., interior point approach) and/or in the numerical integration routines used to solve the underlying ODE or DAE problems. In the method described below, we will deal with this issue and present an approach that rigorously guarantees continuous, not just discrete, satisfaction of the IPCs.

In this study, we describe a new method for the rigorous global optimization of dynamic systems subject to inequality path constraints. This method employs the sequential approach and is based on the use of techniques[34] developed for the verified solution of parametric ODE systems. These techniques provide rigorous interval bounds on the state variables, and thus on the IPCs and objective function in the dynamic optimization problem. Furthermore, these techniques provide explicit analytic representations (Taylor models) of these bounds in terms of the decision variables in the optimization problem. This facilitates the use of constraint propagation techniques that can greatly reduce the space to be searched for the global optimum. Through this approach, the search for the global optimum is restricted to a space in which continuous satisfaction of the IPCs is rigorously guaranteed, and an ϵ-global optimum within this space is determined. The potential of this approach is demonstrated, and its performance studied, through the use of a number of computational examples.

The rest of this paper is structured as follows. In the next section we will formulate the problem to be solved. Then, in section 3, we will provide some background on interval analysis, Taylor models, and use of Taylor models in constraint propagation. In section 4, we will review verified solution methods for systems of parametric ODEs, and then in section 5 we will describe the new algorithm for rigorous global optimization of dynamic systems subject to IPCs. In section 6, the results of some numerical experiments will be presented and discussed, and finally in section 7 we will provide concluding remarks about the algorithm.

2 Problem Statement

In this section, the mathematical formulation of the optimization problem to be considered is given. Assume that the system to be optimized is described by an initial value problem (IVP) with the nonlinear ODE model $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\theta})$ and initial state $\boldsymbol{x}(t_0) = \boldsymbol{x}_0(\boldsymbol{\theta})$. Here $\boldsymbol{x}$ is the vector of state variables (length n) and $\boldsymbol{\theta}$ is a vector (length p) of adjustable parameters. The final state is denoted as $\boldsymbol{x}_\mathrm{f} = \boldsymbol{x}(t_\mathrm{f})$. The parameter vector $\boldsymbol{\theta}$ includes parameters that appear in the equations for $\dot{\boldsymbol{x}}$ and that appear in the statement of initial conditions (e.g., one or more of the initial states may be an adjustable parameter). For the example problems to be considered below, the initial states will be fixed and the parameter vector will result from the parameterization of a control profile $\theta(t)$. The model is expressed in autonomous form; a non-autonomous system can readily be converted to autonomous form by treating the independent variable (t) as an additional state variable. Systems described by DAE models will be considered elsewhere.

The optimization problem to be considered is

$$\begin{aligned} \min_{\boldsymbol{\theta}, \boldsymbol{x}_\eta} \quad & \phi = \phi\left[\boldsymbol{x}(t_\eta, \boldsymbol{\theta}), \boldsymbol{\theta};\ \eta = 0, 1, \ldots, r\right] \\ \text{s.t.} \quad & 0 \geq \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{\theta}) \\ & \dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\theta}) \\ & \boldsymbol{x}(t_0) = \boldsymbol{x}_0(\boldsymbol{\theta}) \\ & t \in [t_0, t_r] = [t_0, t_\mathrm{f}] \\ & \boldsymbol{\theta} \in \boldsymbol{\Theta}. \end{aligned} \tag{1}$$

Throughout this paper, boldface is used to indicate vector-valued quantities and uppercase is used to denote interval-valued quantities, unless noted otherwise. The objective ϕ is expressed in terms of the adjustable parameters $\boldsymbol{\theta}$ and the state values at discrete points t_η, $\eta = 0, 1, \ldots, r$. These discrete points may be specified in the problem formulation, or may arise in the numerical solution of the ODE model. If the objective function involves an integral, it can be eliminated using an appropriate quadrature variable. The IPCs are given by the function vector $\boldsymbol{g}$ (length m). It is assumed that $\boldsymbol{f}$ is $(\kappa - 1)$-times continuously differentiable with respect to the state variables $\boldsymbol{x}$, and $(q + 1)$-times continuously differentiable with respect to the parameters $\boldsymbol{\theta}$. It is also assumed that ϕ and $\boldsymbol{g}$ are $(q + 1)$-times continuously differentiable with respect to $\boldsymbol{\theta}$. Here κ is the order of the truncation error in the interval Taylor series (ITS) method to be used in the integration procedure (to be discussed in section 4), and q is the order of the Taylor model to be used to

represent parameter dependence (to be discussed in section 3.2). When the sequential approach is applied, a numerical ODE solver will provide values of $\boldsymbol{x}_\eta = \boldsymbol{x}(t_\eta, \boldsymbol{\theta})$, $\eta = 1, \ldots, r$ for specified values of $\boldsymbol{\theta}$, thus effectively eliminating $\boldsymbol{x}_\eta$, $\eta = 0, \ldots, r$ and leaving only the adjustable parameters $\boldsymbol{\theta}$ as decision variables. $\boldsymbol{\Theta}$ is a vector of interval bounds on the decision variables, thus defining the space to be searched for the global optimum. The feasible region, denoted as $\mathcal{G}$, is a subset of $\boldsymbol{\Theta}$, given by $\mathcal{G} = \{\boldsymbol{\theta} \in \boldsymbol{\Theta} \mid 0 \geq \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{\theta}),\ \forall t \in [t_0, t_\mathrm{f}]\}$.

We will use a branch-and-bound approach to address the problem stated above. To assure convergence in a finite number of steps, deterministic global optimization algorithms using this approach are typically terminated at a solution point that is within some specified tolerance of the true global minimum. For constrained problems, it is customary to use two such tolerances: 1. The value of the objective function at the solution point may not exceed the true global minimum value by more than some tolerance, usually denoted as ϵ; 2. The solution point may not lie outside of the feasible region by more than some tolerance. That is, the feasible region $\mathcal{G}$ is approximated by an *outer* approximation $\mathcal{G}^+ \supset \mathcal{G}$ that never differs from $\mathcal{G}$ by more than the tolerated amount. Using an outer approximation of the feasible region to determine constraint satisfaction has the advantage that all of $\mathcal{G}$ will be searched for the global minimum, but the disadvantage that the final solution point may in fact be outside of $\mathcal{G}$. In many contexts, such a small constraint violation is tolerable. However, it is not uncommon for IPCs to be safety related, thus dictating a more conservative approach to constraint satisfaction in this context. We thus choose to approximate the feasible region using an *inner* approximation $\mathcal{G}^- \subset \mathcal{G}$ that approaches $\mathcal{G}$ to within some tolerance. This has the advantage that the final solution point is guaranteed to be feasible, but the disadvantage that there may be very small regions of $\mathcal{G}$ that are not conclusively searched for an ϵ-global minimum. In choosing the inner-approximation approach, the search for the global minimum is restricted to the space $\mathcal{G}^-$ in which constraint satisfaction is rigorously guaranteed, and we seek to determine an ϵ-global optimum within this space. We will also bound the potential improvement in the global minimum that might occur if the search was expanded to $\mathcal{G}^+$.

3 Background

The approach presented here uses interval analysis, Taylor models, and constraint propagation with Taylor models. As background, a brief introduction to these topics is provided.

3.1 Interval analysis

A real interval $X = \left[\underline{X}, \overline{X}\right]$ is defined by $X = \left\{x \in \Re \mid \underline{X} \leq x \leq \overline{X}\right\}$. Here an underline is used to indicate the lower bound of an interval and an overline is used to indicate the upper bound. The width of an interval X is defined by $\mathrm{w}(X) = \overline{X} - \underline{X}$. A real interval vector $\boldsymbol{X} = (X_i) = (X_1, X_2, \cdots, X_n)^{\mathrm{T}}$ has n real interval components and can be interpreted geometrically as an n-dimensional rectangle. The width of an interval vector $\boldsymbol{X}$ is the width of its widest element; that is, $\mathrm{w}(\boldsymbol{X}) = \max_i \mathrm{w}(X_i)$. Basic arithmetic operations with intervals are defined by $X \text{ op } Y = \{x \text{ op } y \mid x \in X, y \in Y\}$, op $\in \{+, -, \times, \div\}$, with division in the case of $0 \in Y$ allowed only in extensions of interval arithmetic.[43] Interval versions of the elementary functions are similarly defined. The endpoints of an interval are computed with a directed (outward) rounding; that is, the lower bound is rounded down and the upper bound is rounded up. In this way, interval operations are guaranteed to produce bounds that are rigorous both mathematically and computationally. A number of good introductions to interval analysis and computing with intervals are available.[43–47]

For a real function $f(\boldsymbol{x})$, the interval extension $F(\boldsymbol{X})$ encloses the range of $f(\boldsymbol{x})$ for $\boldsymbol{x} \in \boldsymbol{X}$. A simple way to compute bounds on the range of $f(\boldsymbol{x})$ is to substitute the given interval $\boldsymbol{X}$ into the expression for $f(\boldsymbol{x})$ and then to evaluate with interval arithmetic. However, the tightness of the bounds in this "natural" interval extension depends on the form of the expression used to evaluate $f(\boldsymbol{x})$. If this is a single-use expression, in which no variable appears more than once, then the exact function range will be obtained (within roundout). However, if any variable appears more than once, then overestimation of the range may occur. Such overestimation is due to the "dependency" problem of interval arithmetic. While a variable may take on any value within its interval, it must take on the *same* value each time it occurs in an expression. However, this type of dependency is not detected when the natural interval extension is computed. Another source of overestimation that may occur in the use of interval methods is the "wrapping" effect. This occurs when an interval is used to enclose (wrap) a set of results that is not an interval. One approach that can be used to address both the dependency problem and the wrapping effect is the use of Taylor models.

3.2 Taylor models

Makino and Berz[48,49] have described a remainder differential algebra (RDA) approach for bounding function ranges and controlling the dependency problem of interval arithmetic. In this method, a function is represented by a model consisting of a Taylor polynomial expression and an interval remainder bound.

One way of forming a Taylor model of a function is by using the Taylor theorem. Consider a real function $f(\boldsymbol{x})$ that is $(q+1)$ times partially differentiable for $\boldsymbol{x} \in \boldsymbol{X}$ and let $\boldsymbol{x}_0 \in \boldsymbol{X}$. The Taylor theorem states that for each $\boldsymbol{x} \in \boldsymbol{X}$, there exists a real ζ with $0 < \zeta < 1$ such that

$$f(\boldsymbol{x}) = p_f(\boldsymbol{x} - \boldsymbol{x}_0) + r_f(\boldsymbol{x} - \boldsymbol{x}_0, \zeta), \tag{2}$$

where p_f is a q-th order polynomial (truncated Taylor series) in $(\boldsymbol{x}-\boldsymbol{x}_0)$ and r_f is a remainder, which can be quantitatively bounded over $0 < \zeta < 1$ and $\boldsymbol{x} \in \boldsymbol{X}$ using interval arithmetic or other methods to obtain an interval remainder bound R_f. A q-th order Taylor model $T_f(\boldsymbol{x}) = p_f(\boldsymbol{x} - \boldsymbol{x}_0) + R_f$ for $f(\boldsymbol{x})$ over $\boldsymbol{X}$ then consists of the polynomial expression p_f and the interval remainder bound R_f and is denoted by $T_f = (p_f, R_f)$. The expression $f(\boldsymbol{x})$ can now be bounded for $\boldsymbol{x} \in \boldsymbol{X}$ by seeking bounds on the Taylor model $T_f(\boldsymbol{x})$ for $\boldsymbol{x} \in \boldsymbol{X}$. This could be done using the natural interval extension $T_f(\boldsymbol{X})$, but usually tighter bounds can be obtained using other methods.[34,50–52] Following Lin and Stadtherr,[34] we use an approach in which the sum of all linear and diagonal quadratic terms is bounded exactly, with the remaining terms bounded using interval arithmetic.

In practice, it is more useful to compute Taylor models of functions by performing Taylor model operations. Arithmetic operations with Taylor models can be done using RDA operations,[48,49,53] which include addition, multiplication, reciprocal and intrinsic functions. Using these operations, it is possible to begin with simple functions such as the constant $f(\boldsymbol{x}) = k$, for which $T_f = (k, [0, 0])$, and the identity $f(x_i) = x_i$, for which $T_f = (x_{i0} + (x_i - x_{i0}), [0, 0])$, and to then compute Taylor models for more complicated functions. It has been reported that, compared to other rigorous range-bounding techniques, the use of Taylor models often provides tighter enclosures for functions with modest to complicated dependencies.[48,49,52] The applications and limitations of Taylor models are discussed in more detail elsewhere.[52]

3.3 Constraint propagation with Taylor models

The scheme[34] used in the bounding of Taylor models can be exploited in performing *constraint propagation* with Taylor models. Information provided by a constraint can be used to eliminate incompatible values from the domain of its variables. This domain reduction can then be propagated to all constraints on that variable, where it may be used to further reduce the domains of other variables. This is the process known as constraint propagation.[43,44,54] It is widely used in various forms in connection with interval methods. It has been shown previously how efficient constraint propagation schemes, based on hull consistency and using Taylor models, can be developed for

inequality constraints,[32] bound constraints,[55] and equality constraints.[56] In the method described below, we use the procedure given by Lin and Stadtherr[32] for constraint propagation with Taylor models on an inequality constraint $c(\boldsymbol{x}) \leq 0$, with $\boldsymbol{x} \in \boldsymbol{X}$. Using this constraint propagation procedure (CPP) with a Taylor model T_c of the constraint, a region $\mathcal{X}_{\rm nf} \subseteq \boldsymbol{X}$ that is *guaranteed* not to satisfy the constraint may be identified and removed from $\boldsymbol{X}$ to obtain an updated $\boldsymbol{X}$; that is, $\boldsymbol{X} \leftarrow \boldsymbol{X} \setminus \mathcal{X}_{\rm nf} = \{\boldsymbol{x} \in \boldsymbol{X} \mid \boldsymbol{x} \notin \mathcal{X}_{\rm nf}\}$. The region $\mathcal{X}_{\rm nf}$ is not necessarily an interval, but a set of intervals generated in such a way that $\boldsymbol{X} \setminus \mathcal{X}_{\rm nf}$ is an interval or an empty set. If the reduction in the size of $\boldsymbol{X}$ is sufficient (more than 10% reduction in volume is the heuristic used in our implementation), then the CPP is repeated. In repeating the CPP, we could use VSPODE to recompute the Taylor model of the constraint over the smaller $\boldsymbol{X}$; however, in our experience, this additional expense is usually not justified, as useful domain reduction may be obtained by continuing to use the initial Taylor model. The final outcome of the CPP may be that $\boldsymbol{X}$ is completely eliminated (becomes an empty set), or that $\boldsymbol{X}$ is reduced in size, or that $\boldsymbol{X}$ is not changed at all.

4 Bounding State Variables

When a standard sequential approach is applied to the dynamic optimization problem, the objective function $\phi\left[\boldsymbol{x}(t_\eta, \boldsymbol{\theta}), \boldsymbol{\theta};\ \eta = 0, 1, \ldots, r\right]$ is evaluated, for a given value of $\boldsymbol{\theta}$, by applying an ODE solver to the constraints to eliminate $\boldsymbol{x}(t_\eta, \boldsymbol{\theta})$, $\eta = 1, \ldots, r$. In the deterministic global optimization algorithm described here, we will use a branch-and-bound approach that requires the evaluation of bounds on ϕ, given some bounds $\boldsymbol{\theta} \in \boldsymbol{\Theta}$ on the adjustable parameters. Thus, as emphasized above, a key step is the determination of bounds on the states $\boldsymbol{x}(t_\eta, \boldsymbol{\theta})$, $\eta = 1, \ldots, r$ for $\boldsymbol{\theta} \in \boldsymbol{\Theta}$. For this purpose, we will use the method described by Lin and Stadtherr,[34] and implemented in a code called VSPODE. This is an interval method[57,58] (such methods are also called validated or verified methods) and also uses Taylor models. It is capable of determining mathematically and computationally guaranteed bounds on the state variables for nonlinear ODE systems with interval-valued parameters and/or initial states. We will briefly described the method here. Another interesting approach using Taylor models for state bounding has been described recently by Sahlodin and Chachuat.[31]

Focusing on the ODE constraint $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\theta})$, $\boldsymbol{x}_0 = \boldsymbol{x}_0(\boldsymbol{\theta})$, the goal is to determine bounds on the state variables $\boldsymbol{x}$ for $\boldsymbol{\theta} \in \boldsymbol{\Theta}$ and $\boldsymbol{x}_0 \in \boldsymbol{X}_0(\boldsymbol{\Theta})$. We denote by $\boldsymbol{x}(t; t_j, \boldsymbol{X}_j, \boldsymbol{\Theta})$ the set of

solutions $\boldsymbol{x}(t; t_j, \boldsymbol{X}_j, \boldsymbol{\Theta}) = \{\boldsymbol{x}(t; t_j, \boldsymbol{x}_j, \boldsymbol{\theta}) \mid \boldsymbol{x}_j \in \boldsymbol{X}_j, \boldsymbol{\theta} \in \boldsymbol{\Theta}\}$, where $\boldsymbol{x}(t; t_j, \boldsymbol{x}_j, \boldsymbol{\theta})$ denotes a solution of $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\theta})$ for the initial condition $\boldsymbol{x} = \boldsymbol{x}_j$ at t_j. As one outcome of the method outlined here, we will determine enclosures $\boldsymbol{X}_j$ of the state variables at a series of time steps $j = 1, \ldots, r$, such that $\boldsymbol{x}(t_j; t_0, \boldsymbol{X}_0, \boldsymbol{\Theta}) \subseteq \boldsymbol{X}_j$.

4.1 Phase one

Assume that at t_j we have an enclosure $\boldsymbol{X}_j$ of $\boldsymbol{x}(t_j; t_0, \boldsymbol{X}_0, \boldsymbol{\Theta})$, and that we want to carry out an integration step with stepsize $h_j = t_{j+1} - t_j$ to compute the next enclosure $\boldsymbol{X}_{j+1}$. Then, in the first phase (verification phase) of the method, the goal is to find an *a priori* enclosure $\widetilde{\boldsymbol{X}}_j$ of the solution such that a unique solution $\boldsymbol{x}(t; t_j, \boldsymbol{x}_j, \boldsymbol{\theta}) \in \widetilde{\boldsymbol{X}}_j$ is guaranteed to exist for all $t \in [t_j, t_{j+1}]$, all $\boldsymbol{x}_j \in \boldsymbol{X}_j$, and all $\boldsymbol{\theta} \in \boldsymbol{\Theta}$. This is done with a high-order interval Taylor series (ITS) with respect to time, using the Picard-Lindelöf operator and Banach fixed-point theorem. That is, we determine $\widetilde{\boldsymbol{X}}_j$ such that for $\boldsymbol{X}_j \subseteq \widetilde{\boldsymbol{X}}_j^0$,

$$\widetilde{\boldsymbol{X}}_j = \sum_{i=0}^{\kappa-1} [0, h_j]^i \boldsymbol{F}^{[i]}(\boldsymbol{X}_j, \boldsymbol{\Theta}) + [0, h_j]^\kappa \boldsymbol{F}^{[\kappa]}(\widetilde{\boldsymbol{X}}_j^0, \boldsymbol{\Theta}) \subseteq \widetilde{\boldsymbol{X}}_j^0. \tag{3}$$

Here κ denotes the order of the truncation error, $\widetilde{\boldsymbol{X}}_j^0$ is an initial estimate of $\widetilde{\boldsymbol{X}}_j$, and the coefficients $\boldsymbol{F}^{[i]}$ are interval extensions of the Taylor coefficients $\boldsymbol{f}^{[i]}$ of $\boldsymbol{x}(t)$ with respect to time, which can be obtained recursively in terms of $\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\theta})$ using automatic differentiation. This is an extension to parametric ODEs of the approach used in the popular VNODE method[57,59] for verified solution of ODEs. Satisfaction of eq. (3) verifies that there exists a unique solution $\boldsymbol{x}(t; t_j, \boldsymbol{x}_j, \boldsymbol{\theta}) \in \widetilde{\boldsymbol{X}}_j$ for all $t \in [t_j, t_{j+1}]$, all $\boldsymbol{x}_j \in \boldsymbol{X}_j$, and all $\boldsymbol{\theta} \in \boldsymbol{\Theta}$.

Note that $\widetilde{\boldsymbol{X}}_j$ rigorously encloses all the possible values of the state variables during an entire integration step, not only at t_{j+1}. We will use this enclosure in the optimization algorithm described below to enforce continuous satisfaction of the IPCs. In the usual VSPODE implementation[34] of this method, there are multiple options for the stepsize. An automatic stepsize adjustment procedure[57] is provided that will attempt to use a relatively large h_j. Alternatively, a fixed stepsize h can be specified by the user. However, this stepsize will be accepted as h_j at time t_j only if it results in satisfaction of eq. (3). Otherwise, VSPODE can be set either to continue with automatic adjustment, which will shrink the stepsize but still try to find a stepsize as large as possible, or to start over using a stepsize that has been reduced by some user-specified fraction. For the application of interest here, however, we find it useful to treat the stepsize adjustment procedure externally as part of the optimization algorithm; thus VSPODE will not adjust the stepsize and it is possible (if

h_j, $\boldsymbol{X}_j$ or $\boldsymbol{\Theta}$ is too large) for the verification stage to fail (eq. (3) cannot be satisfied). If such a failure occurs, it is dealt with in the optimization algorithm.

4.2 Phase two

In the second phase of the method, a tighter enclosure $\boldsymbol{X}_{j+1} \subseteq \widetilde{\boldsymbol{X}}_j$ is computed, such that $\boldsymbol{x}(t_{j+1}; t_0, \boldsymbol{X}_0, \boldsymbol{\Theta}) \subseteq \boldsymbol{X}_{j+1}$. This is done by using an ITS approach to compute a Taylor model $\boldsymbol{T}_{\boldsymbol{x}_{j+1}}$ of $\boldsymbol{x}_{j+1}$ in terms of the parameters $\boldsymbol{\theta}$. To do this, the parameters are first expressed as a Taylor model identity function $\boldsymbol{T}_{\boldsymbol{\theta}}$. Then, Taylor models $\boldsymbol{T}_{\boldsymbol{f}^{[i]}}$ of the interval Taylor series coefficients $\boldsymbol{f}^{[i]}(\boldsymbol{x}_j, \boldsymbol{\theta})$ can be determined by using RDA operations to compute $\boldsymbol{T}_{\boldsymbol{f}^{[i]}} = \boldsymbol{f}^{[i]}(\boldsymbol{T}_{\boldsymbol{x}_j}, \boldsymbol{T}_{\boldsymbol{\theta}})$. Using an interval Taylor series for $\boldsymbol{x}_{j+1}$ with coefficients given by $\boldsymbol{T}_{\boldsymbol{f}^{[i]}}$, and using the mean value theorem, one can obtain a result for $\boldsymbol{T}_{\boldsymbol{x}_{j+1}}$ in terms of the parameters $\boldsymbol{\theta}$. In this process, the wrapping effect is reduced by using a new type of Taylor model that uses a parallelepiped (rather than interval) remainder bound. The resulting Taylor model $\boldsymbol{T}_{\boldsymbol{x}_{j+1}}(\boldsymbol{\theta})$ can then be bounded[34] over $\boldsymbol{\theta} \in \boldsymbol{\Theta}$ to obtain $\boldsymbol{X}_{j+1}$. Complete details of the computation of $\boldsymbol{T}_{\boldsymbol{x}_{j+1}}$ are given by Lin and Stadtherr.[34]

4.3 Performance

As seen in eq. (3) for the phase one enclosure $\widetilde{\boldsymbol{X}}_j$, the parameter interval $\boldsymbol{\Theta}$ and the stepsize interval $[0, h_j]$ appear repeatedly in expressions that must be evaluated using interval arithmetic during execution of the VSPODE algorithm. Thus, overestimation of bounds resulting from the dependency problem (section 3.1) of interval arithmetic is inevitable. In the case of a relatively large parameter interval and/or a coarse stepsize, the propagation from time step to time step of relatively loose state enclosures may ultimately lead to verification failure in the first phase (eq. (3) can no longer be satisfied or requires a stepsize h_j that is impractically small). In the global optimization application, the parameter interval represents the decision variable space to be searched, and this may initially be quite large. However, since we will use a branch-and-bound framework for the global optimization algorithm, resulting in repeated subdivision of the search domain, the parameter intervals that must be considered will quickly reach a manageable size. Even if the parameter interval has zero width, some overestimation of the state bounds will still occur due to the stepsize interval. While, as discussed above, there are automatic stepsize adjustment procedures available in VSPODE, we prefer, for purposes of the constrained optimization problem considered here, to take more direct control of the stepsize. To do this, we will select a fixed stepsize h based on the ability to achieve a user-specified resolution in the constraint values. This process

will be implemented as part of an optional "autotuning" procedure for the optimization algorithm, to be discussed in section 5.5.

Two other settings in VSPODE that will affect performance of the optimization procedure are the ITS truncation order κ and the polynomial order q of the Taylor models used. In general, use of a relatively high κ will provide a smaller truncation error and thus permit the use of a larger stepsize in satisfying eq. (3). However, the computational cost of phase one is roughly proportional to κ, so there is also motivation to use a κ that is not unnecessarily large. The use of Taylor models to propagate the state enclosures from one time step to the next allows many of the necessary operations to be performed on the symbolic part of the Taylor model, rather than on intervals, thereby significantly reducing overestimation due to interval dependency and wrapping. Thus, a larger Taylor model order q (more symbolic terms) may lead to reduced propagation of overestimation from time step to time step. This is important since it may delay, or even prevent, the accumulation of bound overestimates that may lead to verification failure, thus extending the time over which the problem can be successfully integrated. However, the computational cost of computing with Taylor models increases significantly with increasing q due to the increasing number of terms in the polynomial part of the Taylor model. For example, if there are $p = 4$ parameters and $q = 3$, then the number of polynomial terms is 35, but for $q = 5$ this increases to 126 terms (the number of terms in a polynomial of order q in p variables is $(q + p)!/(q!p!)$). As part of the autotuning procedure, we will use a relatively large κ and q while determining the stepsize h. Then, once h has been determined, we will try to reduce κ and q while maintaining satisfaction of the user-specified constraint resolution.

5 Global Optimization Algorithm

In this section, we present a new method for the rigorous global optimization of dynamic systems subject to IPCs. As noted previously, when a sequential approach is used, the state variables are effectively eliminated from the objective function using the ODE constraints, leaving a constrained minimization of $\phi(\boldsymbol{\theta})$ with respect to the adjustable parameters (decision variables) $\boldsymbol{\theta}$. The new global optimization algorithm is constructed on a branch-and-bound framework with domain reduction using constraint propagation with Taylor models. Branch-and-bound with domain reduction is sometimes referred to as a branch-and-reduce approach.[60]

As discussed in section 2, our goal is to determine an ϵ-global minimum of $\phi(\boldsymbol{\theta})$ within an inner

approximation $\mathcal{G}^-$ of the feasible region $\mathcal{G}$. Specifically, the feasible region will be approximated, as needed, by an inner subpaving of precision δ. An inner subpaving $\mathcal{G}^-$ is a set of nonoverlapping subintervals, all of which can be rigorously proven to belong to the feasible region $\mathcal{G}$, and which excludes all subintervals that can be proven not be in $\mathcal{G}$ and all subintervals for which membership in $\mathcal{G}$ cannot be decided. An outer subpaving $\mathcal{G}^+$ contains $\mathcal{G}^-$ and those subintervals for which membership in $\mathcal{G}$ cannot be decided. The subpaving precision δ is the width of the largest undecided subinterval. As δ is reduced, $\mathcal{G}^-$ will more closely approach $\mathcal{G}$, with δ providing a measure, in terms of the decision variables, of the closeness of approach. The concept of subpavings is described in detail, with several examples, by Jaulin et al.[44] An entire inner subpaving for the feasible region rarely needs to be generated since many subintervals are eliminated from the search domain based on their objective function values. In the procedure described below, the subpaving precision δ can be specified directly by the user, or it can be tuned (section 5.5) based on a specified constraint resolution parameter ω. In effect, ω will provide an approximate measure, in terms of the constraint values, of the closeness with which $\mathcal{G}^-$ must approach $\mathcal{G}$.

5.1 Initialization

In a branch-and-bound procedure, a key issue is how to initialize $\hat{\phi}$, an upper bound on the global minimum. This is often done by evaluating $\phi(\boldsymbol{\theta})$ at some feasible point $\hat{\boldsymbol{\theta}}$. One simple approach for selecting an appropriate $\hat{\boldsymbol{\theta}}$ is to evaluate $\phi(\boldsymbol{\theta})$ at one or more feasible $\boldsymbol{\theta}$ points, and to designate the best (lowest ϕ value) as $\hat{\boldsymbol{\theta}}$. However, finding and verifying a feasible point may be nontrivial. An alternative approach is to use available local optimization and stochastic global optimization tools to determine a $\hat{\boldsymbol{\theta}}$. In general, we would expect the latter approach to be preferable (this is shown computationally in section 6.1). For this purpose we use the MATLAB toolbox DOTcvp,[21] which solves constrained optimal control problems using a sequential approach. This toolbox provides a variety of local, stochastic global, and hybrid optimization methods together with initial value problem (IVP) solvers from SUNDIALS[61] for integration of the ODE constraints. SUNDIALS includes the Adams-Moulton formulas for nonstiff problems and backward differentiation formulas for stiff problems. Gradient information for use in the optimization can also be obtained from SUNDIALS either by finite difference or by using the sensitivity equations. We use the default method in DOTcvp for optimization, namely FMINCON[62] with a sequential quadratic programming algorithm, thus obtaining a result for $\hat{\boldsymbol{\theta}}$. It has been our experience, however, that the DOTcvp result may, in some cases, have slight violations of the path constraints (even when using

very tight error tolerances in the underlying numerical routines). This is apparently due to the use in DOTcvp of IVP solvers that are not rigorously guaranteed. Such constraint violations can be confirmed using VSPODE, which is a rigorously guaranteed IVP solver. This issue can easily be dealt with by systematically making small changes in the DOTcvp result until its feasibility can be confirmed using VSPODE. To evaluate $\phi(\hat{\boldsymbol{\theta}})$ rigorously, we use a procedure described in Section 5.3 that results in interval bounds on ϕ. The upper bound of this interval is then used as the initial $\hat{\phi}$.

An additional initialization step is to establish a work list (stack) $\mathcal{L}$. The work list $\mathcal{L}$ will contain a sequence of subintervals (boxes) that need to be tested, as described below, and initially $\mathcal{L} = \{\boldsymbol{\Theta}\}$, the entire specified parameter (decision variable) space. Each subinterval in $\mathcal{L}$ has a status flag that can have a value of TRUE, indicating a box that is rigorously guaranteed to satisfy the IPCs and may contain an ϵ-global minimum, a value of UNDECIDED, indicating a box that may satisfy the IPCs and may contain an ϵ-global minimum, or a value of FALSE, indicating that the box is rigorously guaranteed either not to satisfy the IPCs or not to contain an ϵ-global minimum. Once a box is marked FALSE it can be removed from $\mathcal{L}$ and is no longer tested. The initial element of $\mathcal{L}$ is marked UNDECIDED.

We also establish a list $\mathcal{L}^+$, initially empty, which will contain boxes in which testing for an ϵ-global minimum is terminated at an incomplete stage, together with objective function bounds over each such box. These are boxes of width less than the subpaving tolerance δ that belong to the outer subpaving $\mathcal{G}^+$, but not the inner subpaving $\mathcal{G}^-$ within which an ϵ-global minimum is sought. By saving these boxes, we can later establish a bound on the improvement in ϕ that is possible if $\mathcal{G}^+$ is used to approximate the feasible region instead of $\mathcal{G}^-$.

Finally, we set the convergence tolerance ϵ for the objective function, the subpaving precision δ for the feasible region, and the parameters used by VSPODE, namely the stepsize h, the ITS truncation order κ and the Taylor model order q. The subpaving precision and VSPODE parameters can be set by using an optional "autotuning" procedure based on the (user-specified) IPC resolution parameter ω. We will defer explanation of the optional autotuning procedure until after the details of the optimization algorithm are presented.

5.2 Rigorous IPC test

At the k-th iteration of the overall algorithm (to be discussed in section 5.4), a subinterval is removed from the front of $\mathcal{L}$ for testing. This subinterval is denoted $\boldsymbol{\Theta}^{(k)}$. The purpose of the

test procedure described in this section is to determine rigorously whether all $\boldsymbol{\theta} \in \boldsymbol{\Theta}^{(k)}$ lead to states that satisfy the IPCs for all $t \in [t_0, t_f]$, and, if not, to try to shrink $\boldsymbol{\Theta}^{(k)}$ by using constraint propagation. Only boxes flagged as UNDECIDED undergo this test. If a box is already marked TRUE, then it has already been proven that all $\boldsymbol{\theta} \in \boldsymbol{\Theta}^{(k)}$ lead to states that satisfy the IPCs for all $t \in [t_0, t_f]$ and so this test need not be performed again.

Since we will assume that the specified initial states satisfy the IPCs, the subinterval $\boldsymbol{\Theta}^{(k)}$ is first given a status of TRUE as of $t = t_0$. We then begin carrying out integration time steps using information obtained from VSPODE. Consider the time step beginning at $t = t_j$. Recall that in the first phase (verification phase) of VSPODE, rigorous interval bounds $\widetilde{\boldsymbol{X}}_j^{(k)}$ on the state variables can be obtained that are valid for all $\boldsymbol{\theta} \in \boldsymbol{\Theta}^{(k)}$ and for all $t \in [t_j, t_{j+1}]$, that is, for the entire integration time step, not just the endpoint t_{j+1}. Then, using interval arithmetic, the corresponding bounds $\widetilde{\boldsymbol{G}}_j^{(k)} = \boldsymbol{g}(\widetilde{\boldsymbol{X}}_j^{(k)}, \boldsymbol{\Theta}^{(k)})$ on the IPCs can be computed, and these are also valid over the entire time step. However, the bounds $\widetilde{\boldsymbol{X}}_j^{(k)}$ are often only relatively loose bounds, which may lead to poor performance in bounding the IPCs. To improve $\widetilde{\boldsymbol{X}}_j^{(k)}$, we will use an idea suggested by Lin and Stadtherr,[63] based on the observation that over a small time step it is likely that most, if not all, of the state trajectories will increase or decrease monotonically with time. This leads to the following procedure:

1. Compute $\widetilde{\boldsymbol{X}}_j^{(k)}$ using the first phase of VSPODE. If the verification step fails (eq. (3) cannot be satisfied) then integration with VSPODE cannot continue; thus $\boldsymbol{\Theta}^{(k)}$ is marked as UNDECIDED and the IPC test terminates (this is only likely to occur in early iterations, when $\boldsymbol{\Theta}^{(k)}$ may be relatively large). Otherwise continue with computation of $\boldsymbol{X}_{j+1}^{(k)}$ using the second phase of VSPODE.

2. For each variable $i = 1, \ldots, n$, compute bounds on $\dot{x}_i = f_i(\boldsymbol{x}, \boldsymbol{\theta})$ over the current time step by determining $F_i^{(k)} = f_i(\widetilde{\boldsymbol{X}}_j^{(k)}, \boldsymbol{\Theta}^{(k)})$ using interval arithmetic. If $\underline{F_i^{(k)}} \geq 0$ or $\overline{F_i^{(k)}} \leq 0$ (x_i monotonically increases or decreases with respect to time), then the interval hull of $X_{j,i}^{(k)}$ and $X_{j+1,i}^{(k)}$ can be taken as the improved (tightened) $\widetilde{X}_{j,i}^{(k)}$. This is done component by component until the entire improved $\widetilde{\boldsymbol{X}}_j^{(k)}$ is obtained.

This improved $\widetilde{\boldsymbol{X}}_j^{(k)}$ is now used to compute the IPC bounds $\widetilde{\boldsymbol{G}}_j^{(k)}$. By examining $\widetilde{\boldsymbol{G}}_j^{(k)}$, $\boldsymbol{\Theta}^{(k)}$ can be put into one of three categories:

1. If, for some IPC l, $\underline{\widetilde{G}_{j,l}^{(k)}} > 0$, then this is rigorous proof that the IPC $g_l \leq 0$ cannot be satisfied by any point $\boldsymbol{\theta} \in \boldsymbol{\Theta}^{(k)}$. Thus, we can flag $\boldsymbol{\Theta}^{(k)}$ as FALSE and stop the IPC test.

2. If $\overline{\widetilde{\boldsymbol{G}}_j^{(k)}} \leq 0$ and $\boldsymbol{\Theta}^{(k)}$ was TRUE at time t_j, then this is rigorous proof that, up to time t_{j+1}, all $\boldsymbol{\theta} \in \boldsymbol{\Theta}^{(k)}$ satisfy all of the constraints. Thus, we can flag $\boldsymbol{\Theta}^{(k)}$ as TRUE as of time t_{j+1}, and proceed to the next integration time step. Note that a box maintains TRUE status only if constraint satisfaction is rigorously guaranteed for all $\boldsymbol{\theta} \in \boldsymbol{\Theta}^{(k)}$ for all previous time steps in addition to the current one; that is, constraint satisfaction is guaranteed for all $t \in [t_0, t_{j+1}]$.

3. Otherwise, we flag $\boldsymbol{\Theta}^{(k)}$ as UNDECIDED, and proceed with additional testing based on information obtained in the second phase of VSPODE.

From the second phase of VSPODE, we can obtain a Taylor model $\boldsymbol{T}_{\boldsymbol{x}_{j+1}}^{(k)}(\boldsymbol{\theta})$ giving the states $\boldsymbol{x}_{j+1}$ at t_{j+1} in terms of the parameters (decision variables) $\boldsymbol{\theta}$. Using Taylor model arithmetic, we can then compute $\boldsymbol{T}_{\boldsymbol{g}_{j+1}}^{(k)}(\boldsymbol{\theta}) = \boldsymbol{g}_{j+1}(\boldsymbol{T}_{\boldsymbol{x}_{j+1}}^{(k)}, \boldsymbol{T}_{\boldsymbol{\theta}})$, a Taylor model of the IPCs at t_{j+1} in terms of $\boldsymbol{\theta}$. By now bounding[34] $\boldsymbol{T}_{\boldsymbol{g}_{j+1}}^{(k)}(\boldsymbol{\theta})$ over $\boldsymbol{\theta} \in \boldsymbol{\Theta}^{(k)}$, we obtain interval bounds $\boldsymbol{G}_{j+1}^{(k)}$ on the IPCs at t_{j+1}. These bounds may be significantly tighter than $\widetilde{\boldsymbol{G}}_j^{(k)}$, but are valid only at $t = t_{j+1}$ ($\widetilde{\boldsymbol{G}}_j^{(k)}$ gives bounds valid for $t \in [t_j, t_{j+1}]$). Based on this information, there are three cases to consider:

1. If for some IPC l, $\underline{\boldsymbol{G}_{j+1,l}^{(k)}} > 0$, then this is rigorous proof that at $t = t_{j+1}$ the constraint $g_l \leq 0$ is violated at all points $\boldsymbol{\theta} \in \boldsymbol{\Theta}^{(k)}$. Thus, we can flag $\boldsymbol{\Theta}^{(k)}$ as FALSE and stop the IPC test.

2. If $\overline{\boldsymbol{G}_{j+1}^{(k)}} \leq 0$, this shows that all the constraints hold for $\boldsymbol{\theta} \in \boldsymbol{\Theta}^{(k)}$ at $t = t_{j+1}$. But since we require constraint satisfaction at all points in time, not just at the discrete time-step endpoints, we will leave the UNDECIDED flag on $\boldsymbol{\Theta}^{(k)}$, and continue to the next integration time step.

3. Otherwise, use the IPC Taylor model $\boldsymbol{T}_{\boldsymbol{g}_{j+1}}^{(k)}(\boldsymbol{\theta})$ in a constraint propagation procedure (CPP) based on the constraint $\boldsymbol{g}_{j+1} \leq 0$, as discussed in section 3.3. The CPP will attempt to shrink, or perhaps completely eliminate, $\boldsymbol{\Theta}^{(k)}$ by removing regions from it for which it is impossible to satisfy the IPCs. If $\boldsymbol{\Theta}^{(k)}$ is completely eliminated then it is marked FALSE and the IPC test terminates. Otherwise, the updated (possibly smaller) $\boldsymbol{\Theta}^{(k)}$ retains the UNDECIDED flag, and we continue to the next integration time step. If $\boldsymbol{T}_{\boldsymbol{g}_{j+1}}^{(k)}(\boldsymbol{\theta})$ depends on only some of the components of $\boldsymbol{\theta}$, then this can be used to reduce the computational effort needed to apply the CPP.

The IPC test procedure described above is applied to $\boldsymbol{\Theta}^{(k)}$ beginning with $t_j = t_0$. Possible overall outcomes are:

1. Integration terminates early due to failure in the verification phase of VSPODE. In this case, $\boldsymbol{\Theta}^{(k)}$ will have a status of UNDECIDED, and will be bisected upon return to the main algorithm.

2. Integration terminates early due to proof that no $\boldsymbol{\theta} \in \boldsymbol{\Theta}^{(k)}$ can satisfy the IPCs. In this case, $\boldsymbol{\Theta}^{(k)}$ will have a status of FALSE and will not be further tested. On return to the main algorithm, a new subinterval will be taken from the front of $\mathcal{L}$ for testing.

3. Integration terminates at the specified final time t_f and $\boldsymbol{\Theta}^{(k)}$ has a status of TRUE. This is rigorous proof that all the IPCs are satisfied for all $\boldsymbol{\theta} \in \boldsymbol{\Theta}^{(k)}$ and for *all* $t \in [t_0, t_\mathrm{f}]$ (not just for discrete points between t_0 and t_f). On return to the main algorithm this box will be sent to the objective test.

4. Integration terminates at the specified final time t_f and $\boldsymbol{\Theta}^{(k)}$ has a status of UNDECIDED. In this case, $\boldsymbol{\Theta}^{(k)}$ may contain some values of $\boldsymbol{\theta}$ that satisfy the IPCs and some that do not. It is also possible that all $\boldsymbol{\theta} \in \boldsymbol{\Theta}^{(k)}$ satisfy the constraints, or that all $\boldsymbol{\theta} \in \boldsymbol{\Theta}^{(k)}$ do not satisfy the constraints, but that this could not be rigorously verified due to the overestimation of bounds. On return to the main algorithm this box will be sent to the objective test.

5.3 Objective test

The purpose of the test procedure described in this section is to determine rigorously whether the tested interval $\boldsymbol{\Theta}^{(k)}$ contains any points $\boldsymbol{\theta}$ at which $\hat{\phi}$, the current upper bound on the global minimum, may be improved by more than the tolerance ϵ. If possible, we will also seek to improve the value of $\hat{\phi}$ and to shrink $\boldsymbol{\Theta}^{(k)}$ by using constraint propagation.

By running VSPODE we can obtain Taylor models of the state variables over $\boldsymbol{\Theta}^{(k)}$ at all times needed to evaluate the objective function (if possible we will use VSPODE results obtained during the IPC test). Using Taylor model arithmetic, we can then compute $T_\phi^{(k)}(\boldsymbol{\theta})$, a Taylor model over $\boldsymbol{\Theta}^{(k)}$ of the objective function in terms of the decision variables $\boldsymbol{\theta}$. By now bounding[34] $T_\phi^{(k)}(\boldsymbol{\theta})$ over $\boldsymbol{\theta} \in \boldsymbol{\Theta}^{(k)}$, we obtain interval bounds $\Phi^{(k)}$ on the objective function over $\boldsymbol{\Theta}^{(k)}$. Based on these bounds, we proceed as follows:

1. If $(\hat{\phi} - \underline{\Phi^{(k)}}) \leq \epsilon$ then the current $\boldsymbol{\Theta}^{(k)}$ does not contain any points at which $\hat{\phi}$ can be improved by more than the tolerance ϵ. Thus, $\boldsymbol{\Theta}^{(k)}$ is marked as FALSE and the objective test is stopped.

2. If $\mathbf{\Theta}^{(k)}$ is TRUE at this point, then it may contain feasible points that can be used to improve $\hat{\phi}$ by more than ϵ. Therefore, we apply a local optimization procedure over $\mathbf{\Theta}^{(k)}$ to try to determine a better $\hat{\boldsymbol{\theta}}$. For this purpose, we use the bound-constrained quasi-Newton method L-BFGS-B[64] for local optimization, and DDASSL[65] as the integration routine to be called by L-BFGS-B. If a better $\hat{\boldsymbol{\theta}}$ is found, then we proceed to evaluate $\hat{\phi}$ rigorously at the new $\hat{\boldsymbol{\theta}}$. To do this, we run VSPODE on the degenerate interval input $\hat{\mathbf{\Theta}} = [\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\theta}}]$ and use the results to determine a Taylor model for ϕ over $\hat{\mathbf{\Theta}}$. Bounding this Taylor model yields the interval $\hat{\Phi}$ (this is nonzero width because it bounds truncation error in the ODE solver and because of outward rounding in the bounding process). Then, if $\overline{\hat{\Phi}} < \hat{\phi}$, a new upper bound on the global minimum is set as $\hat{\phi} = \overline{\hat{\Phi}}$. Since $\underline{\Phi^{(k)}}$ usually will be somewhat less than the true lower bound on ϕ over $\mathbf{\Theta}^{(k)}$, performing a local optimization in a TRUE box whenever $\hat{\phi} - \underline{\Phi^{(k)}} > \epsilon$, as done here, will lead to some unsuccessful attempts to improve $\hat{\phi}$. An alternative approach is to perform a local optimization only when $\overline{\Phi^{(k)}} < \hat{\phi}$, which guarantees that an improved $\hat{\phi}$ will be found. While the latter approach avoids futile attempts to improve $\hat{\phi}$, the tradeoff is that the former approach is likely to find improved $\hat{\phi}$ values more often, potentially leading to the elimination of some subintervals at an earlier point in the branch-and-bound process. At least for the example problems considered here, the former approach is usually somewhat more efficient.

3. Use the objective function Taylor model $T_{\phi}^{(k)}(\boldsymbol{\theta})$ in a CPP based on the constraint $\phi - \hat{\phi} + \epsilon \leq 0$, which must be satisfied for any point $\boldsymbol{\theta}$ to be a candidate for the global minimum. The CPP will try to reduce $\mathbf{\Theta}^{(k)}$ by removing regions from it for which it is not possible to satisfy this constraint. It is also possible, since the CPP involves a recursive process in which the $\Phi^{(k)}$ interval is tightened as $\mathbf{\Theta}^{(k)}$ is reduced, for the entire $\mathbf{\Theta}^{(k)}$ to be eliminated. In this case, $\mathbf{\Theta}^{(k)}$ is marked FALSE.

For a tested subinterval $\mathbf{\Theta}^{(k)}$, the potential overall outcomes of the objective test are:

1. The subinterval is marked FALSE because it has been shown rigorously to contain no points for which the possibly updated $\hat{\phi}$ can be improved by more than the tolerance ϵ.

2. The subinterval has been reduced in size without change in status (still either TRUE or UNDECIDED).

3. No change in the subinterval or its status has occurred.

5.4 Overall optimization algorithm

The overall global optimization algorithm proceeds as follows:

1. Initialize (section 5.1).
2. Set iteration counter $k = 0$ and (optional) perform automatic tuning (section 5.5).
3. Remove from the front of $\mathcal{L}$ a subinterval $\mathbf{\Theta}^{(k)}$ for testing . This subinterval may have status of either UNDECIDED or TRUE.
4. If $\mathbf{\Theta}^{(k)}$ has status UNDECIDED, then send it to the IPC test (section 5.2). Otherwise, run VSPODE and proceed directly to step 5.
   (a) If the IPC test terminates due to failure of the verification phase in VSPODE, go to step 6 (bisection).
   (b) If $\mathbf{\Theta}^{(k)}$ returns from the IPC test with status FALSE, go to step 7.
   (c) Otherwise, proceed to step 5.
5. Send $\mathbf{\Theta}^{(k)}$ to the objective test (section 5.3).
   (a) If $\mathbf{\Theta}^{(k)}$ returns from the objective test with status FALSE, go to step 7.
   (b) If $\mathbf{\Theta}^{(k)}$ returns from the objective test with status UNDECIDED and its width is less than the subpaving precision δ, then $\mathbf{\Theta}^{(k)}$ belongs to the outer approximation (subpaving) $\mathcal{G}^+$ but not to the inner approximation (subpaving) $\mathcal{G}^-$ in which an ϵ-global minimum is sought. Thus, no further testing of $\mathbf{\Theta}^{(k)}$ is done, and $\mathbf{\Theta}^{(k)}$ and $\Phi^{(k)}$ are added together to the list $\mathcal{L}^+$ of boxes belonging to $\mathcal{G}^+$ in which testing for an ϵ-global minimum was incomplete. Go to step 7.
   (c) If $\mathbf{\Theta}^{(k)}$ returns from the objective test with its volume reduced by 70% or more (compared to its volume at the beginning of step 4), then put the reduced $\mathbf{\Theta}^{(k)}$ at the front of $\mathcal{L}$, increment k by one, and go to step 3. The threshold of 70% volume reduction is a heuristic and may be adjusted if desired.
   (d) Otherwise, proceed to step 6.
6. Bisect $\mathbf{\Theta}^{(k)}$ using the component with the largest relative width. The resulting subintervals retain the status flag of the parent subinterval $\mathbf{\Theta}^{(k)}$ and are placed at the front of $\mathcal{L}$. Increment k by one and go to step 3.

7. If $\mathcal{L}$ is empty, proceed to step 8. Otherwise, increment k by one and return to step 3.

8. Examine objective bounds saved in $\mathcal{L}^+$ to determine the overall lower bound $\phi^+ = \min_{\Phi^{(i)} \in \mathcal{L}^+} \underline{\Phi^{(i)}}$ on ϕ over boxes in $\mathcal{L}^+$. Terminate.

A flowchart of the overall algorithm is given in Fig. 1. At the termination of this procedure, $\mathcal{L}$ is empty and the final $\hat{\phi}$ is an ϵ-global minimum ϕ^* of the dynamic optimization problem subject to IPCs. The corresponding ϵ-global minimizer $\boldsymbol{\theta}^*$ is the final $\hat{\boldsymbol{\theta}}$. In step 8, we analyze the list $\mathcal{L}^+$ of boxes, saved in step 5(b), not completely tested for an ϵ-global minimum because they belong to $\mathcal{G}^+$ but not to $\mathcal{G}^-$. Interval bounds on ϕ in each of these boxes have been generated previously in the objective test and saved in step 5(b). These bounds can be used to establish a lower bound ϕ^+ on the ϕ values possible over all boxes in $\mathcal{L}^+$. This provides a bound, $\phi^* - \phi^+ = \epsilon^+$ on the objective function improvement that is possible if $\mathcal{G}^+$ is used to approximate the feasible region instead of $\mathcal{G}^-$. It is possible, if the boxes in $\mathcal{L}^+$ were put there before the final $\hat{\phi}$ value is reached, that $\epsilon^+ \leq \epsilon$, meaning that ϕ^* is an ϵ-global minimum in both $\mathcal{G}^-$ and $\mathcal{G}^+$. If $\epsilon^+ > \epsilon$, then ϕ^* is an ϵ-global minimum in $\mathcal{G}^-$ and an ϵ^+-global minimum in $\mathcal{G}^+$. Should the global minimum in $\mathcal{G}^+$ be of interest (for reasons explained previously we are interested primarily in $\mathcal{G}^-$) and ϵ^+ is unacceptably large, then the appropriate boxes in $\mathcal{L}^+$ can be further processed, by applying the algorithm with smaller values of δ and using higher precision (smaller h) in VSPODE.

Note that the algorithm is constructed in a way that intertwines testing for IPC satisfaction and ϵ-global optimality. This is much more efficient than an approach in which one first identifies the domain that satisfies the IPCs and then searches that domain for the global optimum. By intertwining the use of the IPC test and the objective test we greatly reduce the need for state bounding and eliminate much redundant work. The algorithm uses a standard depth-first search approach. It is possible that a different approach for ordering the search process could be more efficient.

5.5 Tuning procedure

The parameters that must be set before executing the algorithm described above are the convergence tolerance ϵ for the optimization problem, the stepsize h, the ITS truncation order κ and Taylor model order q used in VSPODE, and the subpaving precision δ. In this subsection we will describe an optional procedure for automatic tuning of h, κ, q and δ, based on user-specified values of ϵ and of ω, a constraint resolution parameter. This constraint resolution parameter is used

only for the purpose of this autotuning procedure. The goal is to find a stepsize h and subpaving precision δ such that the smallest boxes investigated in the optimization algorithm are likely to cover an interval of IPC values within the resolution ω and an interval of objective function values within the tolerance ϵ. Also, to reduce the computational cost, we try to reduce κ and q (from their heuristic default values of $\kappa = 9$ and $q = 5$), while maintaining satisfaction of the ω and ϵ conditions. For doing all this, we use the heuristic procedure outlined below.

The procedure used is based on a random sample of σ points $\boldsymbol{\theta}_s$, $s = 1, \ldots, \sigma$, from the search domain $\boldsymbol{\Theta}$. By adjusting the number of sample points, the user can decide whether to invest more or less computational effort in this tuning process. Because the computational cost of finding a rigorous ϵ-global minimum can become quite large as the number of decision variables increases, we are willing in general to devote a significant computational effort to achieve a reasonable tuning. However, in our experience on the example problems considered below, a relatively small sample of $\sigma = 5$ points is adequate.

In the tuning procedure, first h is adjusted, then δ and finally κ and q, as follows:

1. (Stepsize) As discussed in section 4.3, because of the dependency problem of interval arithmetic, using too large a value of the stepsize h can lead to excessive overestimation of the state bounds and thus of the IPC bounds. We would like to choose h so that, on average, the width of the IPC bounds (including overestimation) does not exceed the specified IPC resolution parameter ω. To do this we will use an iterative process in which VSPODE is run at each iteration using each sampled $\boldsymbol{\theta}_s$ value, treated as a zero-width interval. By using zero-width parameter intervals we are able to focus on the effect of h alone on overestimation. At the u-th iteration the time step will be $h^{(u)}$ and the total number of time steps will be $\tau^{(u)} = (t_\mathrm{f} - t_0)/h^{(u)}$ (assumed to be an integer). After each iteration, the stepsize is reduced by a factor of 10, so $h^{(u+1)} = h^{(u)}/10$ and $\tau^{(u+1)} = 10\tau^{(u)}$.

   The iterative process is:

   (a) Initialize iteration counter $u = 0$ and set $h^{(u)}$ (user specified).

   (b) Run VSPODE with fixed stepsize $h = h^{(u)}$ for each sampled point $\boldsymbol{\theta}_s$, $s = 1, \ldots, \sigma$, thus obtaining for the time step beginning at t_j, $j = 0, \ldots, \tau^{(u)} - 1$ the state bounds $\widetilde{\boldsymbol{X}}_{j,s}^{(u)}$. If verification failure occurs in phase one of VSPODE at any time step, then go to (f).

   (c) Compute the corresponding IPC bounds $\widetilde{\boldsymbol{G}}_{j,s}^{(u)} = \boldsymbol{g}(\widetilde{\boldsymbol{X}}_{j,s}^{(u)}, \boldsymbol{\theta}_s)$ and determine the average

IPC width (over all samples and time steps)

$$\hat{\mathrm{w}}(\widetilde{\boldsymbol{G}}^{(u)}) = \frac{\sum_{s=1}^{\sigma} \sum_{j=0}^{\tau^{(u)}-1} \mathrm{w}(\widetilde{\boldsymbol{G}}_{j,s}^{(u)})}{\sigma\tau^{(u)}} \tag{4}$$

(d) If $\hat{\mathrm{w}}(\widetilde{\boldsymbol{G}}^{(u)}) \leq \omega$, stop iterating and accept the current stepsize $h^{(u)}$ as h.

(e) If $u = 0$, go to (f). Otherwise we compare the average IPC width at the current iteration with the average IPC width over the same time interval during the previous iteration. This is done to provide a measure of the sensitivity of the average IPC width to reduction in the stepsize. The time interval of one time step in the previous iteration corresponds to ten time steps in the current iteration. Thus we must take the IPC bounds in the current iteration and unite them ten at a time, corresponding to the time steps in the previous iteration. This can be expressed by

$$\widetilde{\boldsymbol{\Gamma}}_{l,s}^{(u)} = \bigcup_{j=10l}^{10l+9} \widetilde{\boldsymbol{G}}_{j,s}^{(u)} \tag{5}$$

where $l = 0, \ldots, \tau^{(u-1)} - 1$. Then the average IPC width for comparison to the previous iteration is

$$\hat{\mathrm{w}}(\widetilde{\boldsymbol{\Gamma}}^{(u)}) = \frac{\sum_{s=1}^{\sigma} \sum_{l=0}^{\tau^{(u-1)}-1} \mathrm{w}(\widetilde{\boldsymbol{\Gamma}}_{l,s}^{(u)})}{\sigma\tau^{(u-1)}}. \tag{6}$$

If there has been relatively little improvement in the average IPC width, say less than half of the tolerance ω,

$$\hat{\mathrm{w}}(\widetilde{\boldsymbol{G}}^{(u-1)}) - \hat{\mathrm{w}}(\widetilde{\boldsymbol{\Gamma}}^{(u)}) \leq \frac{\omega}{2}, \tag{7}$$

then we stop iterating and use the stepsize $h^{(u-1)}$ from the previous iteration as h, since using the larger stepsize will save computational effort with relatively little difference in IPC bound quality. If there has been somewhat greater improvement in the average IPC width, say

$$\frac{\omega}{2} \leq \hat{\mathrm{w}}(\widetilde{\boldsymbol{G}}^{(u-1)}) - \hat{\mathrm{w}}(\widetilde{\boldsymbol{\Gamma}}^{(u)}) \leq \omega, \tag{8}$$

then stop iterating and accept the current stepsize $h^{(u)}$ as h. Otherwise, continue.

(f) Set $h^{(u+1)} = h^{(u)}/10$ and $\tau^{(u+1)} = 10\tau^{(u)}$.

(g) Increment u by one and return to (b).

2. (Subpaving precision) In the second tuning step we adjust the subpaving precision δ. Here we try to account for overestimation of the state bounds, and thus the objective and IPC bounds, due to parameter interval width (see section 4.3). We would like to choose δ to be sufficiently small so that, on average, the width of the objective interval (including overestimation) does not exceed the objective tolerance ϵ and the width of the IPC interval (including overestimation) does not exceed the IPC resolution parameter ω. To do this we will use an iterative process in which VSPODE is run at each iteration using a small parameter box, centered on each sampled $\boldsymbol{\theta}_s$ value, $s = 1, \ldots, \sigma$. At the u-th iteration the small box width will be $\delta^{(u)}$. The small parameter boxes used are then $\boldsymbol{\Theta}_s^{(u)} = \boldsymbol{\theta}_s + [-\delta^{(u)}/2, \delta^{(u)}/2]$, $s = 1, \ldots, \sigma$. After each iteration the small box width will be reduced by a factor of 10, so $\delta^{(u+1)} = \delta^{(u)}/10$. The stepsize h used is the value determined in the previous tuning step, and the corresponding number of time steps is τ. The iterative process is:

   (a) Initialize iteration counter $u = 0$ and set $\delta^{(u)}$ (user specified).

   (b) Run VSPODE with fixed stepsize h for each small parameter box $\boldsymbol{\Theta}_s^{(u)}$, $s = 1, \ldots, \sigma$, thus obtaining for the time step beginning at t_j, $j = 0, \ldots, \tau - 1$ the state bounds $\widetilde{\boldsymbol{X}}_{j,s}^{(u)}$. If verification failure occurs in phase one of VSPODE at any time step, then go to (e).

   (c) Based on these state bounds compute the corresponding IPC bounds $\widetilde{\boldsymbol{G}}_{j,s}^{(u)}$ and objective bounds $\Phi_s^{(u)}$ using interval arithmetic. Determine the average IPC width $\hat{\mathrm{w}}(\widetilde{\boldsymbol{G}}^{(u)})$ using eq. (4) (with $\tau^{(u)} = \tau$), and the average objective width from

$$\hat{\mathrm{w}}(\Phi^{(u)}) = \frac{\sum_{s=1}^{\sigma} \mathrm{w}(\Phi_s^{(u)})}{\sigma}. \tag{9}$$

   (d) If $\hat{\mathrm{w}}(\Phi^{(u)}) \leq \epsilon$ and $\hat{\mathrm{w}}(\widetilde{\boldsymbol{G}}^{(u)}) \leq \omega$, stop iterating and accept the current small box tolerance $\delta^{(u)}$ as δ. Otherwise, continue.

   (e) Set $\delta^{(u+1)} = \delta^{(u)}/10$.

   (f) Increment u by one and return to (b).

3. (ITS truncation order and Taylor model order) Finally, to reduce the computational expense in running VSPODE, we try to reduce the ITS truncation order κ and the Taylor model order q, while still maintaining satisfaction of the conditions used to establish the values for h and δ in the previous two tuning steps. We start by reducing to $\kappa = 5$ and $q = 3$, the minimum values that, in our experience,[34] lead to good performance. For these trial values

of κ and q, we then repeat tuning steps 1 and 2 and check whether the criteria for accepting the established h and δ are still met. If so, the trial values of κ and q are accepted. Otherwise we systematically increase first q and then κ until these criteria are satisfied.

There certainly may be other ways to tune the algorithm settings. We have found this heuristic procedure useful to identify and control, before actually attempting the global optimization procedure, the impact of bound overestimation on a specific optimization problem. The autotuning procedure described is relatively expensive computationally, but is still reasonable in this context since the cost of finding a rigorous ϵ-global minimum may become quite large. While this tuning procedure can be considered optional, we recommend its use. In the next section, we will study the computational performance of the ϵ-global optimization algorithm described here.

6 Computational studies

In this section, we apply the algorithm outlined above to three semi-batch reactor problems, each adapted from systems studied by Srinivasan et al.[66] We use these examples to demonstrate the effectiveness of the approach proposed above, as well as to study various aspects of the algorithm. In all cases, there is an active IPC at the globally optimal solution. The first example involves feed rate optimization for a second-order exothermic reaction in an isothermal semi-batch reactor. We also use this problem to consider the effect of the initialization for $\hat{\phi}$. The second problem involves temperature optimization for an exothermic series reaction in a nonisothermal semi-batch reactor. This problem is also used to consider the effects of the IPC resolution parameter ω and stepsize parameter h. The third problem considers feed rate optimization for parallel reactions in an isothermal semi-batch reactor. This example is also used to study the effect of using different values for the ITS truncation order κ and Taylor model order q. All the example problems were solved on a single 2.6 GHz dual-core AMD Opteron CPU running Red Hat Linux. All reported CPU times are rounded to three significant digits. The entire algorithm was implemented in C++.

6.1 Example 1: Second-order exothermic reaction in an isothermal semi-batch reactor

In this section, we consider a feed rate optimization problem involving a second-order exothermic reaction in an isothermal semi-batch reactor. There are two IPCs, one a cooling-failure safety

constraint and the other a maximum volume constraint. We solved two versions of the problem: the first with the safety constraint only and the second with both the safety and volume constraints.

Consider the reaction

$$\mathrm{A} + \mathrm{B} \xrightarrow{k} \mathrm{C}$$

occurring in an isothermal semi-batch reactor fed a stream containing B, and operating at temperature T. For the problem specification considered here, B is the limiting reactant. The objective is to maximize the amount of product C at the final time t_f by manipulating the volumetric flow rate $\theta(t)$ of the feed stream, which contains B at concentration $x_{\mathrm{B,in}}$. The problem is formulated as follows:

$$\begin{array}{rrcl}
\min\limits_{\theta(t)} & \phi & = & x_\mathrm{A}(t_\mathrm{f})V(t_\mathrm{f}) - x_{\mathrm{A}0}V_0 \\
\mathrm{IPC(a)} & T_\mathrm{fail}(t) & = & T + x_\mathrm{B}(t)\left(\dfrac{-\Delta H}{\rho c_\mathrm{p}}\right) \leq T_\mathrm{max} \\
\mathrm{IPC(b)} & V(t) & \leq & V_\mathrm{max} \\
\text{s.t.} & \dot{x}_\mathrm{A} & = & -kx_\mathrm{A}x_\mathrm{B} - \dfrac{\theta}{V}x_\mathrm{A} \\
& \dot{x}_\mathrm{B} & = & -kx_\mathrm{A}x_\mathrm{B} + \dfrac{\theta}{V}(x_{\mathrm{B,in}} - x_\mathrm{B}) \\
& \dot{V} & = & \theta \\
& \theta & \in & [0, 30]\ \mathrm{mL/h} = [0, 0.03]\ \mathrm{L/h} \\
& t & \in & [t_0, t_\mathrm{f}] = [0, 20]\ \mathrm{h}.
\end{array} \tag{10}$$

Here the state variables are $x_\mathrm{A}(t)$ and $x_\mathrm{B}(t)$, the concentrations of A and B, respectively, and the volume $V(t)$. Other model parameters are defined, and values given, in Table 1, which also gives values of the initial states. In IPC(a), T_fail is the temperature that could be reached in the case of a cooling failure (assuming an adiabatic operation after failure). If a cooling failure were to occur, the feed of B could be stopped immediately but the B already in the reactor would continue to react, thus raising the temperature. To guarantee safe operation, T_fail must not be allowed to exceed a specified T_max.

Following the sequential (control parameterization) approach, the flow rate $\theta(t)$ was parameterized as a piecewise constant profile with p equal time intervals. For example, if $p = 3$, then there are three decision variables, θ_1, θ_2, and θ_3, representing the flow rates over the time intervals $[0, t_\mathrm{f}/3]$, $[t_\mathrm{f}/3, 2t_\mathrm{f}/3]$, and $[2t_\mathrm{f}/3, t_\mathrm{f}]$, respectively. For both versions of the problem considered, we solved several different optimization problems, corresponding to different values of p (from $p = 1$

to $p = 7$). All problems were solved to an absolute objective tolerance of $\epsilon = 10^{-4}$ mol, using an IPC resolution of $\omega = 10^{-3}$ [K for IPC(a); L for IPC(b)]. Based on the tuning procedure described above, we used $\delta = 10^{-6}$ L/h, $h = 0.1$ h, $\kappa = 5$ and $q = 3$.

For initializing $\hat{\phi}$, an upper bound on the global minimum, the two approaches noted in section 5.1 were used. In one approach, we randomly sampled points from the search space until a feasible point was found. The value of ϕ at that first feasible point was then used as $\hat{\phi}$. In the other approach, we obtained $\hat{\phi}$ using the local solver DOTcvp, with small adjustments of the DOTcvp result as needed to guarantee evaluation of ϕ at a feasible point (see section 5.1).

We first considered the version of this problem in which only the safety constraint is imposed. Results for the ϵ-global minimum ϕ^* (rounded to the nearest 10^{-5} mol) and the corresponding ϵ-global minimizer $\boldsymbol{\theta}^*$ (rounded to the nearest 10^{-6} L/h) are shown in Table 2 for the case of initialization by random sampling and in Table 3 for the case of initialization by the local solver. As expected, it is clearly more efficient to initialize by using the local solver; for this case we solved problems as large as $p = 7$. For these problems the local solver always provided a better (smaller) $\hat{\phi}$ than random sampling, thus allowing earlier elimination of more of the search domain. However, there is no guarantee that this will always be the case, as a local solver could converge to a poor local minimum. Results of the optimization from the different initializations differ slightly, but are the same within the absolute tolerance of $\epsilon = 10^{-4}$ mol. Such differences arise because use of different initial $\hat{\phi}$ values will result in the search domain being processed through bisection and constraint propagation in a different order.

We then considered the version of this problem in which both the safety and volume IPCs are imposed. The results obtained are shown in Tables 4 and 5 for the two different initialization methods used, and are similar, in terms of trends in computational performance, to those seen for the case of the single IPC. The presence of the additional constraint has little impact on the computational cost, which clearly depends strongly on p, the number of decision variables. The eventual worst-case exponential complexity with respect to p seen in these results, as well as those in Tables 2 and 3, reflects the fact that rigorous global optimization for nonlinear problems is in general an NP-hard problem.

The computation times reported in Tables 2–5 do not include the time spent in the autotuning procedure. In both versions of this problem, and with all the values of p considered, we applied this procedure starting with $\delta = 10^{-5}$ L/h, $h = 1.0$ h, $\kappa = 9$ and $q = 5$ and, using $\sigma = 5$ sample points, arrived at the tuned values of $\delta = 10^{-6}$ L/h, $h = 0.1$ h, $\kappa = 5$ and $q = 3$. The computation times

for autotuning ranged from about 30 s for $p = 1$ to about 950 s for $p = 7$. This suggests that use of this tuning procedure when solving relatively small problems may not be warranted. However, for the type of study considered here, in which optimization is done for increasingly fine control parameterizations, it appears that it is possible to do the tuning for one or two small values of p and then use the results in solving the much more difficult problems with larger p. The impact of the tuned parameters on computational performance is considered in more detail in the next two examples. In the 22 problems considered here, ϵ^+ exceeded $\epsilon = 10^{-4}$ mol in five cases, with largest value being $\epsilon^+ = 1.0042 \times 10^{-4}$ mol (for the $p = 3$ case from Table 4).

6.2 Example 2: Exothermic series reaction in a nonisothermal semi-batch reactor

In this section, we consider a temperature optimization problem involving an exothermic series reaction in a nonisothermal semi-batch reactor. The reactor is jacketed and rapid temperature adjustments can be made. However, the cooling capacity is limited, leading to an IPC on the rate of heat generation in the reactor.

Consider the series reaction

$$\mathrm{A} + \mathrm{B} \xrightarrow{k_1} \mathrm{C} \xrightarrow{k_2} \mathrm{D}$$

occurring in a nonisothermal semi-batch reactor fed a stream containing B, and operated with a temperature profile $T(t)$. The objective is to maximize the concentration of the product C at the final time t_f by manipulation of the temperature $T(t)$. The problem formulation is:

$$\begin{array}{lrcl}
\min\limits_{\theta(t)} & \phi & = & -x_\mathrm{C}(t_\mathrm{f}) \\
\mathrm{IPC} & q_\mathrm{rx}(t) & = & -\Delta H_1 k_1 x_\mathrm{A}(t) x_\mathrm{B}(t) V(t) - \Delta H_2 k_2 x_\mathrm{C}(t) V(t) \le q_\mathrm{rx,max} \\
\text{s.t.} & \dot{x}_\mathrm{A} & = & -k_1 x_\mathrm{A} x_\mathrm{B} - \dfrac{u}{V} x_\mathrm{A} \\
& \dot{x}_\mathrm{B} & = & -k_1 x_\mathrm{A} x_\mathrm{B} + \dfrac{u}{V}(x_\mathrm{B,in} - x_\mathrm{B}) \\
& \dot{x}_\mathrm{C} & = & k_1 x_\mathrm{A} x_\mathrm{B} - k_2 x_\mathrm{C} - \dfrac{u}{V} x_\mathrm{C} \\
& \dot{V} & = & u \\
& k_i & = & k_{i,0} \exp(\dfrac{-E_i}{RT}) = k_{i,0} \exp(\dfrac{-E_i \theta}{R}), \quad i = 1, 2 \\
& T & \in & [293.15, 323.15] \text{ K} \\
& \theta & \in & [3.095, 3.411] \times 10^{-3} \text{ K}^{-1} \\
& t & \in & [t_0, t_\mathrm{f}] = [0, 0.5] \text{ h}
\end{array} \tag{11}$$

Here the state variables are $x_{\mathrm{A}}(t)$, $x_{\mathrm{B}}(t)$ and $x_{\mathrm{C}}(t)$, the concentrations of A, B and C respectively, and the volume $V(t)$. We find it convenient to use the reciprocal temperature $\theta(t) = 1/T(t)$ as the manipulated variable. Other model parameters are defined, with values given, in Table 6, which also gives the initial states. In the IPC, q_{rx} is the rate of heat generation in the reactor. Because of limited cooling capacity, this must not exceed a specified $q_{\mathrm{rx,max}}$.

The reciprocal temperature $\theta(t)$ was parameterized with a piecewise constant control profile on an equally spaced mesh with the number of time intervals varying from $p = 1$ to $p = 3$. The problem was solved to an absolute objective tolerance of $\epsilon = 10^{-4}$ mol/L. We will use this example to consider the effect on performance of using different values of the IPC resolution parameter ω and stepsize h. Since ω is used only in the autotuning procedure, in part to determine h, in effect we will study the impact of the stepsize h on the performance of the global optimization algorithm. For an IPC resolution of $\omega = 1$ kJ/h, the tuning procedure gave $h = 10^{-3}$ h, and for $\omega = 0.1$ kJ/h the result was $h = 10^{-4}$ h. We will also consider the case of $h = 10^{-5}$ h. Other parameters were autotuned to $\delta = 10^{-7}$ K^{-1}, $\kappa = 5$ and $q = 3$. For the case of one time interval ($p = 1$), random sampling was used to initialize $\hat{\phi}$, with the same value used for all stepsizes. For $p > 1$, we used the adjusted DOTcvp result from the $p = 1$ case for the initialization, again with the same value used for all stepsizes. We initialized $\hat{\phi}$ in this way so that these values were not "too good"; this means an increased computational effort, making the effect of using different stepsize values easier to observe.

Results for the ϵ-global minimum ϕ^* (rounded to the nearest 10^{-5} mol/L), for the corresponding ϵ-global minimizer $\boldsymbol{\theta}^*$ (rounded to the nearest 10^{-8} K^{-1}), and for ϵ^+ are shown in Table 7 for the different values of h considered. We make the following observations:

1. Comparing the results for $\omega = 1$ kJ/h ($h = 10^{-3}$ h) and $\omega = 0.1$ kJ/h ($h = 10^{-4}$ h), we see that the results for ϕ^* differ by more than the tolerance ϵ. The subpaving tolerance δ is the same in both cases, but with the greater constraint resolution (tighter constraint bounding) in the $\omega = 0.1$ kJ/h ($h = 10^{-4}$ h) case, there are many fewer boxes of width less than δ that remain undecided. That is, more of the search domain can be assigned to $\mathcal{G}^-$ and removed from $\mathcal{G}^+$. Thus, in the $h = 10^{-4}$ h case, there is a slightly larger $\mathcal{G}^-$ over which continuous satisfaction of the IPC can be rigorously guaranteed, and so a slightly better value of ϕ^* can be found. In using an even smaller $h = 10^{-5}$ h, no further improvements in ϕ^* are noted (differences are within the ϵ tolerance).

2. For the lowest resolution case of $\omega = 1$ kJ/h ($h = 10^{-3}$ h), the values of ϵ^+ are significantly larger than ϵ, but, for the higher resolution cases, ϵ^+ is less than or only slightly larger than ϵ. As just explained, higher resolution (smaller h) results in a somewhat larger $\mathcal{G}^-$ and somewhat smaller $\mathcal{G}^+$. That is, as h is reduced, $\mathcal{G}^-$ and $\mathcal{G}^+$ approach each other (both become better approximations of $\mathcal{G}$). Thus, as h is reduced, an ϵ-global minimum in $\mathcal{G}^-$ is more likely to also be an ϵ-global minimum in $\mathcal{G}^+$.

3. Reducing overestimation of the IPC bounds and objective bounds by reducing the stepsize makes it easier to rigorously eliminate boxes in the IPC test and objective test. This reduces the number of iterations required (number of boxes that must be tested). However, there is a tradeoff—using a smaller stepsize will reduce the number of iterations, but each iteration will require more computation time.

4. When the autotuning option is used, and a specified IPC resolution parameter ω is used to tune h, the use of too loose a resolution (such as in the $\omega = 1$ kJ/h case) can increase the computation cost. This may seem counterintuitive, but can be understood in terms of bound overestimation. Toleration of too much overestimation can have a negative effect on computational performance.

6.3 Example 3: Parallel reactions in an isothermal semi-batch reactor

In this section, we consider a feed rate optimization problem involving the parallel reactions

$$\begin{aligned} \mathrm{A} + \mathrm{B} &\xrightarrow{k_1} \mathrm{C} \\ 2\mathrm{B} &\xrightarrow{k_2} \mathrm{D} \end{aligned}$$

occurring in an isothermal semi-batch reactor fed a stream containing B, and operating at temperature T. For safety reasons, the concentration of B in the reactor must not be allowed to exceed a specified maximum value. This constraint also helps maintain selectivity to the desired product C. The objective is to maximize the amount of product C at the final time t_f by manipulating the volumetric flow rate $\theta(t)$ of the feed stream, which contains B at concentration $x_{\mathrm{B,in}}$. The problem

formulation is as follows:

$$
\begin{array}{lrcl}
\min\limits_{\theta(t)} & \phi & = & x_{\mathrm{A}}(t_{\mathrm{f}})V(t_{\mathrm{f}}) - x_{\mathrm{A0}}V_0 \\
\text{IPC} & x_{\mathrm{B}}(t) & \leq & x_{\mathrm{B,max}} \\
\text{s.t.} & \dot{x}_{\mathrm{A}} & = & -k_1 x_{\mathrm{A}} x_{\mathrm{B}} - \dfrac{\theta}{V} x_{\mathrm{A}} \\
 & \dot{x}_{\mathrm{B}} & = & -k_1 x_{\mathrm{A}} x_{\mathrm{B}} - 2k_2 {x_{\mathrm{B}}}^2 + \dfrac{\theta}{V}(x_{\mathrm{B,in}} - x_{\mathrm{B}}) \\
 & \dot{V} & = & \theta \\
 & \theta & \in & [0, 1000]\ \mu\mathrm{L/min} = [0, 0.001]\ \mathrm{L/min} \\
 & t & \in & [t_0, t_{\mathrm{f}}] = [0, 250]\ \mathrm{min}
\end{array}
\tag{12}
$$

Here the state variables are $x_{\mathrm{A}}(t)$ and $x_{\mathrm{B}}(t)$, the concentrations of A and B, respectively, and the volume $V(t)$. Other model parameters are defined, and values given, in Table 8, which also gives values of the initial states. The flow rate $\theta(t)$ was parameterized as a piecewise constant profile with p equal time intervals. The problem was solved to an absolute objective tolerance of $\epsilon = 10^{-4}$ mol, using an IPC resolution of $\omega = 10^{-4}$ mol/L. Using steps 1 and 2 of the autotuning procedure, a stepsize of $h = 1.0$ min and a subpaving precision of $\delta = 10^{-7}$ L/min were obtained. We will use this example to study the effect on performance of the tuning parameters κ and q. Adjusted results from DOTcvp are used to initialize $\hat{\phi}$ in all cases.

We first fixed the ITS truncation order at $\kappa = 17$ and considered different Taylor model orders q, with results for the ϵ-global minimum ϕ^* (rounded to the nearest 10^{-5} mol/L) and the corresponding ϵ-global minimizer $\boldsymbol{\theta}^*$ (rounded to the nearest 10^{-7} L/min) shown in Table 9 for $p \leq 4$. We then fixed the Taylor model order at $q = 3$ and considered different ITS truncation orders κ, with results shown in Table 10, again for $p \leq 4$. The ranges of the κ and q values used covers values that we have found useful in various other applications[33,34,55,56,63] involving VSPODE. For each value of p, the ϵ-global optimum found is the same, within the ϵ tolerance, for all the combinations of κ and q values. We make the following observations:

1. Increasing either q or κ increases the computational expense. This is consistent with the expected VSPODE performance, as discussed in section 4.3.

2. Changing either q or κ has little effect on the number of iterations. In general, we would expect that use of a relatively high κ would permit the use of a larger stepsize in satisfying eq. (3) in the first phase of VSPODE. However, in this case, we have already tuned the stepsize to meet a specified IPC tolerance, and thus the stepsize is already sufficiently small

that increasing κ is not worthwhile. Similarly, by fixing a sufficiently small stepsize, we have effectively enabled use of a small value of q.

Finally, using $q = 3$ and $\kappa = 5$, we solved larger versions of this problem, with up to $p = 8$ decision variables. These results are presented in Table 11. Again we see that rigorous global optimization for nonlinear problems is in general an NP-hard problem. For the $p = 4$ cases (Tables 9 and 10) and for $p \geq 6$ (Table 11), ϵ^+ exceeded $\epsilon = 10^{-4}$ mol, with largest value being $\epsilon^+ = 1.272 \times 10^{-4}$ mol in the $p = 6$ case.

7 Concluding remarks

We have demonstrated here a new approach for the rigorous, deterministic global optimization of dynamic systems subject to IPCs. While various methods have been proposed for the deterministic global optimization of unconstrained dynamic systems, the approach presented here appears to be the first to address this constrained problem. The method employs a branch-and-reduce approach based on use of interval analysis and Taylor models. Unlike previous optimization methods for problems with IPCs, which enforce the IPCs only at the level of the discretization used in the problem formulation or in the numerical routines used, in this new approach the IPCs are guaranteed to be satisfied *continuously*, rather than just at discrete points in time. Since the IPCs may be safety related, a conservative approach is followed in which the search for the global optimum is restricted to a space in which continuous satisfaction of the IPCs is rigorously guaranteed, and an ϵ-global optimum within this space is determined.

A technique for the automatic tuning of the algorithm parameters was also suggested. This method is based on a stochastic sampling of the search domain and determines parameter values that are then applied over this entire domain. We anticipate that the computational performance of the global optimization algorithm can be improved by using an adaptive tuning, in which algorithm parameters may differ in different parts of the search domain, based perhaps on local sampling or current search box size.

Acknowledgements

This work was supported in part by the University of Notre Dame Center for Applied Mathematics. Computational resources were provided by the University of Notre Dame Center for Research Computing.

References

(1) Bryson, A. E.; Ho, Y. C. *Applied Optimal Control;* Hemisphere Publishing Corporation: New York, NY, 1975.

(2) Pontryagin, L. *The Mathematical Theory of Optimal Processes;* Interscience Publishers: New York, NY, 1962.

(3) Brusch, R. G.; Schappelle, R. H. Solution of highly constrained optimal control problems using nonlinear programming. *AIAA J.* **1973,** *11,* 135–136.

(4) Goh, C. J.; Teo, K. L. Control parametrization: A unified approach to optimal control problems with general constraints. *Automatica* **1988,** *24,* 3–18.

(5) Vassiliadis, V. S.; Sargent, R. W.; Pantelides, C. C. Solution of a class of multistage dynamic optimization problems. 1. Problems without path constraints. *Ind. Eng. Chem. Res.* **1994,** *33,* 2111–2122.

(6) Vassiliadis, V. S.; Sargent, R. W.; Pantelides, C. C. Solution of a class of multistage dynamic optimization problems. 2. Problems with path constraints. *Ind. Eng. Chem. Res.* **1994,** *33,* 2123–2133.

(7) Feehery, W. F.; Barton, P. I. Dynamic optimization with state variable path constraints. *Comput. Chem. Eng.* **1998,** *22,* 1241–1256.

(8) Singer, A. B.; Barton, P. I. Global optimization with nonlinear ordinary differential equations. *J. Global Optim.* **2006,** *34,* 159–190.

(9) Neuman, C.; Sen, A. A suboptimal control algorithm for constraint problems using cubic splines. *Automatica* **1973,** *9,* 601–613.

(10) Tsang, T. H.; Himmelblau, D. M.; Edgar, T. F. Optimal control via collocation and nonlinear programming. *Int. J. Control* **1975,** *21,* 763–768.

(11) Biegler, L. T. An overview of simultaneous strategies for dynamic optimization. *Chem. Eng. Process.* **2007,** *46,* 1043–1053.

(12) Biegler, L. T.; Cervantes, A. M.; Wächter, A. Advances in simultaneous strategies for dynamic process optimization. *Chem. Eng. Sci.* **2002,** *57,* 575–593.

(13) Toumi, A.; Engell, S.; Diehl, M.; Bock, H. G.; Schlöder, J. Efficient optimization of simulated moving bed processes. *Chem. Eng. Process* **2007,** *46,* 1067–1084.

(14) Vetukuri, S. R. R.; Biegler, L. T.; Walther, A. An inexact trust-region algorithm for the optimization of periodic adsorption processes. *Ind. Eng. Chem. Res.* **2010,** *49,* 12004–12013.

(15) Agarwal, A.; Biegler, L. T.; Zitney, S. E. Superstructure-based optimal synthesis of pressure swing adsorption cycles for precombustion CO_2 capture. *Ind. Eng. Chem. Res.* **2010,** *49,* 5066–5079.

(16) Agarwal, A.; Biegler, L. T.; Zitney, S. E. A superstructure-based optimal synthesis of PSA cycles for post-combustion CO_2 capture. *AIChE J.* **2010,** *56,* 1813–1828.

(17) Bartl, M.; Li, P.; Biegler, L. T. Improvement of state profile accuracy in nonlinear dynamic optimization with the quasi-sequential approach. *AIChE J.* **2011,** *57,* 2185–2197.

(18) Hong, W.; Wang, S.; Li, P.; Wozny, G.; Biegler, L. T. A quasi-sequential approach to large-scale dynamic optimization problems. *AIChE J.* **2006,** *52,* 255–268.

(19) Luus, R.; Cormack, D. E. Multiplicity of solutions resulting from the use of variational methods in optimal control problems. *Can. J. Chem. Eng.* **1972,** *50,* 309–311.

(20) Banga, J. R.; Balsa-Canto, E.; Moles, C. J. Dynamic optimization of bioprocesses: Efficient and robust numerical strategies. *J. Biotechnol.* **2005,** *117,* 409–419.

(21) Hirmajer, T.; Balsa-Canto, E.; Banga, J. R. DOTcvp: Dynamic optimization toolbox with control vector parameterization approach for handling continuous and mixed-integer dynamic optimization problems, Technical Report, Instituto de Investigaciones Marinas - CSIC, Vigo, Spain, 2010 (see http://www.iim.csic.es/∼dotcvp/).

(22) Floudas, C. A.; Gounaris, C. E. A review of recent advances in global optimization. *J. Global Optim.* **2009,** *45,* 3–38.

(23) Chachuat, B.; Latifi, M. A. A new approach in deteterministic global optimisation of problems with ordinary differential equations. In *Nonconvex Optimization and Its Application*; Kluwer Academic Publishers: Dordrecht, The Netherlands, 2004.

(24) Papamichail, I.; Adjiman, C. S. Global optimization of dynamic systems. *Comput. Chem. Eng.* **2004,** *28,* 403-415.

(25) Papamichail, I.; Adjiman, C. S. A rigorous global optimization algorithm for problems with ordinary differential equations. *J. Global Optim.* **2002,** *24,* 1–33.

(26) Esposito, W. R.; Floudas, C. A. Deterministic global optimization in nonlinear optimal control problems. *J. Global Optim.* **2000,** *17,* 97–126.

(27) Esposito, W. R.; Floudas, C. A. Global optimization for the parameter estimation of differential-algebraic systems. *Ind. Eng. Chem. Res.* **2000,** *39,* 1291–1310.

(28) Scott, J. K.; Barton, P. I. Tight, efficient bounds on the solutions of chemical kinetics models. *Comput. Chem. Eng.* **2010,** *34,* 717–731.

(29) Sahlodin, A. M.; Chachuat, B. Discretize-then-relax approach for state relaxations in global dynamic optimization. In *20th European Symposium of Computer Aided Process Engineering*, Vol. 28; Pierucci, S.; Buzzi Ferraris, G., Eds.; Elsevier: Oxford, UK, 2010.

(30) Sahlodin, A. M.; Chachuat, B. Discretize-then-relax approach for convex/concave relaxations of the solutions of parametric ODEs. *Appl. Num. Math.* **2011,** *61,* 803–820.

(31) Sahlodin, A. M.; Chachuat, B. Convex/concave relaxations of parametric ODEs using Taylor models. *Comput. Chem. Eng.* **2011,** *35,* 844–857.

(32) Lin, Y.; Stadtherr, M. A. Deterministic global optimization for parameter estimation of dynamic systems. *Ind. Eng. Chem. Res.* **2006,** *45,* 8438–8448.

(33) Lin, Y.; Stadtherr, M. A. Deterministic global optimization of nonlinear dynamic systems. *AIChE J.* **2007,** *53,* 866–875.

(34) Lin, Y.; Stadtherr, M. A. Validated solutions of initial value problems for parametric ODEs. *Appl. Num. Math.* **2007,** *57,* 1145–1162.

(35) Bryson, A. E. *Dynamic Optimization;* Pearson Education: Upper Saddle River, NJ, 1998.

(36) Xing, A.; Wang, C. Applications of the exterior penalty method in constrained optimal control problems. *Opt. Contr. Appl. Meth.* **1989,** *10,* 333–345.

(37) Jacobson, D.; Lele, M. A transformation technique for optimal control problems with a state variable inequality constraint. *IEEE T. Automat. Contr.* **1969,** *14,* 457–464.

(38) Jacobson, D.; Lele, M. New necessary conditions of optimality for control problems with state variable inequality constraints. *J. Math. Anal. Appl.* **1971,** *35,* 255–284.

(39) Chen, W. C.; Vassiliadis, V. S. Inequality path constraints in optimal control: A finite iteration ϵ-convergent scheme based on pointwise discretization. *J. Process Contr.* **2005,** *15,* 353–362.

(40) Charalambides, M. *Optimal Design of Integrated Batch Processes,* Ph.D. Thesis. University of London, London, UK, 1996.

(41) Loxton, R. C.; Teo, K. L.; Rehbock, V. Optimal control problems with a continuous inequality constraint on the state and the control. *Automatica* **2009,** *45,* 2250–2257.

(42) Feehery, W. F.; Barton, P. I. Dynamic simulation and optimization with inequality path constraints. *Comput. Chem. Eng.* **1996,** *20,* 707–712.

(43) Hansen, E.; Walster, G. W. *Global Optimization Using Interval Analysis;* Marcel Dekker: New York, NY, 2004.

(44) Jaulin, L.; Kieffer, M.; Didrit, O.; Walter, É. *Applied Interval Analysis;* Springer-Verlag: London, England, 2001.

(45) Kearfott, R. B. *Rigorous Global Search: Continuous Problems;* Kluwer Academic Publishers: Dordrecht, The Netherlands, 1996.

(46) Neumaier, A. *Interval Methods for Systems of Equations;* Cambridge University Press: Cambridge, England, 1990.

(47) Moore, R. E.; Kearfott, R. B.; Cloud, M. J. *Introduction to Interval Analysis;* SIAM Press: Philadelphia, PA, 2009.

(48) Makino, K.; Berz, M. Efficient control of the dependency problem based on Taylor model methods. *Reliab. Comput.* **1999,** *5,* 3–12.

(49) Makino, K.; Berz, M. Remainder differential algebras and their applications. In *Computational Differentiation: Techniques, Applications, and Tools*; Berz, M.; Bishof, C.; Corliss, G.; Griewank, A., Eds.; SIAM: Philadelphia, PA, 1996.

(50) Makino, K.; Berz, M. Taylor model range bounding schemes, Presented at Third International Workshop on Taylor Methods, Miami Beach, FL, USA, 2004 (see http://bt.pa.msu.edu/TM/Miami2004/TM-Slides/Miami-GO04.pdf).

(51) Makino, K.; Berz, M. Verified global optimization with Taylor model-based range bounders. *Transactions on Computers* **2005,** *11,* 1611–1618.

(52) Neumaier, A. Taylor forms—Use and limits. *Reliab. Comput.* **2003,** *9,* 43–79.

(53) Makino, K.; Berz, M. Taylor models and other validated functional inclusion methods. *Int. J. Pure Appl. Math.* **2003,** *4,* 379–456.

(54) Kearfott, R. B. Validated constraint solving—Practicalities, pitfalls, and new developments. *Reliab. Comput.* **2005,** *5,* 383–391.

(55) Lin, Y.; Stadtherr, M. A. Guaranteed state and parameter estimation for nonlinear continuous-time systems with bounded-error measurements. *Ind. Eng. Chem. Res.* **2007,** *27,* 7198–7207.

(56) Lin, Y.; Enszer, J. A.; Stadtherr, M. A. Enclosing all solutions of two-point boundary value problems for ODEs. *Comput. Chem. Eng.* **2008,** *32,* 1714–1725.

(57) Nedialkov, N. S.; Jackson, K. R.; Corliss, G. F. Validated solutions of initial value problems for ordinary differential equations. *Appl. Math. Comput.* **1999,** *105,* 21–68.

(58) Neher, M.; Jackson, K. R.; Nedialkov, N. S. On Taylor model based integration of ODEs. *SIAM J. Numer. Anal.* **2007,** *45,* 236–262.

(59) Nedialkov, N. S.; Jackson, K. R.; Pryce, J. D. An effective high-order interval method for validating existence and uniqueness of the solution of an IVP for an ODE. *Reliab. Comput.* **2001,** *7,* 449–465.

(60) Sahinidis, N. V.; Bliek, C.; Jermann, C.; Neumaier, A. Global optimization and constraint satisfaction: The branch-and-reduce approach. *Lect. Notes Comp. Sci.* **2003,** *2861,* 1–16.

(61) Hindmarsh, A. C.; Brown, P. N.; Grant, K. E.; Lee, S. L.; Serban, R.; Shumaker, D. E.; Woodward, C. S. SUNDIALS: Suite of nonlinear and differential/algebraic equation solvers. *ACM T. Math. Software* **2005,** *31,* 363–396.

(62) The MathWorks, Inc., Natick, MA, USA. Optimization Toolbox 5: User's Guide, 2010.

(63) Lin, Y.; Stadtherr, M. A. Rigorous model-based safety analysis for nonlinear continuous-time systems. *Comput. Chem. Eng.* **2009,** *33,* 493–502.

(64) Byrd, R. H.; Lu, P.; Nocedal, J.; Zhu, C. A limited memory algorithm for bound constrained optimization. *SIAM J. Sci Comput.* **1995,** *16,* 1190–1208.

(65) Brenan, K. E.; Campbell, S. L.; Petzold, L. R. *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations;* SIAM Press: Philadelphia, PA, 1996.

(66) Srinivasan, B.; Palanki, S.; Bonvin, D. Dynamic optimization of batch processes: I. Characterization of the nominal solution. *Comput. Chem. Eng.* **2003,** *27,* 1–26.

Table 1: Model parameters for Example 1.

Parameter	Description	Value	Units
k	Reaction rate constant	0.0482	L/(mol h)
x_{A0}	Initial concentration of A	2	mol/L
x_{B0}	Initial concentration of B	0.5	mol/L
V_0	Initial volume	0.7	L
$V_{\max}$	Maximum volume	1.1	L
$x_{\mathrm{B,in}}$	Inlet concentration of B	2	mol/L
T	Reactor temperature	343.15	K
$T_{\max}$	Maximum allowable temperature	353.15	K
c_{p}	Heat capacity	4.2	J/(g K)
ΔH	Heat of reaction	-60000	J/mol
ρ	Density	900	g/L

Table 2: Results for Example 1 with safety constraint only, and initialization of $\hat{\phi}$ with random sampling.

p	ϕ^* (mol)	$\boldsymbol{\theta}^*$ (L/h) $\times 10^2$	CPU time (s)	Iterations
1	-0.58445	(2.4648)	3.35	32
2	-0.60786	(3.0000, 2.1002)	3.60	30
3	-0.60930	(3.0000, 2.7349, 1.9238)	906	8183
4	-0.61060	(3.0000, 3.0000, 2.3578, 1.8789)	1370	10353

Table 3: Results for Example 1 with safety constraint only, and initialization of $\hat{\phi}$ with adjusted DOTcvp results.

p	ϕ^* (mol)	$\boldsymbol{\theta}^*$ (L/h) $\times 10^2$	CPU time (s)	Iterations
1	-0.58444	(2.4647)	0.87	9
2	-0.60785	(3.0000, 2.0997)	1.81	17
3	-0.60934	(3.0000, 2.7349, 1.9279)	21.9	195
4	-0.61055	(3.0000, 3.0000, 2.3578, 1.8706)	37.9	286
5	-0.61128	(3.0000, 3.0000, 2.8649, 2.0941, 1.8516)	493	3214
6	-0.61128	(3.0000, 3.0000, 3.0000, 2.5084, 2.0331, 1.8219)	1830	8671
7	-0.61176	(3.0000, 3.0000, 3.0000, 2.9744, 2.1658, 1.9910, 1.7925)	32600	107719

Table 4: Results for Example 1 with both safety and volume constraints, and initialization of $\hat{\phi}$ with random sampling.

p	ϕ^* (mol)	$\boldsymbol{\theta}^*$ (L/h) $\times 10^2$	CPU time (s)	Iterations
1	-0.54380	(2.0000)	0.60	6
2	-0.58356	(3.0000, 1.0000)	4.25	37
3	-0.59276	(3.0000, 2.7349, 0.2558)	33.4	284
4	-0.59432	(3.0000, 3.0000, 1.9965, 0.0029)	97.2	701

Table 5: Results for Example 1 with both safety and volume constraints, and initialization of $\hat{\phi}$ with adjusted DOTcvp results.

p	ϕ^* (mol)	$\boldsymbol{\theta}^*$ (L/h) $\times 10^2$	CPU time (s)	Iterations
1	-0.54372	(1.9991)	0.26	3
2	-0.58356	(3.0000, 1.0000)	2.59	24
3	-0.59278	(3.0000, 2.7349, 0.2574)	24.7	218
4	-0.59432	(3.0000, 3.0000, 1.9966, 0.0029)	49.4	369
5	-0.59460	(3.0000, 3.0000, 2.8649, 1.1309, 0.0014)	532	3342
6	-0.59536	(3.0000, 3.0000, 3.0000, 2.5084, 0.4835, 0.0046)	3320	14685
7	-0.59633	(3.0000, 3.0000, 3.0000, 2.9530, 2.0437, 0.0029, 0.0004)	40093	126243

Table 6: Model parameters for Example 2.

Parameter	Description	Value	Units
$k_{1,0}$	Pre-exponential factor	4	L/(mol h)
$k_{2,0}$	Pre-exponential factor	800	h^{-1}
E_1	Activation energy	6×10^3	J/mol
E_2	Activation energy	20×10^3	J/mol
R	Gas constant	8.3145	J/(mol K)
ΔH_1	Reaction enthalpy	-30	kJ/mol
ΔH_2	Reaction enthalpy	-10	kJ/mol
x_{A0}	Initial concentration of A	10	mol/L
x_{B0}	Initial concentration of B	0.0	mol/L
x_{C0}	Initial concentration of C	0.0	mol/L
V_0	Initial volume	1.0	L
$x_{\mathrm{B,in}}$	Inlet concentration of B	20	mol/L
$q_{\mathrm{rx,max}}$	Maximum heat generation rate	150	kJ/h
u	Feed rate of B	0.35	L/h

Table 7: Results for Example 2.

ω & h	p	ϕ^* (mol/L)	$\boldsymbol{\theta}^*$ (K^{-1}) $\times 10^3$	ϵ^+ (mol/L)	CPU time (s)	Iterations
$\omega = 1$ kJ/h	1	-1.35686	(3.26923)	9.04×10^{-4}	61.2	121
$h = 10^{-3}$ h	2	-1.37487	(3.09504, 3.23517)	5.81×10^{-4}	1330	1687
	3	-1.38568	(3.09504, 3.09504, 3.21285)	3.20×10^{-4}	17500	26250
$\omega = 0.1$ kJ/h	1	-1.35766	(3.26691)	1.051×10^{-4}	151	39
$h = 10^{-4}$ h	2	-1.37536	(3.09504, 3.23329)	$< \epsilon$	540	103
	3	-1.38594	(3.09502, 3.09502, 3.21152)	$< \epsilon$	1140	120
	1	-1.35770	(3.26679)	$< \epsilon$	1850	43
$h = 10^{-5}$ h	2	-1.37541	(3.09502, 3.23311)	$< \epsilon$	2670	67
	3	-1.38596	(3.09504, 3.09504, 3.21145)	$< \epsilon$	8700	95

Table 8: Model parameters for Example 3.

Parameter	Description	Value	Units
k_1	Reaction rate constant	0.053	L/(mol min)
k_2	Reaction rate constant	0.128	L/(mol min)
x_{A0}	Initial concentration of A	0.72	mol/L
x_{B0}	Initial concentration of B	0.05	mol/L
V_0	Initial volume	1.0	L
$x_{\mathrm{B,in}}$	Inlet concentration of B	5	mol/L
$x_{\mathrm{B,max}}$	Maximum concentration of B	0.06	mol/L

Table 9: Results for Example 3 with fixed ITS truncation order $\kappa = 17$ and different Taylor model orders q.

q	p	ϕ^* (mol/L)	$\boldsymbol{\theta}^*$ (L/min) $\times 10^4$	CPU time (s)	Iterations
3	1	−0.36074	(4.526)	11.3	25
	2	−0.37835	(5.371, 4.424)	21.5	45
	3	−0.38447	(5.749, 4.976, 4.379)	562	1090
	4	−0.38741	(5.970, 5.301, 4.786, 4.355)	894	1488
5	1	−0.36074	(4.526)	17.4	31
	2	−0.37831	(5.370, 4.422)	70.4	99
	3	−0.38442	(5.748, 4.9751, 4.383)	844	815
	4	−0.38742	(5.963, 5.310, 4.785, 4.355)	2500	1374
7	1	−0.36074	(4.526)	22.2	31
	2	−0.37828	(5.370, 4.421)	109	102
	3	−0.38446	(5.748, 4.976, 4.378)	1640	672
	4	−0.38741	(5.960, 5.311, 4.785, 4.355)	8150	1266

Table 10: Results for Example 3 with fixed Taylor model order $q = 3$ and different ITS truncation orders κ.

κ	p	ϕ^* (mol/L)	$\boldsymbol{\theta}^*$ (L/min) $\times 10^4$	CPU time (s)	Iterations
5	1	−0.36074	(4.526)	3.46	25
	2	−0.37828	(5.371, 4.420)	15.0	100
	3	−0.38446	(5.749, 4.976, 4.380)	157	981
	4	−0.38741	(5.970, 5.306, 4.779, 4.355)	247	1393
11	1	−0.36074	(4.526)	6.53	25
	2	−0.37834	(5.370, 4.424)	27.6	92
	3	−0.38445	(5.748, 4.976, 4.379)	252	845
	4	−0.38741	(5.970, 5.301, 4.786, 4.353)	393	1154
17	1	−0.36074	(4.526)	11.3	25
	2	−0.37835	(5.371, 4.424)	21.5	45
	3	−0.38447	(5.749, 4.976, 4.379)	562	1090
	4	−0.38741	(5.970, 5.301, 4.786, 4.355)	894	1488

Table 11: Results for Example 3 with $\kappa = 5$ and $q = 3$.

p	ϕ^* (mol/L)	$\boldsymbol{\theta}^*$ (L/min) $\times 10^4$	CPU time (s)	Iterations
5	−0.38919	(6.123, 5.531, 5.067, 4.667, 4.333)	791	3562
6	−0.39022	(6.246, 5.686, 5.271, 4.908, 4.594, 4.297)	5510	17757
7	−0.39100	(6.353, 5.799, 5.423, 5.096, 4.800, 4.538, 4.292)	20200	47523
8	−0.39156	(6.449, 5.886, 5.550, 5.245 4.958, 4.721, 4.498, 4.295)	107000	182508

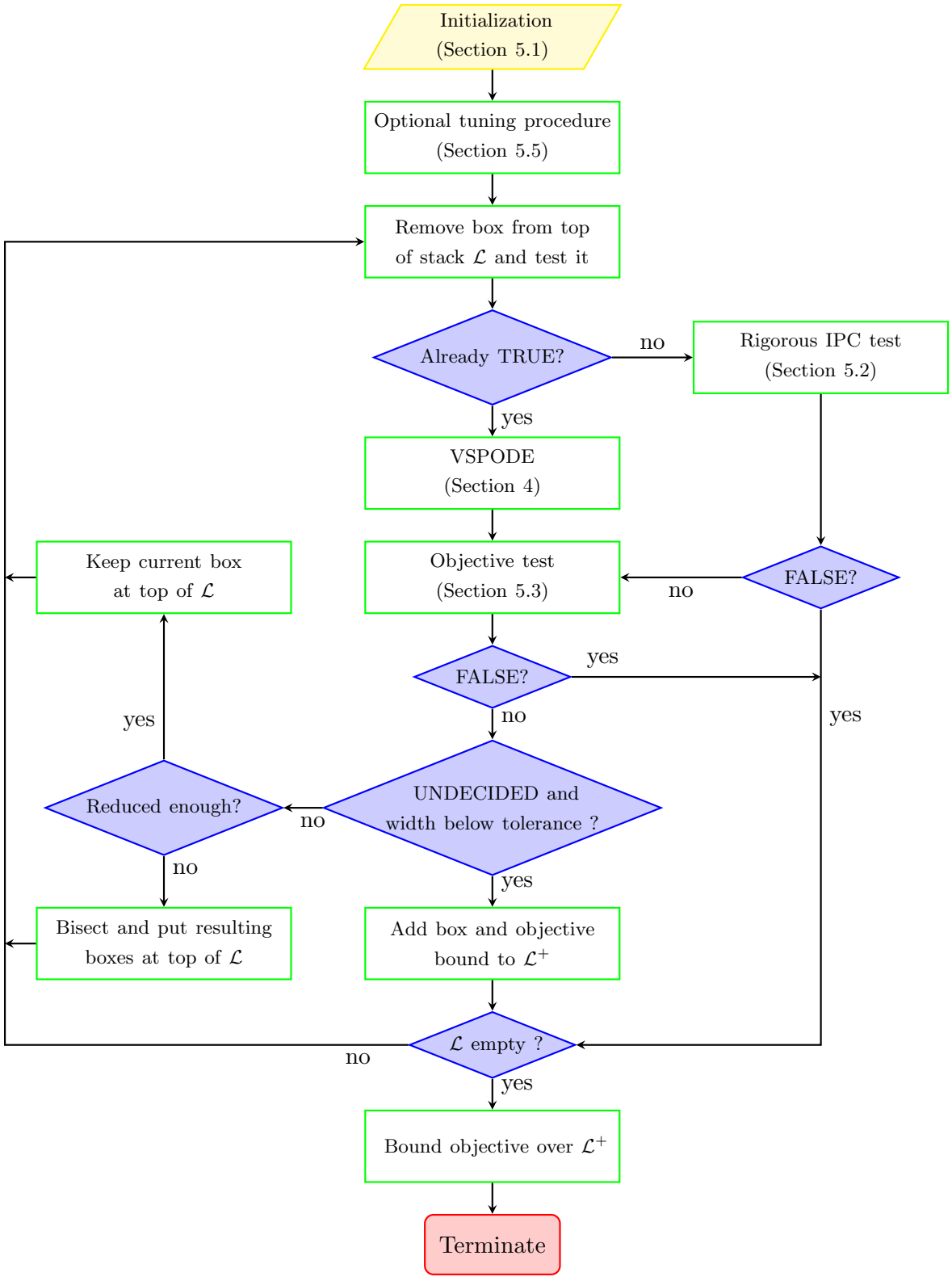

Figure 1: Overall algorithm flowchart. For clarity, step 4(a) (section 5.4), which occurs only in cases of verification failure in VSPODE, is omitted.