J Glob Optim (2009) 45:63–93
DOI 10.1007/s10898-008-9395-6

Towards global bilevel dynamic optimization

Alexander Mitsos · Benoît Chachuat · Paul I. Barton

Received: 10 December 2008 / Accepted: 17 December 2008 / Published online: 6 January 2009

Abstract The global solution of bilevel dynamic optimization problems is discussed. An overview of a deterministic algorithm for bilevel programs with nonconvex functions participating is given, followed by a summary of deterministic algorithms for the global solution of optimization problems with nonlinear ordinary differential equations embedded. Improved formulations for scenario-integrated optimization are proposed as bilevel dynamic optimization problems. Solution procedures for some of the problems are given, while for others open challenges are discussed. Illustrative examples are given.

Keywords Bilevel program · Dynamic optimization · Nonconvex optimization · Global optimization · Scenario-integrated optimization

1 Introduction

After several decades of intensive research, deterministic algorithms for the global solution of nonconvex nonlinear programs (NLPs) and mixed-integer nonlinear programs (MINLPs) have reached a level of maturity and commercial software implementations exist [47]. For a recent review of advances and challenges the reader is referred to [19]. In this article optimization problems which have differential equations and/or additional optimization problems embedded are considered. The former are termed dynamic optimization problems and the

A. Mitsos · P. I. Barton (✉)
Department of Chemical Engineering, Massachusetts Institute of Technology, 66-464,
77 Massachusetts Avenue, Cambridge, MA 02139, USA
e-mail: pib@mit.edu

A. Mitsos
e-mail: amitsos@alum.mit.edu

B. Chachuat
Department of Chemical Engineering, McMaster University, Hamilton, ON, Canada
e-mail: bchachua@alum.mit.edu

latter bilevel programs. The focus is on the combination of these to consider bilevel dynamic optimization problems, in particular in the presence of nonconvexity.

This article considers deterministic global optimization methods, which guarantee obtaining a global solution by furnishing a certificate of optimality. Establishing such a certificate has several advantages. Besides to often obtaining much better solutions than through local methods, the optimality guarantees allow the invalidity of models to be established during parameter estimation [46]. An additional motivation for bilevel and semi-infinite programs is that the global solution of the lower-level program is required just to establish feasibility of a given point. In Sect. 4 global optimality is required to design a safe operation with a failure mode.

In Sect. 2 a recent algorithm [34] for the global solution of bilevel programs with nonconvex lower-level programs is summarized. Section 3 gives an overview of several recent contributions to the global solution of dynamic optimization problems. Then in Sect. 4 scenario-integrated dynamic optimization is discussed. Formulations by Abel and Marquardt [1] are analyzed and extended, and conceptual solution methods are proposed. Note that in the scenario-integrated formulations a global solution of the lower-level program is required, for this gives the constraints for the overall program and a local solution would result in a relaxation of the feasible set, or potentially unsafe operation.

2 Bilevel nonlinear programs

Bilevel programs are programs where an *upper-level program* (or *outer program*, superscript u) is constrained by an embedded *lower-level program* (or *inner program*, superscript 1). In this section regular nonlinear bilevel programs (without dynamics embedded) with (Euclidean) compact host sets P^u, P^l and continuous functions are considered:

$$
\begin{aligned}
f^{u,*} = \min_{\mathbf{p}^u,\mathbf{p}^l} \; & f^u(\mathbf{p}^u,\mathbf{p}^l) \\
\text{s.t. } & \mathbf{g}^u(\mathbf{p}^u,\mathbf{p}^l) \le \mathbf{0} \\
& \mathbf{p}^l \in \arg\min_{\mathbf{p}^m} f^l(\mathbf{p}^u,\mathbf{p}^m) \\
& \qquad \text{s.t. } \mathbf{g}^{l,1}(\mathbf{p}^u,\mathbf{p}^m) \le \mathbf{0} \\
& \qquad\qquad \mathbf{g}^{l,2}(\mathbf{p}^m) \le \mathbf{0} \\
& \mathbf{p}^u \in P^u \subset \mathbb{R}^{n_u}, \quad \mathbf{p}^l,\mathbf{p}^m \in P^l \subset \mathbb{R}^{n_l}.
\end{aligned}
\tag{1}
$$

Unlike the majority of contributions in the literature, here neither uniqueness nor convexity are assumed. Typically in bilevel programs the symbols $\mathbf{x}$ and $\mathbf{y}$ are used for the optimization variables. Here the symbols $\mathbf{p}$ are chosen in accordance with the usual notation in dynamic optimization problems where $\mathbf{x}$ stand for the state variables and $\mathbf{p}$ for the parameters. Note the use of dummy variables for the lower level problem ($\mathbf{p}^m$ instead of $\mathbf{p}^l$).

Nonuniqueness of the minimum of the upper-level program can be handled similarly to single-level nonlinear programs (NLPs), but nonuniqueness in the lower-level program requires special attention. In the case that for a given $\mathbf{p}^u$ the lower-level program has multiple minima $\mathbf{p}^l$, we allow the optimizer of the upper-level program to choose among them. This is the so-called co-operative (or optimistic) formulation [13]. An alternative is to use the so-called pessimistic formulation [34].

Similarly, nonconvexity in the upper-level program poses no essential difficulty compared to NLPs. On the other hand, nonconvexity in the lower-level program is a major complication. Just to establish the feasibility of a given pair $(\bar{\mathbf{p}}^u, \bar{\mathbf{p}}^l)$ the only known way is to solve the lower-level program to global optimality. Since in general only ε-optimality can be guaranteed finitely, one has to resort to ε-feasibility; for a formal definition see Sect. 2.1.

To the authors' best knowledge, the first valid algorithms to solve bilevel programs to guaranteed global optimality when nonconvexity is present in the lower-level program were proposed in [33,34]. Here an overview of the simplest variant is given which does not consider branching nor tightening of the lower-bounding problem by the KKT necessary conditions. The algorithm is similar to the procedure by Blankenship and Falk [10] for semi-infinite programs (SIPs). By adding constraints (cuts) the lower-bounding problems become successively tighter, until the upper-bounding problem is guaranteed to generate a feasible point. In SIPs this is relatively easy, but in general bilevel programs significantly more difficult. The proposal in [33,34] relies on the generation of parametric upper bounds to the lower-level problem.

Among the vast literature on bilevel programs, see e.g., [4,13,42,53] for reviews, the most relevant contributions are briefly mentioned here. Bard [3] considered a simpler formulation without upper-level constraints and with a unique minimum for the lower-level problem. Floudas and coworkers [18,21] applied concepts of global optimization to bilevel programs with convex lower-level programs. Pistikopoulos and coworkers [15,41] considered bilevel programs with linear and quadratic functions and used a parametric optimization approach. Tuy et al. [52] proposed an algorithm for bilevel programs satisfying a monotonicity assumption. Algorithms that guarantee convergence to the global solution or stationary points have been proposed for related programs under nonconvexity, such as min–max programs [16,57], semi-infinite programs (SIP) [9,10,17,35], and generalized semi-infinite programs (GSIP) [28].

2.1 Definitions

Definition 1 (*Lower-Level Program*) For a fixed $\mathbf{p}^u$ we denote:

$$\begin{aligned} \min_{\mathbf{p}^l}\ & f^l(\mathbf{p}^u, \mathbf{p}^l) \\ \text{s.t. } & \mathbf{g}^{l,1}(\mathbf{p}^u, \mathbf{p}^l) \le \mathbf{0} \\ & \mathbf{g}^{l,2}(\mathbf{p}^l) \le \mathbf{0} \\ & \mathbf{p}^l \in P^l, \end{aligned} \tag{2}$$

the *lower-level program*.

Definition 2 (*Parametric Optimal Solution Function*) The parametric optimal solution value of (2) as a function of the upper-level variables is denoted $\bar{f}^l(\mathbf{p}^u)$. For infeasible lower-level programs the convention $\bar{f}^l(\mathbf{p}^u) = +\infty$ is used.

Note that depending on the value of $\mathbf{p}^u$, the set of optimal points of the lower-level problem can be empty, a singleton, a finite set or even an uncountable set.

In NLPs it is common practice to consider a point feasible if it satisfies the equality constraints within a prescribed tolerance. Similarly, for bilevel programs (1) with nonconvex lower-level programs it is only plausible to expect a finitely terminating algorithm to provide a guarantee for ε-feasibility [33]:

Definition 3 (ε-*Feasibility*) A pair $(\bar{\mathbf{p}}^u, \bar{\mathbf{p}}^l)$ is called ε-feasible if it satisfies the constraints of the lower- and upper-level programs and ε_{f^l}-optimality in the lower-level program, i.e.:

$$\begin{aligned}
\mathbf{g}^u(\bar{\mathbf{p}}^u, \bar{\mathbf{p}}^l) &\leq \mathbf{0} \\
\mathbf{g}^{l,1}(\bar{\mathbf{p}}^u, \bar{\mathbf{p}}^l) &\leq \mathbf{0} \\
\mathbf{g}^{l,2}(\bar{\mathbf{p}}^l) &\leq \mathbf{0} \\
f^l(\bar{\mathbf{p}}^u, \bar{\mathbf{p}}^l) &\leq \bar{f}^l(\bar{\mathbf{p}}^u) + \varepsilon_{f^l}.
\end{aligned}$$

An ε-feasible point is called ε-optimal if it satisfies ε_{f^u}-optimality in the upper-level program, i.e.:

$$f^u(\bar{\mathbf{p}}^u, \bar{\mathbf{p}}^l) \leq f^{u,*} + \varepsilon_{f^u}.$$

Definition 4 ($\mathbf{p}^u$ *Feasible in the Upper-Level Program*) The subset of P^u which is admissible in the upper-level program is denoted:

$$P^u_{\text{upper}} = \{\mathbf{p}^u \in P^u : \exists\, \bar{\mathbf{p}}^l \in P^l : \mathbf{g}^u(\mathbf{p}^u, \bar{\mathbf{p}}^l) \leq \mathbf{0}\}.$$

Definition 5 ($\mathbf{p}^u$ *Feasible in the Lower-Level Program*) The subset of P^u which is admissible in the lower-level program is denoted:

$$P^u_{\text{lower}} = \{\mathbf{p}^u \in P^u : \exists\, \bar{\mathbf{p}}^l \in P^l : \mathbf{g}^{l,1}(\mathbf{p}^u, \bar{\mathbf{p}}^l) \leq \mathbf{0},\ \mathbf{g}^{l,2}(\bar{\mathbf{p}}^l) \leq \mathbf{0}\}.$$

2.2 Algorithmic design and overview

The algorithm described here employs the following reformulation

$$\begin{aligned}
&\min_{\mathbf{p}^u, \mathbf{p}^l} f^u(\mathbf{p}^u, \mathbf{p}^l) \\
&\text{s.t. } \mathbf{g}^u(\mathbf{p}^u, \mathbf{p}^l) \leq \mathbf{0} \\
&\quad \mathbf{g}^{l,1}(\mathbf{p}^u, \mathbf{p}^l) \leq \mathbf{0} \\
&\quad\quad \mathbf{g}^{l,2}(\mathbf{p}^l) \leq \mathbf{0} \\
&\mathbf{p}^u \in P^u \Rightarrow f^l(\mathbf{p}^u, \mathbf{p}^l) \leq \bar{f}^l(\mathbf{p}^u) \\
&\qquad\qquad \mathbf{p}^u \in P^u, \quad \mathbf{p}^l \in P^l.
\end{aligned} \tag{3}$$

This reformulation has also been used by Tuy et al. [50,51] in an algorithm for linear bilevel problems. This reformulation has the advantage that while the lower-level program may have infinitely many minima, it always has a unique optimal objective value. Thus, by using this reformulation multiple global minima in the lower-level program pose no essential complication.

The basic design principle for the algorithm described is to formulate a series of single-level global optimization subproblems. While these subproblems are expensive to solve, there exist deterministic global optimization algorithms to solve them. In general it is not a good practice to embed a computationally expensive procedure inside another one. However, as discussed previously, just to establish the feasibility of a given pair $(\bar{\mathbf{p}}^u, \bar{\mathbf{p}}^l)$ requires the solution of the lower-level program to global optimality and, therefore, a nested approach seems inevitable. Since with current computational capabilities it is only practical to solve small-scale bilevel programs with nonconvex lower-level programs, the nested approach does not seem like a major drawback. On the contrary, it has the advantage that state-of-the art global solvers for single-level NLPs can be employed. Moreover, it seems almost natural to employ a nested algorithm to solve a nested problem. Finally, this approach can be extended relatively easily to more difficult formulations, such as with dynamics embedded or mixed-integer formulations.

The main steps in the algorithm presented are (i) the lower-bounding problem which furnishes a lower bound to the optimal objective value as well as a candidate point $\bar{\mathbf{p}}^u$; (ii) the solution of the lower-level program at $\bar{\mathbf{p}}^u$; (iii) the generation of a Slater point $\mathbf{p}^{l,k}$ and a box $P^{u,k} \subset P^u$; and (iv) probing the feasibility of $\bar{\mathbf{p}}^u$. These steps are described and followed by a formal statement of the algorithm.

2.3 Lower-bounding procedure

To obtain a lower bound the feasible set needs to be relaxed. Similar to single-level optimization the constraints of the upper-level program could be relaxed by one of many well-known convex or affine relaxation techniques, see e.g., [47]. In contrast, a different approach is needed for the special constraint "$\mathbf{p}^l$ is a global minimum of the lower-level program". A possibility is the constraint "$\mathbf{p}^l$ is feasible in the lower-level program" [53], however this does not suffice for convergence of the lower bound and here parametric upper bounds for the optimal solution function of the lower-level program for specific subsets of P^u are included to achieve convergence. It would be extremely computationally expensive to obtain the exact solution of the lower-level problem for the entire set P^u. Instead, here a finite collection of pairs $(P^{u,k}, \mathbf{p}^{l,k})$ is obtained, composed of sets $P^{u,k} \subset P^u$ and points $\mathbf{p}^{l,k} \in P^l$, such that for each $\mathbf{p}^{l,k}$ the lower-level constraints are satisfied for all $\bar{\mathbf{p}}^u \in P^{u,k}$, i.e.,

$$\begin{aligned} \mathbf{g}^{l,1}(\bar{\mathbf{p}}^u, \mathbf{p}^{l,k}) \le \mathbf{0}, &\quad \forall \bar{\mathbf{p}}^u \in P^{u,k} \\ \mathbf{g}^{l,2}(\mathbf{p}^{l,k}) \le \mathbf{0} & \end{aligned} \tag{4}$$

From these constraints and the definition of the optimal solution function $\bar{f}^l$ it follows

$$\bar{f}^l(\bar{\mathbf{p}}^u) \le f^l(\bar{\mathbf{p}}^u, \mathbf{p}^{l,k}), \quad \forall \bar{\mathbf{p}}^u \in P^{u,k}. \tag{5}$$

Consider now a finite index set K and any point $(\bar{\mathbf{p}}^u, \bar{\mathbf{p}}^l) \in P^u \times P^l$ which is feasible in (3). By the feasibility of this point, it directly follows

$$\mathbf{g}^u(\bar{\mathbf{p}}^u, \bar{\mathbf{p}}^l) \le \mathbf{0}, \quad \mathbf{g}^{l,1}(\bar{\mathbf{p}}^u, \bar{\mathbf{p}}^l) \le \mathbf{0}, \quad \mathbf{g}^{l,2}(\bar{\mathbf{p}}^l) \le \mathbf{0}. \tag{6}$$

Furthermore, since $\bar{\mathbf{p}}^l$ is a global minimum of the lower-level program for $\bar{\mathbf{p}}^u$ we have $f^l(\bar{\mathbf{p}}^u, \bar{\mathbf{p}}^l) = \bar{f}^l(\bar{\mathbf{p}}^u)$. Together with (5) it follows

$$f^l(\bar{\mathbf{p}}^u, \bar{\mathbf{p}}^l) = \bar{f}^l(\bar{\mathbf{p}}^u) \le f^l(\bar{\mathbf{p}}^u, \mathbf{p}^{l,k}), \quad \forall k \in K : \bar{\mathbf{p}}^u \in P^{u,k}. \tag{7}$$

By (6) and (7) the point $(\bar{\mathbf{p}}^u, \bar{\mathbf{p}}^l)$ is feasible in the program

$$\begin{aligned} &\min_{\mathbf{p}^u, \mathbf{p}^l} f^u(\mathbf{p}^u, \mathbf{p}^l) \\ &\text{s.t. } \mathbf{g}^u(\mathbf{p}^u, \mathbf{p}^l) \le \mathbf{0} \\ &\quad \mathbf{g}^{l,1}(\mathbf{p}^u, \mathbf{p}^l) \le \mathbf{0} \\ &\quad \mathbf{g}^{l,2}(\mathbf{p}^l) \le \mathbf{0} \\ &\mathbf{p}^u \in P^{u,k} \Rightarrow f^l(\mathbf{p}^u, \mathbf{p}^l) \le f^l(\mathbf{p}^u, \mathbf{p}^{l,k}), \quad \forall k \in K \\ &\quad \mathbf{p}^u \in P^u, \quad \mathbf{p}^l \in P^l. \end{aligned} \tag{8}$$

In other words (8) provides a relaxation of (3). Therefore, a valid lower bound can be obtained from the global solution value of (8). Note that the use of logical constraints is well established, see, e.g., [7,38,55]. A simple implementation is using the big-M method and more elaborate formulations such as the convex hull formulation are possible, see e.g., [20]. It is also possible to avoid the logical constraints by a specialized branching in the upper-level variables [34].

In the remainder of this subsection a three-step procedure is summarized to obtain points $\mathbf{p}^{l,k}$ and sets $P^{u,k}$ that satisfy (4). The first step is to fix the variables $\mathbf{p}^u$ to the values of the optimal solution $\bar{\mathbf{p}}^u$ obtained by the lower-bounding problem (8) and to solve the lower-level problem globally

$$\begin{aligned} f^{l,*} = \min_{\mathbf{p}^l} \; & f^l(\bar{\mathbf{p}}^u, \mathbf{p}^l) \\ \text{s.t. } & \mathbf{g}^{l,1}(\bar{\mathbf{p}}^u, \mathbf{p}^l) \leq \mathbf{0} \\ & \mathbf{g}^{l,2}(\mathbf{p}^l) \leq \mathbf{0} \\ & \mathbf{p}^l \in P^l. \end{aligned} \tag{9}$$

The results of this step are also used for the upper-bounding procedure, see Sect. 2.4. Feasibility of (9) is guaranteed by the solution of (8). Similar to the solution of (8), the final lower bound from the global solver needs to be used for $f^{l,*}$.

The second step is to pick $\varepsilon_{f_2^l} > 0$ and to find a point $\mathbf{p}^{l,k}$ such that $\mathbf{g}^{l,1}(\bar{\mathbf{p}}^u, \mathbf{p}^{l,k}) < \mathbf{0}$, $\mathbf{g}^{l,2}(\mathbf{p}^{l,k}) \leq \mathbf{0}$ and $f^l(\bar{\mathbf{p}}^u, \mathbf{p}^{l,k}) \leq f^{l,*} + \varepsilon_{f_2^l}$. A simple way to do so is to solve the optimization problem

$$\begin{aligned} \min_{\mathbf{p}^l, u} \; & u \\ \text{s.t. } & f^l(\bar{\mathbf{p}}^u, \mathbf{p}^l) \leq f^{l,*} + \varepsilon_{f_2^l} \\ & g_i^{l,1}(\bar{\mathbf{p}}^u, \mathbf{p}^l) \leq u, \quad i = 1, \ldots, n_{g^{l,1}} \\ & \mathbf{g}^{l,2}(\mathbf{p}^l) \leq \mathbf{0} \\ & \mathbf{p}^l \in P^l, \quad u \leq 0. \end{aligned} \tag{10}$$

This problem is feasible by the solution of (9). Provided that condition (12) is satisfied (see the section describing the convergence of the algorithm, Sect. 2.5.1), the optimal solution value of (10) is negative and $\mathbf{p}^{l,k}$ satisfies the required properties. To accelerate convergence, the solution of the lower-level program (9) can be used as an initial guess. Finite convergence of the algorithm is guaranteed for sufficiently small $\varepsilon_{f_2^l}$, see Sect. 2.5.1. If there are multiple solutions to (10), the path followed by the algorithm in finding an optimal solution of the bilevel program may change depending on which solution of (10) is obtained.

The third step is to identify a set $P^{u,k}$, that satisfies (4) and contains $\bar{\mathbf{p}}^u$ in its interior. In the case that $\bar{\mathbf{p}}^u$ is a boundary point of P^u, instead part of the boundary of $P^{u,k}$ coincides with part of the boundary of P^u. A very simple possibility to do this is used by Oluwole et al. [39] in the context of kinetic model reduction. Essentially, successively smaller boxes $P^{u,k}$ are guessed until (4) can be verified via an overestimation through interval analysis.

2.4 Upper-bounding procedure

Currently no method exists that provides valid, convergent upper bounds for bilevel programs with nonconvex lower-level programs without the global solution of the lower-level problem.

Recent publications on (G)SIPs enable calculation of upper bounds without solving a global optimization problem by overestimating the lower-level program via interval analysis [8,9, 28] or convex/affine relaxations [17,35]. This is possible because for (G)SIPs a relaxation of the lower-level program gives a restriction of the overall problem [35]. This relation is not true in general for the bilevel problem (1) because a relaxation of the lower-level program changes the feasible set of the overall program [33]. On the other hand, the relation is in principle true for the reformulation (3). However (3) does not have a Slater point, which is required for finite convergence of the methods based on a restriction of the lower-level program (note that allowing ε-feasible points in essence gives such a Slater point). Therefore, here the upper-bounding procedure is based on probing the feasibility of a candidate solution $\bar{\mathbf{p}}^u$. Given a candidate $\bar{\mathbf{p}}^u$, the first step is to solve the nonconvex lower-level program (9) globally and obtain an optimal solution $\bar{\mathbf{p}}^l$ and an optimal solution value $f^{l,*}$. For an arbitrary point $\bar{\mathbf{p}}^u$, this program may be infeasible, in which case no solution to the bilevel program exists for $\mathbf{p}^u = \bar{\mathbf{p}}^u$ and no upper bound can be obtained. The algorithm only considers candidates generated by the solution of the lower-bounding problem (8) for which the feasibility of (9) is guaranteed. Recall that (9) is used for the lower-bounding problem as well. Given the solution $f^{l,*}$ an augmented upper-level problem is solved for the fixed $\bar{\mathbf{p}}^u$

$$\begin{aligned} \min_{\mathbf{p}^l}\ & f^l(\bar{\mathbf{p}}^u, \mathbf{p}^l) \\ \text{s.t. } & \mathbf{g}^u(\bar{\mathbf{p}}^u, \mathbf{p}^l) \leq \mathbf{0} \\ & \mathbf{g}^{l,1}(\bar{\mathbf{p}}^u, \mathbf{p}^l) \leq \mathbf{0} \\ & \mathbf{g}^{l,2}(\mathbf{p}^l) \leq \mathbf{0} \\ & f^l(\bar{\mathbf{p}}^u, \mathbf{p}^l) \leq f^{l,*} + \varepsilon_{f^l} \\ & LBD \leq f^u(\bar{\mathbf{p}}^u, \mathbf{p}^l) \\ & \mathbf{p}^l \in P^l, \end{aligned} \tag{11}$$

allowing an ε_{f^l}-violation of lower-level program optimality. This step is performed, because, due to potential non-uniqueness of the solutions of the lower-level program, a valid upper bound may be obtained even if the solution to (9) does not satisfy the upper-level constraints. If (11) is infeasible then no solution exists for $\mathbf{p}^u = \bar{\mathbf{p}}^u$; otherwise an upper bound is obtained. The inequality $LBD \leq f^u(\bar{\mathbf{p}}^u, \mathbf{p}^l)$ is added to accelerate convergence of (11) and to alleviate partially the consequences of allowing ε_{f^l}-optimality in the lower-level program. Note that the solution of (11) is only an upper bound in the sense of an ε-feasible point.

2.5 Description of bilevel algorithm

Now a formal statement of the algorithm is given. Input to the algorithm are the optimality tolerances ε_{f^u} (for the upper-level program, defining ε-optimality), ε_{f^l} (for the lower-level program, defining ε-feasibility) and $\varepsilon_{f_2^l}$ (for the lower-level program in the calculation of points $\mathbf{p}^{l,k}$). Finally, the optimality tolerance for the single-level optimizer ε_{NLP} is specified.

Algorithm 1 (Basic Algorithm)

1. (**Initialization**)
   Set $LBD = -\infty$, $UBD = +\infty$, $K = \emptyset$, $k = 1$.

2. (**Lower Bounding**)
   Solve (8) globally.
   IF Feasible **THEN**

   - Set LBD to the optimal objective value (final lower bound).
   - Set $\bar{\mathbf{p}}^u$ equal to the solution point (ε_{NLP}-optimal point).

   ELSE (**Infeasible Problem**)

   - Terminate.

   END
3. (**Termination**)
   IF $LBD \geq UBD - \varepsilon_{f^u}$ **THEN** Terminate.
4. (**Lower-Level Program**)
   Solve NLP (9) globally for $\mathbf{p}^u = \bar{\mathbf{p}}^u$. (Recall that feasibility of this program is guaranteed.)
   Set $f^{l,*}$ equal to the optimal objective value (final lower bound).
5. (**Populate Parametric Upper Bounds to Lower-Level Program**)
   Solve (10). (Recall that feasibility of this program is guaranteed.)

   - Set $\mathbf{p}^{l,k}$ equal to the solution point.
   - Obtain an appropriate set $P^{u,k}$.
   - Insert k in K.
   - Set $k = k + 1$.

6. (**Upper Bounding**)
   Solve NLP (11) for $\mathbf{p}^u = \bar{\mathbf{p}}^u$ with $f^{l,*}$ as the upper bound for $f^l(\bar{\mathbf{p}}^u, \mathbf{p}^l)$ and (if feasible) obtain an ε_{NLP}-optimal point $\bar{\mathbf{p}}^l$.
   IF Feasible and $f^u(\bar{\mathbf{p}}^u, \bar{\mathbf{p}}^l) < UBD$ **THEN** set $UBD = f^u(\bar{\mathbf{p}}^u, \bar{\mathbf{p}}^l)$ and $(\mathbf{p}^{u,*}, \mathbf{p}^{l,*}) = (\bar{\mathbf{p}}^u, \bar{\mathbf{p}}^l)$.
7. (**Loop**)
   IF $LBD \geq UBD - \varepsilon_{f^u}$ **THEN** Terminate **ELSE** Goto step 2.

A direct consequence of the validity of the lower and upper-bounding procedures is that on termination of the algorithm, if $UBD = +\infty$, the instance is infeasible. Otherwise, UBD is an ε_{f^u}-estimate of the optimal solution value ($UBD \leq f^{u,*} + \varepsilon_{f^u}$) and $(\mathbf{p}^{u,*}, \mathbf{p}^{l,*})$ is an ε-optimal point (see Definition 3) at which UBD is attained.

2.5.1 Convergence

In this section the convergence of Algorithm 1 is summarized; for a formal convergence proof the reader is referred to [34]. Note again that no convexity or uniqueness assumptions are made for either the lower- or upper-level programs.

In addition to compact host sets, an assumption for the lower-level program is used for the convergence proof: there exists some $\tilde{\varepsilon}_{f^u} > 0$ such that for each point $\bar{\mathbf{p}}^u \in P^u_{\text{upper}} \cap P^u_{\text{lower}}$ at least one of the following two conditions holds:

1. For any $\varepsilon_{f_l^1} > 0$ there exists a point $\tilde{\mathbf{p}}^l \in P^l$ such that

$$\mathbf{g}^{l,1}(\bar{\mathbf{p}}^u, \tilde{\mathbf{p}}^l) < \mathbf{0}, \quad \mathbf{g}^{l,2}(\tilde{\mathbf{p}}^l) \leq \mathbf{0}, \quad f^l(\bar{\mathbf{p}}^u, \tilde{\mathbf{p}}^l) \leq \bar{f}^l(\bar{\mathbf{p}}^u) + \varepsilon_{f_l^1}. \tag{12}$$

2. The upper-level objective value is $\tilde{\varepsilon}_{f^u}$ worse than the optimal objective value $f^{u,*}$

$$f^u(\bar{\mathbf{p}}^u, \bar{\mathbf{p}}^l) > f^{u,*} + \tilde{\varepsilon}_{f^u}, \quad \forall \bar{\mathbf{p}}^l \in P^l : \mathbf{g}^{l,1}(\bar{\mathbf{p}}^u, \bar{\mathbf{p}}^l) \leq \mathbf{0}, \quad \mathbf{g}^{l,2}(\bar{\mathbf{p}}^l) \leq \mathbf{0}, \quad \mathbf{g}^u(\bar{\mathbf{p}}^u, \bar{\mathbf{p}}^l) \leq \mathbf{0}.$$

Based on this assumption it is easy to show that the optimal objective function $\bar{f}^l : P^u \to \mathbb{R}$ of the lower-level problem is continuous for all $\mathbf{p}^u \in P^u_{\text{lower}} \cap P^u_{\text{upper}}$ satisfying (12), and as a consequence either (1) is infeasible or the minimum of (1) exists. Moreover, based on this assumption it is possible to show that the sets $P^{u,k}$ have a nonempty interior. By construction the lower-bounding problem visits only points $\bar{\mathbf{p}}^u \in P^u_{\text{upper}} \cap P^u_{\text{lower}}$. Therefore by (12), at these points it is possible to construct the parametric upper bounds to the optimal solution value of the lower-level program via the pairs $(P^{u,k}, \mathbf{p}^{l,k})$. The corresponding logical constraints augmented to the lower-bounding problem successively tighten the lower-bounding problem to the extent that it will either become infeasible or furnish a point inside an existing $P^{u,k}$ which is also a ε-optimal point. The tolerances must satisfy

$$\begin{aligned} 0 &< \varepsilon_{NLP} \leq \min\{\varepsilon_{f^u}/2, \varepsilon_{f^l}, \tilde{\varepsilon}_{f^u}\} \\ 0 &< \varepsilon_{f^l_2} < \varepsilon_{f^l} - \varepsilon_{NLP} \end{aligned}$$

for finite convergence.

2.6 Illustrative example

Example 2.1 The bilevel program

$$\begin{aligned} \min_{\mathbf{p}^u, \mathbf{p}^l} \; & p_1^u p_1^l + p_2^u (p_1^l)^2 - p_1^u p_2^u p_3^l \\ \text{s.t. } & 0.1\, p_1^l p_2^l - (p_1^u)^2 \leq 0 \\ & p_2^u (p_1^l)^2 \leq 0 \\ & \mathbf{p}^l \in \arg\min_{\mathbf{p}^m} p_1^u (p_1^m)^2 + p_2^u p_2^m p_3^m \\ & \qquad \text{s.t. } (p_1^m)^2 - p_2^m p_3^m \leq 0 \\ & \qquad (p_2^m)^2 p_3^m - p_1^m p_1^u \leq 0 \\ & \qquad -(p_3^m)^2 + 0.1 \leq 0 \\ & \mathbf{p}^u \in [-1, 1]^2 \qquad \mathbf{p}^l, \mathbf{p}^m \in [-1, 1]^3 \end{aligned}$$

has the best known upper bound -1.

Consider the application of Algorithm 1. At the first iteration for the lower-bounding problem

$$\begin{aligned} \min_{\mathbf{p}^u, \mathbf{p}^l} \; & p_1^u p_1^l + p_2^u (p_1^l)^2 - p_1^u p_2^u p_3^l \\ \text{s.t. } & 0.1\, p_1^l p_2^l - (p_1^u)^2 \leq 0 \\ & p_2^u (p_1^l)^2 \leq 0 \\ & (p_1^l)^2 - p_2^l p_3^l \leq 0 \\ & (p_2^l)^2 p_3^l - p_1^l p_1^u \leq 0 \\ & -(p_3^l)^2 + 0.1 \leq 0 \\ & \mathbf{p}^u \in [-1, 1]^2, \quad \mathbf{p}^l \in [-1, 1]^3 \end{aligned}$$

is solved obtaining $LBD = -3$, $\bar{p}_1^u = 1$, $\bar{p}_2^u = -1$, $\bar{p}_1^l = -1$, $\bar{p}_2^l = -1$, $\bar{p}_3^l = -1$. Then the lower-level program is solved for $\bar{\mathbf{p}}^u = (1, -1)$

$$\begin{aligned} \min_{\mathbf{p}^l}\ & 1(p_1^l)^2 + (-1)p_2^l p_3^l \\ \text{s.t. } & (p_1^l)^2 - p_2^l\, p_3^l \le 0 \\ & (p_2^l)^2\, p_3^l - p_1^l(-1) \le 0 \\ & -(p_3^l)^2 + 0.1 \le 0 \\ & \mathbf{p}^l \in [-1, 1]^3 \end{aligned}$$

obtaining $\bar{p}_1^l = 0$, $\bar{p}_2^l = -1$, $\bar{p}_3^l = -1$. The p^u-dependent lower-level constraint ($g^{l,1}$) is inactive at the optimal solution. Therefore $\mathbf{p}^{l,1} = (0, -1, -1)$ is used for the parametric upper bounds to the lower-level program. Moreover, the entire host set P^u can be used as $P^{u,1}$. The first iteration is concluded by solving the upper-level problem for fixed $\bar{\mathbf{p}}^u = (1, -1)$

$$\begin{aligned} \min_{\mathbf{p}^l}\ & 1p_1^l - 1\,(p_1^l)^2 - 1\,(-1)\,p_3^l \\ \text{s.t. } & 0.1\, p_1^l p_2^l - (1)^2 \le 0 \\ & (-1)(p_1^l)^2 \le 0 \\ & (p_1^l)^2 - p_2^l\, p_3^l \le 0 \\ & (p_2^l)^2\, p_3^l - p_1^l(-1) \le 0 \\ & -(p_3^l)^2 + 0.1 \le 0 \\ & \mathbf{p}^l \in [-1, 1]^3 \end{aligned}$$

obtaining $UBD = -1$, $\bar{p}_1^u = 1$, $\bar{p}_2^u = -1$, $\bar{p}_1^l = 0$, $\bar{p}_2^l = -1$, $\bar{p}_3^l = -1$.

A second iteration is required to confirm optimality. Now the lower-bounding problem contains a parametric upper bound to the lower-level problem

$$\begin{aligned} \min_{\mathbf{p}^u,\mathbf{p}^l}\ & p_1^u p_1^l + p_2^u\,(p_1^l)^2 - p_1^u\, p_2^u\, p_3^l \\ \text{s.t. } & 0.1\, p_1^l p_2^l - (p_1^u)^2 \le 0 \\ & p_2^u (p_1^l)^2 \le 0 \\ & (p_1^l)^2 - p_2^l\, p_3^l \le 0 \\ & (p_2^l)^2\, p_3^l - p_1^l p_1^u \le 0 \\ & -(p_3^l)^2 + 0.1 \le 0 \\ & p_1^u (p_1^l)^2 + p_2^u p_2^l p_3^l \le p_1^u 0 + p_2^u(-1)(-1) \\ & \mathbf{p}^u \in [-1, 1]^2, \quad \mathbf{p}^l \in [-1, 1]^3. \end{aligned}$$

This program gives $LBD = -1$. Since $LBD = UBD$ the algorithm terminates.

The reader is directed to [34] for extensive numerical testing of the algorithm.

3 Global dynamic optimization

Dynamic optimization refers to mathematical programs where the objective and constraint functions depend on the solution of a set of differential equations. In this section, dynamic optimization problems embedding general, nonlinear ODEs are considered:

$$\begin{aligned}
\min_{\mathbf{p}\in P} \quad & \phi_0(\mathbf{x}(t_\mathrm{f};\mathbf{p}),\mathbf{p}) + \int_{t_0}^{t_\mathrm{f}} \psi_0(\mathbf{x}(t;\mathbf{p}),\mathbf{p})dt \\
\text{s.t.} \quad & \phi_k(\mathbf{x}(t_\mathrm{f};\mathbf{p}),\mathbf{p}) + \int_{t_0}^{t_\mathrm{f}} \psi_k(\mathbf{x}(t;\mathbf{p}),\mathbf{p})dt \le 0, \quad k = 1,\ldots,n_c \\
& \dot{\mathbf{x}}(t;\mathbf{p}) = \mathbf{f}(\mathbf{x}(t;\mathbf{p}),\mathbf{p}), \quad \forall t \in (t_0, t_\mathrm{f}\,] \\
& \mathbf{x}(t_0;\mathbf{p}) = \mathbf{h}(\mathbf{p}).
\end{aligned} \tag{13}$$

In this formulation, $t \in [t_0, t_\mathrm{f}]$ stands for the independent variable (e.g., time). $\mathbf{p} \in P$ denote the continuous time-invariant parameters, with P a nonempty compact convex subset of $\mathbb{R}^{n_p} \cdot \mathbf{x}$ are the continuous variables describing the state of the system. Further, let X be a nonempty compact subset of $\mathbb{R}^{n_x}$ such that $\mathbf{x}(t;\mathbf{p}) \in X, \forall(t,\mathbf{p}) \in [t_0, t_\mathrm{f}] \times P$. The mappings $\phi_k, \psi_k : X \times P \to \mathbb{R}, k = 0,\ldots,n_c$ are assumed to be continuous, but allowed to be nonconvex. Similarly, $\mathbf{f} : X \times P \to \mathbb{R}^{n_x}$ is assumed to be Lipschitz-continuous on X and continuous on P, while $\mathbf{h} : P \to \mathbb{R}^{n_x}$ is assumed to be continuous on P.

The foregoing assumptions ensure that a solution to the initial value problem (IVP) in ODEs exists and is unique on $[t_0, t_\mathrm{f}]$, for every $\mathbf{p} \in P$, and that this solution depends continuously on $\mathbf{p}$. A sufficient condition for a solution to exist and be unique on $[t_0, t_\mathrm{f}]$ is indeed that $\mathbf{f}$ be locally Lipschitz for each $\mathbf{p} \in P$ and there exists a compact subset $X \subset \mathbb{R}^{n_x}$ such that $\mathbf{x}(t;\mathbf{p}) \in X, \forall(t,\mathbf{p}) \in [t_0, t_\mathrm{f}] \times P$ [24, Theorem 3.3]. At this point, one can then invoke Weierstrass' theorem to guarantee existence of a minimum to problem (13), provided that the set of feasible points is nonempty (constraints satisfied for some point.)

For notational simplicity, neither the right-hand side of the differential equations nor the integrands in the objective and constraints functions in problem (13) depend explicitly on t, i.e., the problem is autonomous. It should be noted, however, that all the results presented later on in this section extend readily to non-autonomous problems.

Problem (13) is effectively an optimization problem on an Euclidean space. Solution methods for such dynamic optimization problems can be subdivided into two broad categories, known as simultaneous and sequential. In the simultaneous approach [49], the state variables are discretized, usually via orthogonal collocation on finite elements. This transforms (13) into a large-scale NLP problem, with both the parameters $\mathbf{p}$ and the collocation coefficients being the decision variables. In the sequential approach [48], only the parameters $\mathbf{p}$ are the decision variables in a master NLP, and function evaluations are provided to this NLP via numerical solution of the fully-determined IVP in ODEs given by fixing the parameter values. Only this latter approach shall be considered subsequently.

Since dynamic optimization problems are often nonconvex and even the simplest problems may exhibit suboptimal local solutions [31], the development of deterministic global optimization algorithms that can rigorously guarantee optimal performance has been a topic of significant interest in recent years. Early attempts have been made to extend the αBB method [2] for regular NLP problems to dynamic embedded NLPs [12,14,40]. Although several such algorithms have yielded rigorous global optimization approaches, they are very computationally expensive for they require second-order sensitivities or adjoints to be calculated; moreover, they require twice-continuous differentiability for the solutions of the differential equations and do not address the critical issue of non-quasi-monotone differential equations.

Lin and Stadtherr [29] have proposed a branch-and-reduce algorithm that utilizes interval analysis and Taylor models for producing bounds enclosing the solutions of ODE systems with parametric Taylor models. Significant improvements in the tightness of the state bounds can be obtained with this approach, but the approach scales exponentially in the number of parameters. Another drawback of this approach is that it requires that the right-hand sides of the ODEs be several times continuously differentiable with respect to both the state and decision variables for construction of the Taylor models, depending on the order of the Taylor models considered.

Methods based on McCormick's relaxation technique [32] have also been proposed that provide convex relaxations for dynamic optimization problems embedding IVPs in nonlinear ODEs [44,45]. McCormick-based relaxations apply to a much wider class of ODEs, for they only require the right-hand side of the differential equations to be Lipschitz-continuous and factorable. Moreover, these methods were shown to outperform αBB-based relaxations both in terms of tightness and computational expense, and allow treatment of non-quasi-monotone differential equations. Notwithstanding some attractive features offered by other approaches, we shall restrict the discussion to this latter class of relaxations in the remainder of this section.

3.1 Preliminaries

This subsection summarizes a number of definitions and results needed in the lower-bounding procedure for problem (13).

Definition 6 Let P be a nonempty, convex set in $\mathbb{R}^{n_p}$, and $f : P \to \mathbb{R}$. The function $u : P \to \mathbb{R}$ is said to be a *convex underestimator* of f on P if (i) u is convex on P, and (ii) $u(\mathbf{p}) \le f(\mathbf{p})$ for all $\mathbf{p} \in P$. The function $o : P \to \mathbb{R}$ is said to be a *concave overestimator* of f on P if $-o$ is a convex underestimator of $-f$ on P.

Definition 7 Let $P \subset \mathbb{R}^{n_p}$ be a nonempty convex set, and let $u : P \to \mathbb{R}$ be convex on P. The vector $\boldsymbol{\xi}_{\mathbf{p}*} \in \mathbb{R}^{n_p}$ is said to be a *subgradient* of u at $\mathbf{p}^* \in P$ if

$$\mathcal{L}^{-}_{u,\mathbf{p}^*}(\mathbf{p}) := u(\mathbf{p}^*) + \boldsymbol{\xi}^{\mathrm{T}}_{\mathbf{p}*}(\mathbf{p} - \mathbf{p}^*) \le u(\mathbf{p}), \quad \forall \mathbf{p} \in P.$$

That is, $\mathcal{L}^{-}_{u,\mathbf{p}^*}$ is a supporting hyperplane from below of u on P. Analogously, a supporting hyperplane from above at $\mathbf{p}^*$ of a concave function o on P is denoted $\mathcal{L}^{+}_{o,\mathbf{p}^*}$.

It is a well-known result that a convex (or a concave) function admits at least one subgradient at any interior point of its domain of definition (see, [6, Chap. III, Theorem 2.2]).

Singer and Barton [43] showed that a convex underestimator for an integral can be constructed by integrating a relaxation of the integrand, provided this latter relaxation is convex for each $t \in [t_0, t_\mathrm{f}]$.

Theorem 1 ([43]) *Let $P \subset \mathbb{R}^{n_p}$ be a nonempty convex set, and let $F : P \to \mathbb{R}$ be defined by $F(\mathbf{p}) := \int_{t_0}^{t_\mathrm{f}} f(t,\mathbf{p})dt$, where $f : [t_0, t_\mathrm{f}] \times P \to \mathbb{R}$ is a continuous function. Consider a continuous function $u : [t_0, t_\mathrm{f}] \times P \to \mathbb{R}$ such that $u(t,\cdot)$ is a convex underestimator of $f(t,\cdot)$ on P for each $t \in [t_0, t_\mathrm{f}]$. Then, $U := \int_{t_0}^{t_\mathrm{f}} u(t,\cdot)dt$ is a convex underestimator of F on P.*

Theorem 1 allows one to derive a convex underestimator for an integral by integrating a relaxation of the integrand, provided this latter relaxation is convex for each $t \in [t_0, t_\mathrm{f}]$.

Theorem 1 builds upon McCormick's composition result:

Theorem 2 ([32]) *Let $P \subset \mathbb{R}^{n_p}$ be a nonempty convex set. Consider the function $g \circ f$, where $f : P \to \mathbb{R}$ is continuous and let $f(P) \subset [a, b]$. Suppose that a convex underestimator u_f and a concave overestimator o_f of f on P are known. Suppose also that a convex underestimator u_g and a concave overestimator o_g of g on $[a, b]$ are available. Let $z_{\min}$ be a point at which u_g attains its infimum on $[a, b]$, and let $z_{\max}$ be a point at which o_g attains its supremum on $[a, b]$. Then, $u_{g \circ f} := u_g[\mathrm{mid}\{u_f, o_f, z_{\min}\}]$ is a convex underestimator of $g \circ f$ on P, and $o_{g \circ f} := o_g[\mathrm{mid}\{u_f, o_f, z_{\max}\}]$ is a concave overestimator of $g \circ f$ on P, where the* mid *function selects the middle value of three scalars.*

In his seminal paper, McCormick [32] also showed how a convex and a concave relaxation can be obtained for any factorable function from recursive application of his composition technique. A factorable function therein is defined as a finite recursive composition of binary sums, binary products and univariate functions with known convex and concave relaxations. Factorable functions cover a quite general class of functions, the simple subclass being those defined without recursion such as $g_1(f_1(\mathbf{p})) + g_2(f_2(\mathbf{p})) \times g_3(f_3(\mathbf{p}))$.

3.2 Branch-and-bound algorithm

The focus is on the use of branch-and-bound algorithms [23] in order to guarantee location of a global solution of the optimization problem, even when the objective function and/or constraints are nonconvex. A branch-and-bound algorithm begins by constructing a relaxation of the original nonconvex problem. This relaxation is solved to generate a lower bound on the solution of the original problem, and should, in some way, be easier to solve than the original problem. In the current context, the relaxation is a convex optimization problem whose objective function underestimates the nonconvex objective function on P and whose feasible set contains that of the nonconvex problem. This can be achieved by constructing functions that are convex relaxations of the objective function and constraint functions on P, and formulating a convex optimization problem from these relaxed functions. Because every local minimum of a convex optimization problem is a global minimum, a lower bound can be found reliably, e.g., upon application of the sequential approach of dynamic optimization. In general, generating an upper bound on the solution may be a difficult task too, e.g., in SIP problems [9]. In global dynamic optimization, however, an upper bound is generated easily from the value of the nonconvex objective function at any feasible point (e.g., a local minimum found by the sequential approach). If the lower and upper bounds are not within some ε tolerance, a branching heuristic is used to partition the set P (normally an interval derived from physical considerations) into two new subproblems (e.g., bisect on one of the variables). Relaxations are then constructed on these two smaller sets, and lower and upper bounds are computed for these partitions. If the lower bound on a partition is greater than the current best upper bound, the global solution cannot exist in that partition and the partition is excluded from further consideration (*fathoming by value dominance*). Fathoming is also performed when the lower bounding problem is infeasible (*fathoming by infeasibility*). This process of branching, bounding and fathoming continues until the lower bound on all active partitions is within ε of the current best upper bound.

The most difficult step in applying the branch-and-bound algorithm to problem (13) lies in the construction of the lower-bounding problem, a convex optimization problem of the form:

$$\min_{\mathbf{p} \in P} u_{\phi_0}(\mathbf{p}) + \int_{t_0}^{t_f} u_{\psi_0}(t, \mathbf{p}) dt$$

$$\text{s.t. } u_{\phi_k}(\mathbf{p}) + \int_{t_0}^{t_\mathrm{f}} u_{\psi_k}(t, \mathbf{p})dt \le 0, \quad k = 1, \ldots, n_c, \tag{14}$$

where u_{ϕ_k}, $k = 0, \ldots, n_c$, are convex underestimators of $\phi_k(\mathbf{x}(t_\mathrm{f}; \cdot), \cdot)$ on P, and $u_{\psi_k}(t, \cdot)$ are pointwise-in-time convex underestimators of $\psi_k(\mathbf{x}(t; \cdot), \cdot)$ on P. Similar to factorable functions, convex relaxations for the terminal and integrand terms of the objective and constraint functions in (13) can be obtained upon application of McCormick's composition technique recursively. However, for functions with state variables participating, this composition requires:

1. a convex function $\mathbf{u_x}(t; \cdot)$ on P and a concave function $\mathbf{o_x}(t; \cdot)$ on P satisfying $\mathbf{u_x}(t; \mathbf{p}) \le \mathbf{x}(t; \mathbf{p}) \le \mathbf{o_x}(t; \mathbf{p})$ for all $\mathbf{p} \in P$, pointwise for all $t \in [t_0, t_\mathrm{f}]$;
2. a time-varying enclosure $\mathcal{X}(t) := [\mathbf{x}^L(t), \mathbf{x}^U(t)]$ for the solutions of the embedded parametric ODEs on P, at each $t \in [t_0, t_\mathrm{f}]$.

Techniques that provide such relaxations and bounds are presented subsequently in Sects. 3.2.1 and 3.2.2.

To illustrate the construction of convex relaxations for terminal terms, consider the simple composite function $\varphi[x_i(t_\mathrm{f}; \cdot)]$ for some $i \in \{1, \ldots, n_x\}$. Let u_{φ,t_f} be a convex underestimator of φ on $\mathcal{X}_i(t_\mathrm{f})$, and denote $z_{\min}(t_\mathrm{f})$ a point at which u_{φ,t_f} attains its infimum on $\mathcal{X}_i(t_\mathrm{f})$. By Theorem 2,

$$u_{\varphi\circ x_i}(\mathbf{p}) := u_{\varphi,t_\mathrm{f}}\left[\mathrm{mid}\left\{u_{x\,i}(t_\mathrm{f}; \mathbf{p}), o_{x\,i}(t_\mathrm{f}; \mathbf{p}), z_{\min}(t_\mathrm{f})\right\}\right]$$

is a convex underestimator for $\varphi[x_i(t_\mathrm{f}; \cdot)]$ on P. Regarding integral terms, a convex underestimator for an integral is obtained by integrating a pointwise-in-time convex underestimator for the corresponding integrand by Theorem 1. Therefore, a convex relaxation u_Φ for the simple integral term $\Phi(\mathbf{p}) := \int_{t_0}^{t_\mathrm{f}} \varphi[x_i(t; \mathbf{p})]dt$, $i \in \{1, \ldots, n_x\}$, is obtained as

$$u_\Phi(\mathbf{p}) := \int_{t_0}^{t_\mathrm{f}} u_{\varphi,t}\left[\mathrm{mid}\left\{u_{x\,i}(t; \mathbf{p}), o_{x\,i}(t; \mathbf{p}), z_{\min}(t)\right\}\right] dt$$

where $u_{\varphi,t}$ denotes a convex underestimator of φ on $\mathcal{X}_i(t)$ at each $t \in [t_0, t_\mathrm{f}]$.

An important property of the lower-bounding approach described herein is that the relaxation step and the bounding step of state relaxation are independent from each other. In the previous example, for instance, neither $u_{x\,i}(t; \mathbf{p})$ nor $o_{x\,i}(t; \mathbf{p})$ are required to be in the domain of $u_{\varphi,t}$—which is $\mathcal{X}_i(t)$, by virtue of the composition of $u_{\varphi,t}$ with the mid function. To see this, observe first that the following relations hold true by construction:

$$z_{\min}(t) \in \mathcal{X}_i(t), \tag{15}$$

$$u_{x\,i}(t; \mathbf{p}) \le o_{x\,i}(t; \mathbf{p}), \quad \forall \mathbf{p} \in P. \tag{16}$$

Moreover,

$$u_{x\,i}(t; \mathbf{p}) \le x_i^U(t), \quad \forall \mathbf{p} \in P, \tag{17}$$

for if the converse were true, then $\left[u_{x\,i}(t; \mathbf{p}), o_{x\,i}(t; \mathbf{p})\right] \cap \mathcal{X}_i(t) = \emptyset$, which contradicts the fact that the state trajectory $\mathbf{x}(t; \mathbf{p})$ exists and is unique since $\mathbf{x}(t; \mathbf{p}) \in \left[\mathbf{u_x}(t; \mathbf{p}), \mathbf{o_x}(t; \mathbf{p})\right]$ and $\mathbf{x}(t; \mathbf{p}) \in \mathcal{X}(t)$ for all $\mathbf{p} \in P$. Likewise,

$$o_{x\,i}(t; \mathbf{p}) \ge x_i^L(t), \quad \forall \mathbf{p} \in P. \tag{18}$$

By definition of the mid function and from (16), we have either

(i) $u_{xi}(t;\mathbf{p}) \le z_{\min}(t) \le o_{xi}(t;\mathbf{p})$, and from (15), $\text{mid}\,\{u_{xi}(t;\mathbf{p}), o_{xi}(t;\mathbf{p}), z_{\min}(t)\} = z_{\min}(t) \in \mathcal{X}_i(t)$; or

(ii) $z_{\min}(t) \le u_{xi}(t;\mathbf{p}) \le o_{xi}(t;\mathbf{p})$, and from (15) and (17), $\text{mid}\,\{u_{xi}(t;\mathbf{p}), o_{xi}(t;\mathbf{p}), z_{\min}(t)\} = u_{xi}(t;\mathbf{p}) \in \mathcal{X}_i(t)$; or

(iii) $u_{xi}(t;\mathbf{p}) \le o_{xi}(t;\mathbf{p}) \le z_{\min}(t)$, and from (15) and (18), $\text{mid}\,\{u_{xi}(t;\mathbf{p}), o_{xi}(t;\mathbf{p}), z_{\min}(t)\} = o_{xi}(t;\mathbf{p}) \in \mathcal{X}_i(t)$.

Overall, $\text{mid}\,\{u_{xi}(t;\mathbf{p}), o_{xi}(t;\mathbf{p}), z_{\min}(t)\} \in \mathcal{X}_i(t)$ for each $\mathbf{p} \in P$, regardless of the range of the convex and concave relaxations $u_{xi}(t;\mathbf{p}), o_{xi}(t;\mathbf{p})$.

3.2.1 Convex/concave relaxations for states variables

A method for constructing a convex underestimator $\mathbf{u_x}(t;\cdot)$ and a concave overestimator $\mathbf{o_x}(t;\cdot)$ of $\mathbf{x}(t;\cdot)$ on P via solution of a set of linear auxiliary ODEs is discussed in this subsection. It assumes the availability of a time-varying enclosure $\mathcal{X}(t)$ for the state variables.

Theorem 3 ([44]) *Let $P \subset \mathbb{R}^{n_p}$ be a nonempty convex set. Consider the parametric ODEs*

$$\dot{\mathbf{x}}(t;\mathbf{p}) = \mathbf{f}(\mathbf{x}(t;\mathbf{p}), \mathbf{p})$$

for $t \in [t_0, t_f]$ and $\mathbf{p} \in P$, with the initial conditions $\mathbf{x}(t_0;\mathbf{p}) = \mathbf{h}(\mathbf{p})$. Suppose that its solution $\mathbf{x}(t;\mathbf{p})$ is bounded by the set $\mathcal{X}(t) \subset \mathbb{R}^{n_x}$ of time-varying state bounds, at each $t \in [t_0, t_f]$. For each $i = 1, \ldots, n_x$, let $u_{f_i} : \mathcal{X}(t) \times P \to \mathbb{R}$ and $o_{f_i} : \mathcal{X}(t) \times P \to \mathbb{R}$ be a convex underestimator and a concave overestimator for f_i on $\mathcal{X}(t) \times P$, respectively. Let also $u_{hi} : P \to \mathbb{R}$ and $o_{hi} : P \to \mathbb{R}$ be a convex underestimator and a concave overestimator for h_i on P, respectively. Consider the parametric differential equations

$$\dot{u}_{xi}(t;\mathbf{p}) = \inf_{\mathbf{z}}\{\mathcal{L}^{-}_{u_{f_i},(\mathbf{x}^*(t),\mathbf{p}^*)}(\mathbf{z},\mathbf{p}) : \mathbf{u_x}(t;\mathbf{p}) \le \mathbf{z} \le \mathbf{o_x}(t;\mathbf{p}), z_i = u_{xi}(t;\mathbf{p})\} \quad (19)$$

$$\dot{o}_{xi}(t;\mathbf{p}) = \sup_{\mathbf{z}}\{\mathcal{L}^{+}_{o_{f_i},(\mathbf{x}^*(t),\mathbf{p}^*)}(\mathbf{z},\mathbf{p}) : \mathbf{u_x}(t;\mathbf{p}) \le \mathbf{z} \le \mathbf{o_x}(t;\mathbf{p}), z_i = o_{xi}(t;\mathbf{p})\} \quad (20)$$

for each $i = 1, \ldots, n_x$ and $t \in [t_0, t_f]$, with initial conditions

$$u_{xi}(t_0;\mathbf{p}) = \mathcal{L}^{-}_{u_{hi},\mathbf{p}^*}(\mathbf{p}) \quad (21)$$

$$o_{xi}(t_0;\mathbf{p}) = \mathcal{L}^{+}_{o_{hi},\mathbf{p}^*}(\mathbf{p}) \quad (22)$$

for some reference trajectory $(\mathbf{x}^*(t), \mathbf{p}^*)$ in the interior of $\mathcal{X}(t) \times P$. Then, $\mathbf{u_x}(t;\cdot)$ and $\mathbf{o_x}(t;\cdot)$ are respectively a convex underestimator and a concave overestimator of $\mathbf{x}(t;\cdot)$ on P, for each $t \in [t_0, t_f]$.

The most difficult aspect of obtaining state relaxations is constructing the right-hand sides of (19, 20) as well as of (21, 22). A way of obtaining the convex/concave relaxations $\mathbf{u_f}$, $\mathbf{o_f}$, $\mathbf{u_h}$ and $\mathbf{o_h}$ is by applying McCormick's composition technique recursively [32]. Yet, any other relaxation procedure can be used provided it possesses the property that as the parameter set decreases, the relaxations become closer to the original function with monotonic pointwise convergence.

In Theorem 3, a reference trajectory $(\mathbf{x}^*(t), \mathbf{p}^*)$ must be chosen for constructing the auxiliary differential system. Observe that *any* choice of this reference trajectory in the interior of $\mathcal{X}(t) \times P$ gives a valid convex underestimator $\mathbf{u_x}(t;\cdot)$ and a valid concave overestimator $\mathbf{o_x}(t;\cdot)$ of $\mathbf{x}(t;\cdot)$ on P, at each $t \in [t_0, t_f]$. In particular, the trajectories $\mathbf{x}^*(t)$ need not be the solution of the ODEs for $\mathbf{p}^*$ or any other values of the parameters. Moreover, if either of the convex/concave relaxations $\mathbf{u_f}$, $\mathbf{o_f}$, $\mathbf{u_h}$ or $\mathbf{o_h}$ admits several subgradients at a reference point, any of the corresponding hyperplanes supporting that relaxation can be considered.

3.2.2 Implied state bounds

The problem of estimating the image of the parameter set P under the solution of the ODEs is addressed in this subsection. In general, obtaining the exact bounds for nonlinear ODEs (i.e., the tightest possible implied state bounds) is itself a nonconvex optimal control problem. Nevertheless, a number of techniques exist that provide rigorous pointwise-in-time enclosures of the image set.

Interval methods for ODEs provide a natural approach for computing the desired enclosures at a finite number of times $t \in [t_0, t_f]$. For example, the popular VNODE package [37] can be used by treating the parameters as additional state variables with time derivatives equal to zero. Recently, Lin and Stadtherr [30] have also developed a validated solver for parametric ODEs.

Another way of getting state bounds is via the theory of differential inequalities [22,54]. The benefit of using differential inequalities over the aforementioned validated ODE methods lies in the fact that state bounds can be obtained at any time instant, as the solution of an auxiliary set of ODEs. It is well known, however, that the application of standard differential inequalities to non-quasi-monotone ODEs leads to bounds that explode on short time scales. In response to this, Singer and Barton [44] have developed new differential inequality results that incorporate a priori knowledge concerning constraints and solution invariants in the computation of state bounds. Assuming that a set of natural bounds, $\overline{\mathcal{X}}(t, \mathbf{p})$, is known a priori for a system, the following theorem derives tighter state bounds for systems of non-quasi-monotone ODEs.

Theorem 4 ([44]) *Let $P \subset \mathbb{R}^{n_p}$ be a nonempty convex set. Consider the parametric ODEs*

$$\dot{\mathbf{x}}(t; \mathbf{p}) = \mathbf{f}(\mathbf{x}(t; \mathbf{p}), \mathbf{p})$$

for $t \in [t_0, t_f]$ and $\mathbf{p} \in P$, with the initial conditions $\mathbf{x}(t_0) = \mathbf{h}(\mathbf{p})$. For a given $\mathbf{p} \in P$, suppose that the state variables $\mathbf{x}(t; \mathbf{p})$ lie in the set $\overline{\mathcal{X}}(t, \mathbf{p}) \subset \mathbb{R}^{n_x}$, known independently from the solution of the ODEs. Furthermore, let $\overline{\mathcal{X}}(t)$ be defined pointwise in time by

$$\overline{\mathcal{X}}(t) := [\bar{\mathbf{x}}^L(t), \bar{\mathbf{x}}^U(t)] \text{ such that } \bar{x}_i^L(t) := \inf_{\mathbf{q} \in P} \overline{\mathcal{X}}_i(t, \mathbf{q}),$$

$$\bar{x}_i^U(t) := \sup_{\mathbf{q} \in P} \overline{\mathcal{X}}_i(t, \mathbf{q}), \quad \forall i = 1, \dots, n_x.$$

Then, any trajectories $\mathbf{x}^L(t)$ and $\mathbf{x}^U(t)$ satisfying the differential inequalities

$$\dot{x}_i^L(t) \le \inf_{\mathbf{z},\mathbf{q}} \{f_i(\mathbf{z}, \mathbf{q}) : \mathbf{z} \in \mathcal{X}(t) \cap \overline{\mathcal{X}}(t), z_i = x_i^L(t), \mathbf{q} \in P\} \tag{23}$$

$$\dot{x}_i^U(t) \ge \sup_{\mathbf{z},\mathbf{q}} \{f_i(\mathbf{z}, \mathbf{q}) : \mathbf{z} \in \mathcal{X}(t) \cap \overline{\mathcal{X}}(t), z_i = x_i^U(t), \mathbf{q} \in P\} \tag{24}$$

for each $i = 1, \dots, n_x$ and $t \in [t_0, t_f]$, along with the initial conditions

$$\mathbf{x}^L(t_0) \le \mathbf{h}(\mathbf{p}) \le \mathbf{x}^U(t_0), \quad \forall \mathbf{p} \in P,$$

are such that

$$\mathbf{x}^L(t) \le \mathbf{x}(t; \mathbf{p}) \le \mathbf{x}^U(t), \quad \forall \mathbf{p} \in P,$$

for each $t \in [t_0, t_f]$.

The most difficult aspect of obtaining implied state bounds appears to be bounding the solutions of the optimization problems defining the right-hand sides of (23, 24). Solving these optimization problems at each integration step in a numerical integration would be a prohibitively expensive task. Instead, their solutions can be estimated by *interval arithmetic* pointwise in time, e.g., via *natural interval extension* [36]. Given the inclusion monotonicity property of natural interval extensions, theoretical guarantees can be given that the state bounds approach the corresponding state trajectory when the parameter set approaches degeneracy. This property is indeed necessary to guarantee finite ε-convergence of the branch-and-bound algorithm, as explained in the following subsection.

3.2.3 Algorithm summary and convergence

A formal statement of the branch-and-bound algorithm is given in this subsection. Input to the algorithm is the optimality tolerance $\varepsilon_{BB} > 0$.

Algorithm 2 (Branch-and-Bound Algorithm)

1. (**Initialization**)
   Set $UBD = +\infty$, $\mathcal{S} = \{P\}$.
2. (**Lower Bounding**)
   Select and remove $N \in \mathcal{S}$.
   Solve (14) for $\mathbf{p} \in N$ and (if feasible) set $LB(N)$ to the optimal objective value.
   IF Infeasible or $LB(N) \geq UBD - \varepsilon_{BB}$ **THEN** (**Fathoming**)
     Goto step 5.
   END
3. (**Upper Bounding**)
   Solve (13) (locally) for $\mathbf{p} \in N$ and (if feasible) set $UB(N)$ to the optimal objective value and $\mathbf{p}(N)$ to a local optimal point.
   IF Feasible and $UB(N) < UBD$ **THEN** Set $UBD = UB(N)$ and $\mathbf{p}^* = \mathbf{p}(N)$.
4. (**Branching**)
   IF $LB(N) < UBD - \varepsilon_{BB}$ **THEN**
     Bisect the set N into two nonempty subsets N^L and N^R.
     Insert N^L and N^R in $\mathcal{S}$.
   END
5. (**Loop**)
   IF $\mathcal{S} = \emptyset$ **THEN** Terminate **ELSE** Goto step 2.

If $UBD = +\infty$ upon termination of the algorithm, the instance is infeasible. Otherwise, UBD is an ε_{BB} estimate of the optimal solution value and $\mathbf{p}^*$ is a feasible point at which UBD is attained.

A sufficient condition for ε_{BB}-convergence to the global solution in finite time of the branch-and-bound Algorithm 2 is that (i) the selection operation be bound improving, and (ii) the bounding operation be consistent [23, Theorem IV.3]. A bound improving selection operation refers to the property that, after a finite number of steps, at least one partition element where the actual lower bound is attained is selected for further partitioning. Provided that partitions on which the global solution value is attained are retained and exhaustive search is utilized (e.g., by using bisection for branching), and by the finite dimensionality of the decision variable space, Algorithm 2 can be shown to satisfy this condition. On the other hand, a consistent bounding operation refers to the property that, at every step, any unfathomed partition can be further refined and, in any infinite sequence of nested partitions, the lower bound converges to the upper bound. This condition can be shown to be satisfied by Algorithm 2 upon utilizing Theorems 3 and 4 together with Theorem 2 for relaxation of factorable functions. In particular, generating state bounds via Theorem 4 guarantees that $\mathcal{X}(t)$ become degenerate as P itself becomes degenerate. Moreover, the convex/concave state relaxations generated via Theorem 3 can also be shown to converge to the original function as P becomes degenerate when the convex/concave relaxations $\mathbf{u_f}$, $\mathbf{o_f}$, $\mathbf{u_h}$ and $\mathbf{o_h}$ themselves possess a consistent bounding operation (which is the case when McCormick relaxations are considered). The reader is referred to [45] for a formal convergence proof and related discussions.

3.3 Extension to optimal control and multistage problems

The results presented earlier in this section can be readily extended to address problems wherein the objective and constraint functions include a finite number of point or integral terms on a fixed partition of the time domain. Extensions also exist that encompass solution of problems embedding multistage ODEs, possibly with varying transition times, including varying initial and/or terminal times [25]. These extensions are illustrated later on in this subsection through the case study of a simple two-stage dynamic system with fixed intermediate time and varying terminal time.

Finally, problems containing function valued decision variables $\mathbf{u}(t)$, such as in optimal control, can be converted in the formulation given in (13) upon application of control vector parameterization (CVP) techniques [11,48]. CVP approximates $\mathbf{u}(t)$ with parametric functions $\boldsymbol{\omega}(t, \mathbf{p})$, such as piecewise constant or piecewise linear functions. As such, it yields a suboptimal solution to the original control problem, although it is well known that even coarse parameterizations usually give solution values close to the optimal solution value. Often, optimal control problems also contain inequality path constraints of the form

$$\eta(\mathbf{x}(t), \mathbf{u}(t)) \leq 0,$$

which must be satisfied at every $t \in [t_0, t_f]$. Two popular approximate methods for handling path constraints are [48]: (i) discretization as interior-point constraints,

$$\eta(\mathbf{x}(t_q), \boldsymbol{\omega}(t_q, \mathbf{p})) \leq 0,$$

where $t_q \in [t_0, t_f]$ are a finite set of points; and (ii) transcription as an integral constraint,

$$\int_{t_0}^{t_f} \max\{0, \eta(\mathbf{x}(t), \boldsymbol{\omega}(t, \mathbf{p}))\}^2 dt = 0. \tag{25}$$

Observe that the isoperimetric constraint (25) is not regular because the gradient of that integral is equal to zero whenever this constraint is satisfied; in response to this, (25) is usually relaxed as an inequality constraint,

$$\int_{t_0}^{t_f} \max\{0, \eta(\mathbf{x}(t), \boldsymbol{\omega}(t, \mathbf{p}))\}^2 dt \leq \epsilon,$$

where ϵ is a small nonnegative constant. Using either or both of the foregoing approaches for dealing with path constraints in conjunction with CVP, path constrained optimal control problems thus conform to the problem formulation given in (13).

3.3.1 Illustrative example

Consider the two-stage dynamic system

$$\dot{x}(t) = -0.1\,[x(t) - u(t)]^2\,, \quad \forall t \in (0, 1]$$
$$\dot{x}(t) = x(t), \quad \forall t \in (1, T]$$

with initial condition $x(0) = u(0)^2 - 0.5$ and state continuity at time $t = 1$. The objective here is to maximize the value of $x(T)$ by varying the control $u(t) \in [-3, 2], 0 \leq t \leq 1$ as well as the terminal time $T \in [1, 2]$.

For simplicity, the control is parameterized by a single constant parameter, $u(t) \equiv \omega, 0 \leq t \leq 1$. Moreover, the dynamics of the system in the second stage are scaled by $(T - 1)$, so that the differential system in each stage is now defined on a fixed partition interior to the time domain [0,2]. With these reformulations, the following (nonconvex) multistage dynamic optimization problem is obtained:

$$\min_{\omega,T} \quad \phi_0[x(2)] := -x(2) \tag{26}$$
$$\text{s.t.} \quad \dot{x}(\tau) = -0.1\,[x(\tau) - \omega]^2\,, \quad \forall \tau \in (0, 1] \tag{27}$$
$$\dot{x}(\tau) = [T - 1]\,x(\tau), \quad \forall \tau \in (1, 2] \tag{28}$$
$$x(0) = \omega^2 - 0.5 \tag{29}$$
$$\omega \in [-3, 2], \quad T \in [1, 2]. \tag{30}$$

In order to solve this problem using branch-and-bound, one needs to construct lower-bounding problems on arbitrary subpartitions $[\omega^L, \omega^U] \times [T^L, T^U]$ of the parameter domain $P := [-3, 2] \times [1, 2]$. Time-varying enclosures $\mathcal{X}(\tau) = [x^L(\tau), x^U(\tau)]$ are easily obtained from Theorem 4 as the solutions of the following auxiliary, two-stage ODE system:

$$\left.\begin{aligned} \dot{x}^L(\tau) &= -0.1 \max\left\{(x^L(\tau) - \omega^L)^2, (x^L(\tau) - \omega^U)^2\right\} \\ \dot{x}^U(\tau) &= -0.1 \operatorname{mid}\left\{x^U(\tau) - \omega^U, x^U(\tau) - \omega^L, 0\right\}^2 \end{aligned}\right\} \forall \tau \in (0, 1] \tag{31}$$

$$\left.\begin{aligned} \dot{x}^L(\tau) &= \min\big\{[T^L - 1]x^L(\tau), [T^U - 1]x^L(\tau), [T^L - 1]x^U(\tau), [T^U - 1]x^U(\tau)\big\} \\ \dot{x}^U(\tau) &= \max\big\{[T^L - 1]x^L(\tau), [T^U - 1]x^L(\tau), [T^L - 1]x^U(\tau), [T^U - 1]x^U(\tau)\big\} \end{aligned}\right\} \forall \tau \in (1, 2] \tag{32}$$

$$x^L(0) = \operatorname{mid}\left\{\omega^L, \omega^U, 0\right\}^2 - 0.5$$
$$x^U(0) = \max\left\{(\omega^L)^2, (\omega^U)^2\right\} - 0.5. \tag{33}$$

Next, using McCormick's relaxation technique to obtain convex/concave relaxations for the right-hand side of the ODEs and their initial conditions, state relaxations $u_x(\tau), o_x(\tau)$ are obtained from Theorem 3 as the solutions of the auxiliary, two-stage ODE system:

$$\left.\begin{aligned} \dot{u}_x(\tau) = {} & -0.1(u_x(\tau) - \omega)(x^U(\tau) + x^L(\tau) - \omega^U - \omega^L) \\ & + 0.1(x^U(\tau) - \omega^L)(x^L(\tau) - \omega^U) \\ \dot{o}_x(\tau) = {} & -0.2(o_x(\tau) - \omega)(x^\star(\tau) - \omega^\star) - 0.1(x^\star(\tau) - \omega^\star)^2 \end{aligned}\right\} \forall \tau \in (0, 1] \tag{34}$$

$$\left.\begin{aligned}\dot{u}_x(\tau) = \max\big\{&[T - T^L]x^L(\tau) + [T^L - 1]\,\mathrm{mid}\{u_x(\tau), o_x(\tau), z_u^L(\tau)\},\\ &[T - T^U]x^U(\tau) + [T^U - 1]\,\mathrm{mid}\{u_x(\tau), o_x(\tau), z_u^U(\tau)\}\big\}\\ \dot{o}_x(\tau) = \min\big\{&[T - T^L]x^U(\tau) + [T^L - 1]\,\mathrm{mid}\{u_x(\tau), o_x(\tau), z_o^L(\tau)\},\\ &[T - T^U]x^L(\tau) + [T^U - 1]\,\mathrm{mid}\{u_x(\tau), o_x(\tau), z_o^U(\tau)\}\big\}\end{aligned}\right\}\ \forall\tau \in (1, 2] \tag{35}$$

$$\begin{aligned}u_x(0) &= 2\,\omega\,\omega^\star + (\omega^\star)^2 - 0.5\\ o_x(0) &= \omega(\omega^L + \omega^U) - \omega^L\omega^U - 0.5,\end{aligned} \tag{36}$$

for any reference trajectory $(x^\star(\tau), \omega^\star, T^\star) \in \mathcal{X}(\tau) \times [\omega^L, \omega^U] \times [T^L, T^U]$, and

$$z_u^L(\tau) = \begin{cases} x^L(\tau) & \text{if } T^L \geq 1\\ x^U(\tau) & \text{otherwise,}\end{cases} \quad z_u^U(\tau) = \begin{cases} x^L(\tau) & \text{if } T^U \geq 1\\ x^U(\tau) & \text{otherwise,}\end{cases}$$

$$z_o^L(\tau) = \begin{cases} x^U(\tau) & \text{if } T^L \geq 1\\ x^L(\tau) & \text{otherwise,}\end{cases} \quad z_o^U(\tau) = \begin{cases} x^U(\tau) & \text{if } T^U \geq 1\\ x^L(\tau) & \text{otherwise.}\end{cases}$$

Finally, a convex underestimator for the objective function $\phi_0[x(2)]$ is easily obtained as:

$$u_{\phi_0 \circ x} = -\,\mathrm{mid}\{u_x(2), o_x(2), x^U(2)\}.$$

A representation of the nonconvex objective function and its convex underestimator is shown in Fig. 1. Here, the parameters are allowed to vary in the entire P set, and the reference trajectory is chosen as $(x^\star(\tau), \omega^\star, T^\star) = (\frac{x^L(\tau)+x^U(\tau)}{2}, \frac{\omega^L+\omega^U}{2}, \frac{T^L+T^U}{2})$.

The results obtained on application of the branch-and-bound algorithm presented in Sect. 3.2.3 are summarized in Table 1. At each node N, a lower bound, $LB(N)$, and (if needed) an upper bound, $UB(N)$, are generated as:

$$\begin{aligned} UB(N) := \min\nolimits_{\omega,T}\ & -x(2) \\ \text{s.t.}\ \ & \text{Eqs. (27–29)} \\ & (\omega, T) \in N, \end{aligned} \qquad \begin{aligned} LB(N) := \min\nolimits_{\omega,T}\ & -\mathrm{mid}\{u_x(2), o_x(2), x^U(2)\} \\ \text{s.t.}\ \ & \text{Eqs. (31–36)} \\ & (\omega, T) \in N. \end{aligned}$$

These dynamic optimization subproblems are solved by using the sequential approach. At each level, the node N having the lowest lower-bounding value $LB(N)$ is selected best-bound search heuristic. Branching via bisection on one the variables is employed, with variable selection based on the least-reduced axis rule. No domain reduction heuristic is used, and the branch-and-bound tolerance is set to $\varepsilon_{BB} = 10^{-3}$. The algorithm requires seven iterations for termination, and the global solution $\phi^* = -8.9821$ is found at node 3.

Fig. 1 Nonconvex objective function and convex underestimator versus parameter $\omega \in [-3, 2]$ and terminal time $T \in [1, 2]$

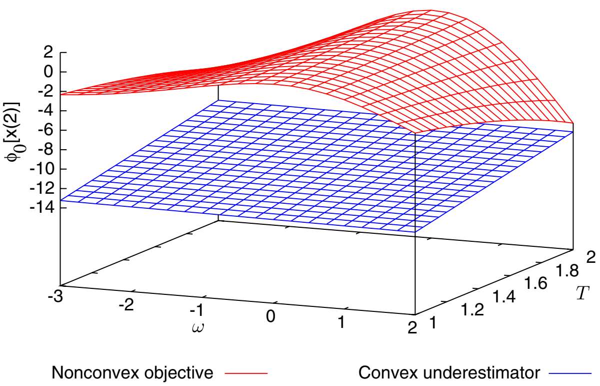

Table 1 Branch-and-bound iterations for problem (26–30)

Node	N	UBD	$LB(N)$	$UB(N)$	Action	$\mathcal{S}$
1	$[-3, 2] \times [1, 2]$	$+\infty$	-13.5069	-6.3848	Update UBD and branch	$\{[-3, -0.5] \times [1, 2], [-0.5, 2] \times [1, 2]\}$
2	$[-3, -0.5] \times [1, 2]$	-6.3848	-10.5178	-6.3848	Branch	$\{[-3, -0.5] \times [1, 1.5], [-3, -0.5] \times [1.5, 2], [-0.5, 2] \times [1, 2]\}$
3	$[-0.5, 2] \times [1, 2]$	-6.3848	-8.9822	-8.9821	Update UBD and fathom by value dominance	$\{[-3, -0.5] \times [1, 1.5], [-3, -0.5] \times [1.5, 2]\}$
4	$[-3, -0.5] \times [1, 1.5]$	-8.9821	-5.6843	–	Fathom by value dominance	$\{[-3, -0.5] \times [1.5, 2]\}$
5	$[-3, -0.5] \times [1.5, 2]$	-8.9821	-9.3718	-6.3848	Branch	$\{[-3, -1.75] \times [1.5, 2], [1.75, -0.5] \times [1.5, 2]\}$
6	$[-3, -1.75] \times [1.5, 2]$	-8.9821	-6.9487	–	Fathom by value dominance	$\{[-1.75, -0.5] \times [1.5, 2]\}$
7	$[-1.75, -0.5] \times [1.5, 2]$	-8.9821	-4.5737	–	Fathom by value dominance	$\emptyset$

4 Scenario-integrated dynamic optimization

This section discusses how to extend and combine the bilevel and dynamic optimization algorithms presented in the previous sections to scenario-integrated dynamic optimization. The goal of the formulations in this section is to account for uncertainty in the operation of dynamic systems. In particular, events, outside of the control of the optimizer, can occur that change the dynamic behavior or the constraints of the system and can lead to loss of profit or to catastrophic failures. For instance, external economic factors, such as drastic price changes may render the operation suboptimal. A more extreme case is at the operation of a chemical plant where external factors such as the sudden occurrence of rain may lead to plant failure. The main objective is to optimize the *nominal operation* (i.e., the operation without the occurrence of the external events), but in addition, the *scenario operation* (i.e., the operation once the event has set in), must be feasible and/or optimal. A major complication is that the time of event occurrence is in general unknown and this gives semi-infinite formulations, as in robust optimization [5]. To ensure feasibility, the global solution of the lower-level programs is mandated. An additional complication is that the scenario operation may be governed by objectives different from the nominal operation.

The formulations proposed here are inspired by Abel and Marquardt [1]. The notation is changed for consistency with the previous sections. The superscripts u and l denote the upper and lower-level programs respectively. The superscripts c, s and f denote respectively intermediate, transition and final time. The dependence of the state variables $\mathbf{x}$ on the optimization parameters is omitted to simplify the notation. Time derivatives of the state variables $\mathbf{x}$ are denoted with $\dot{\mathbf{x}}$.

It is assumed that the objective depends only on the state variables at the final time. Integral objective functions can be treated as described in Sect. 3.2 and are omitted for simplicity. Furthermore, path constraints are excluded to keep notation to a minimum. As described in Sect. 3 path constraints can be approximated as interior-point constraints.

Throughout this section a single scenario is considered. The extension to multiple failure scenarios would not add conceptual difficulty, but would significantly complicate the notation. On the other hand, the extension to nested scenarios is a major complication, both conceptually and computationally.

It is assumed that the failure can occur at any point during the nominal operation, irrespectively of the values of the state variables, desicion variables, or time itself. Depending on the application this may not be an appropriate assumption; for instance if the source of failure are weather phenomena, it is known that they can occur only at certain periods of the year. This assumption is however made because the extension to state-variable dependent failure time requires significant additional considerations.

For problems with variable final time, the reformulation discussed in Sect. 3.3 is performed. This scaling of the dynamics makes notation simpler and the dynamic optimization problems more tractable. Moreover, it transforms GSIP into SIP, compare also [27]. Without loss of generality the initial time is assumed to be zero and the time is scaled to [0,1]. The explicit time-dependence of the right-hand sides and initial conditions on time is omitted for simplicity.

An inherent advantage of the algorithms presented in the previous sections is their modular nature, since this allows their relatively simple combination. Recall that the bilevel algorithm summarized in Sect. 2 formulates a series of single-level optimization problems which can again be solved using black-boxes. Recall also that in the algorithms for global dynamic optimization described in Sect. 3 the sequential approach is used for solving the optimization problem only in the parameters, while treating the functions as black boxes which return their values, gradients and convex relaxations.

4.1 Scenario mode without operational degrees of freedom

In this subsection the formulation (SIOP6) by Abel and Marquardt [1] is revisited and a solution method is proposed based on the assumptions made. This formulation considers the case that in the failure mode no operational degrees of freedom are available, due to constraints, or a predetermined shutdown policy. Here, it is assumed that the time horizon of the scenario operation is known a priori, or its end time is uniquely determined by a state event, which for simplicity is not included in the formulations. Depending on the application, this may be a significant restriction, but the generalization requires considerations which go beyond the scope of this article. In the car example by Abel and Marquardt the end time is uniquely determined by the car coming to a halt, i.e., by $v^l(t^l = t^{l,f}) = 0$. It is also assumed that an upper bound for $t^{u,f}$ is known and that the parameters $\mathbf{p}^u$ are bounded.

After a change of notation and scaling the following upper-level program is obtained:

$$
\begin{aligned}
f^{u,*} = \min_{\mathbf{p}^u, t^{u,f}} \; & f^u(\mathbf{x}^u(\tau^u = 1), \mathbf{p}^u) \\
\text{s.t. } & \mathbf{g}^{u,f}(\mathbf{x}^u(\tau^u = 1), \mathbf{p}^u) \le \mathbf{0} \\
& \mathbf{x}^u(\tau^u = 0) = \mathbf{x}_0^u(\mathbf{p}^u) \\
& \dot{\mathbf{x}}^u(\tau^u) = t^{u,f}\mathbf{g}^{u,c}(\mathbf{x}^u(\tau^u), \mathbf{p}^u), \quad \forall \tau^u \in (0, 1] \\
& v(t^{u,f}, \mathbf{p}^u) \le 0 \\
& \mathbf{p}^u \in P^u \subset \mathbb{R}^{n_u}, \quad t^{u,f} \ge 0.
\end{aligned}
\tag{37}
$$

The upper-level variables are the parameters of the nominal operation $\mathbf{p}^u$ and the final time of the nominal operation (without failure). For a given choice of upper-level variables, the following lower-level program gives the maximum constraint violation during scenario operation:

$$
\begin{aligned}
v(t^{u,f}, \mathbf{p}^u) = \max_{\tau^{u,s}} \max_k \; & g_k^{l,f}(\mathbf{x}^l(\tau^l = 1), \mathbf{p}^u) \\
\text{s.t. } & \mathbf{x}^l(\tau^l = 0) = \mathbf{x}_0^l(\mathbf{x}^u(\tau^u = \tau^{u,s}), \mathbf{p}^u) \\
& \dot{\mathbf{x}}^l(\tau^l) = (t^{l,f} - t^{u,f}\,\tau^{u,s})\,\mathbf{g}^{l,c}(\mathbf{x}^l(\tau^l), \mathbf{p}^u), \quad \forall \tau^l \in (0, 1] \\
& \tau^{u,s} \in [0, 1].
\end{aligned}
\tag{38}
$$

The lower-level program (38) is box-constrained, always feasible (under the assumptions on existence of solution to the ODE described in Sect. 3) and its feasible region does not depend on the upper-level variables. The overall program can be therefore classified as a dynamic extension to a regular SIP. This in turn is a special case of a dynamic bilevel optimization problem. Without the scaling of the dynamics the feasible set of the lower-level program would be the dynamic analog of a GSIP, compare also [27]. Note that the transformation to an SIP is only possible by the assumption that the failure time can occur at any point in time. If the failure time was dependent on the decision variables, a GSIP would be obtained. Note also that typically the maximum over a finite number of functions in the objective ($\max_k$) is reformulated with an auxiliary variable. Another interpretation is that the semi-infinite constraint is a generalized path constraint in the dynamic optimization problem.

The lower-level problem (38) is interesting in its own right, because it can be used to check for the feasibility (or the extent of constraint violation) of a given scenario operation, e.g., obtained by an approximate method. Recall that this program must be solved to global optimality, and this gives a motivation for the algorithms for dynamic optimization developed in Sect. 3. Under the assumptions made on $t^{l,f}$ the lower-level problem (38) can be readily solved with these algorithms.

In principle one could extend the (G)SIP proposals of [8,9,17,28,35] which are based on a relaxation of the semi-infinite constraint. This is a very elaborate endeavor, but has the advantage that a feasible point would be obtained finitely. A simpler (at least conceptually) approach is to apply the approach described for bilevel programs in Sect. 2, or equivalently extend the SIP algorithm by Blankenship and Falk [10]. A drawback of this algorithm is that in general it only generates ε-feasible points finitely. Assuming that the constraint has a built-in safety margin, this small violation would however be acceptable. In addition to the lower-level program (38) the algorithm requires the solution of the lower-bounding problem:

$$\begin{aligned}
\min_{\mathbf{p}^u, t^{u,f}} \; & f^u(\mathbf{x}^u(\tau^u = 1), \mathbf{p}^u) \\
\text{s.t. } & \mathbf{g}^{u,f}(\mathbf{x}^u(\tau^u = 1), \mathbf{p}^u) \leq \mathbf{0} \\
& \mathbf{x}^u(\tau^u = 0) = \mathbf{x}_0^u(\mathbf{p}^u) \\
& \dot{\mathbf{x}}^u(\tau^u) = t^{u,f}\mathbf{g}^{u,c}(\mathbf{x}^u(\tau^u), \mathbf{p}^u), \quad \forall \tau^u \in (0, 1] \\
& v^d(t^{u,f}, \mathbf{p}^u, \tau^{u,s}) \leq 0, \quad \forall \tau^{u,s} \in T^{u,d},
\end{aligned} \tag{39}$$

where $T^{u,d}$ is a finite subset of [0,1], and v^d is given by a function evaluation (i.e., a dynamic simulation) of the lower-level program for fixed values of $\tau^{u,s}$. Note that the lower-bounding problem is significantly simplified by the assumption that the final time of the failure mode $t^{l,f}$ is known.

The lower-bounding problem (39) can be easily reformulated as a multi-stage dynamic optimization problem. Formalizing and implementing an efficient algorithm for (39) is a non-trivial task. However, the theoretical development given in Sect. 3.3 suffices since both the transition times and mode sequence is known at the transition times $\tau^{u,s}$. Compare also the algorithm by Lee et al. [25,26] for hybrid systems with fixed mode sequence and transition times.

The overall algorithm for the solution of (37) can be summarized by:

Algorithm 3 (Algorithm for scenario-integrated optimization without operational degrees of freedom during scenario mode)

1. (**Initialization**)
   Set $T^{u,d} = \emptyset$.
2. (**Lower Bounding**)
   Solve (39) globally.
   IF Feasible **THEN**
   - Set $(\bar{\mathbf{p}}^u, \bar{t}^{u,f})$ equal to the solution point (ε_{NLP}-optimal point).

   ELSE
   - Terminate.

   END
3. (**Lower-Level Program**)
   Solve dynamic optimization problem (38) globally for $\mathbf{p}^u = \bar{\mathbf{p}}^u$ and $t^{u,f} = \bar{t}^{u,f}$.
   Add the optimal solution point (ε_{NLP}-optimal point) to $T^{u,d}$.

Set v^* equal to the optimal objective value (final overestimate).
IF $v^* < \varepsilon$ **THEN**

- $(\bar{\mathbf{p}}^u, \bar{t}^{u,f})$ is an ε-optimal point; terminate.

ELSE

- Goto step 2.

END

This algorithm is guaranteed to converge finitely because the set $T^{u,d}$ increases in dimensionality and in the worse case converges to the interval [0,1]. A formal proof can follow the line of proof in [34].

Recall that it is necessary to solve (38) to global optimality. On the other hand, it is acceptable to solve (39) to local optimality. In that case at termination Algorithm 3 will generate an ε-feasible point (unless the solution of (39) fails).

4.2 Scenario mode with operational degrees of freedom

Abel and Marquardt [1] discussed also the case that the scenario operation has degrees of freedom and gave a formulation (SIOP4) for a fixed failure time. In this subsection a conceptual solution of this formulation is given, followed by a discussion of the complications to extending to the more realistic case of a unknown failure time.

4.2.1 Fixed relative failure time

After a change of notation, formulation (SIOP4) by Abel and Marquardt [1] is given by the following,

$$\begin{aligned}
f^{u,*} = \min_{\mathbf{p}^u,\mathbf{p}^l,t^{u,f},t^{l,f}} \; & f^u(\mathbf{x}^u(\tau^u = 1), \mathbf{p}^u) \\
\text{s.t. } & \mathbf{g}^{u,f}(\mathbf{x}^u(\tau^u = 1), \mathbf{p}^u) \leq \mathbf{0} \\
& \mathbf{x}^u(\tau^u = 0) = \mathbf{x}_0^u(\mathbf{p}^u) \\
& \dot{\mathbf{x}}^u(\tau^u) = t^{u,f}\mathbf{g}^{u,c}(\mathbf{x}^u(\tau^u), \mathbf{p}^u), \quad \forall \tau^u \in (0,1] \\
& (\mathbf{p}^l, t^{l,f}) \in \arg\min_{\mathbf{p}^m,t^{m,f}} f^l(\mathbf{x}^l(\tau^l = 1), \mathbf{p}^u, \mathbf{p}^m) \\
& \qquad \text{s.t. } \mathbf{g}^{l,f}(\mathbf{x}^l(\tau^l = 1), \mathbf{p}^u, \mathbf{p}^m) \leq \mathbf{0} \\
& \qquad\qquad \mathbf{x}^l(\tau^l = 0) = \mathbf{x}_0^l(\mathbf{x}^u(\tau^u = \tau^{u,s}), \\
& \qquad\qquad\quad \mathbf{p}^u, \mathbf{p}^m) \\
& \qquad\qquad \dot{\mathbf{x}}^l(\tau^l) = (t^{m,f} - t^{u,f}\,\tau^{u,s})\mathbf{g}^{l,c} \\
& \qquad\qquad\quad (\mathbf{x}^l(\tau^l), \mathbf{p}^u, \mathbf{p}^m), \\
& \qquad\qquad\quad \forall \tau^l \in (0,1] \\
& \mathbf{p}^u \in P^u \subset \mathbb{R}^{n_u}, \quad \mathbf{p}^l, \mathbf{p}^m \in P^l \subset \mathbb{R}^{n_l}, \quad t^{u,f} \geq 0, \\
& t^{m,f} \geq t^{u,f}\,\tau^{u,s},
\end{aligned} \tag{40}$$

where $\tau^{u,s} \in [0,1]$ is now a constant. Note that the lower-level optimization variables ($\mathbf{p}^l$ and $t^{l,f}$) do not participate in the constraints or objective of the upper-level program. However, they are included as variables of the upper-level optimization, to emphasize that it is necessary to have a feasible lower-level program and that they are part of the solution of (40). Therefore, it is essentially a special case of the problem described by Yunt et al. [56] and could be solved as a single-stage optimization problem. Instead, in the following the conceptual solution of (40) in two steps is described since this leads to the more interesting case of variable failure time.

In the first step a nominal operation (parameterized by $\mathbf{p}^u$ and $t^{u,f}$) is sought which minimizes the objective of the nominal operation, allowing for safe operation for the given transition time; to do so a corresponding scenario operation (parameterized by $\mathbf{p}^l$ and $t^{l,f}$) must be found. This is given by the following single-level optimization problem:

$$
\begin{aligned}
f^{u,*} = \min_{\mathbf{p}^u,\mathbf{p}^l,t^{u,f},t^{l,f}} \; & f^u(\mathbf{x}^u(\tau^u=1),\mathbf{p}^u)\\
\text{s.t. } & \mathbf{g}^{u,f}(\mathbf{x}^u(\tau^u=1),\mathbf{p}^u) \le \mathbf{0}\\
& \mathbf{x}^u(\tau^u=0) = \mathbf{x}_0^u(\mathbf{p}^u)\\
& \dot{\mathbf{x}}^u(\tau^u) = t^{u,f}\mathbf{g}^{u,c}(\mathbf{x}^u(\tau^u),\mathbf{p}^u), \quad \forall \tau^u \in (0,1]\\
& \mathbf{g}^{l,f}(\mathbf{x}^l(\tau^l=1),\mathbf{p}^u,\mathbf{p}^l) \le \mathbf{0}\\
& \mathbf{x}^l(\tau^l=0) = \mathbf{x}_0^l(\mathbf{x}^u(\tau^u=\tau^{u,s}),\mathbf{p}^u,\mathbf{p}^l)\\
& \dot{\mathbf{x}}^l(\tau^l) = (t^{l,f}-t^{u,f}\,\tau^{u,s})\,\mathbf{g}^{l,c}(\mathbf{x}^l(\tau^l),\mathbf{p}^u,\mathbf{p}^l),\\
& \qquad \forall \tau^l \in (0,1]\\
& \mathbf{p}^u \in P^u \subset \mathbb{R}^{n_u}, \quad \mathbf{p}^l \in P^l \subset \mathbb{R}^{n_l}, \quad t^{u,f} \ge 0, \quad t^{l,f} \ge t^{u,f}\,\tau^{u,s},\\
& t^{m,f} \ge t^{u,f}\,\tau^{u,s},
\end{aligned}
$$

which is essentially a multi-stage dynamic optimization problem that can be solved with the methods discussed in Sect. 3. Suppose that $(\bar{\mathbf{p}}^u, \bar{\mathbf{p}}^l, \bar{t}^{u,f}, \bar{t}^{l,f})$ is a global solution of this program. Then $(\bar{\mathbf{p}}^u, \bar{t}^{u,f})$ gives the optimal nominal operation of (40). The reformulation from bilevel to single-level is possible because the lower-level optimization variables ($\mathbf{p}^l$ and $t^{l,f}$) do not participate in the constraints or objective of the upper-level program [56].

To find the optimal scenario operation, $\mathbf{p}^u = \bar{\mathbf{p}}^u$, $t^{u,f} = \bar{t}^{u,f}$ are fixed and the following problem is solved

$$
\begin{aligned}
\min_{\mathbf{p}^l,t^{l,f}} \; & f^l(\mathbf{x}^l(\tau^l=1),\bar{\mathbf{p}}^u,\mathbf{p}^l)\\
\text{s.t. } & \mathbf{g}^{l,f}(\mathbf{x}^l(\tau^l=1),\bar{\mathbf{p}}^u,\mathbf{p}^l) \le \mathbf{0}\\
& \mathbf{x}^l(\tau^l=0) = \mathbf{x}_0^l(\mathbf{x}^u(\tau^u=\tau^{u,s}),\bar{\mathbf{p}}^u,\mathbf{p}^l)\\
& \dot{\mathbf{x}}^l(\tau^l) = (t^{l,f}-t^{u,f}\,\tau^{u,s})\,\mathbf{g}^{l,c}(\mathbf{x}^l(\tau^l),\bar{\mathbf{p}}^u,\mathbf{p}^l),\\
& \qquad \forall \tau^l \in (0,1]\\
& \mathbf{p}^l \in P^l \subset \mathbb{R}^{n_l}, \qquad t^{l,f} \ge t^{u,f}\,\tau^{u,s}.
\end{aligned}
\tag{41}
$$

This is a regular dynamic optimization problem and can be solved with the methods discussed in Sect. 3.

It should be noted that the scenario operation is only guaranteed to be optimal for the particular values of the upper-level variables $\bar{\mathbf{p}}^u$, $\bar{t}^{u,f}$. A much lower objective value for the scenario operation could be achieved for different values of the upper-level variables. This is not a result of the two-step procedure utilized, but rather an inherent property of the bilevel formulation. Bilevel programs have a hierarchy of objective functions, with the primary goal of optimizing the upper-level program.

4.2.2 Variable failure time

In reality, the transition time $\tau^{u,s}$ is typically unknown and can take any value in the interval [0,1]. Abel and Marquardt [1] discuss an approximation where the transition time is only allowed to take a finite number of values. This is not rigorous because feasibility of the scenario-mode operation is not guaranteed. One could contemplate solving (40) parametrically as a function of the transition time, but this would only be rigorous if the transition time was known prior to the decision of the nominal operation, which is not realistic. Instead the nominal operation must be independent of the failure time, while the scenario operation is dependent on the failure time with the semi-infinite constraint that for any failure time there exists a feasible scenario operation.

Similarly to the solution of the fixed failure time problem (40) the overall problem could be solved in two steps. In the first step a nominal operation (parameterized by $\mathbf{p}^u$) is sought which minimizes the nominal operation, allowing for safe operation for any transition time.

$$
\begin{aligned}
f^{u,*} = \min_{\mathbf{p}^u, t^{u,f}} \; & f^u(\mathbf{x}^u(\tau^u = 1), \mathbf{p}^u) \\
\text{s.t. } & \mathbf{g}^{u,f}(\mathbf{x}^u(\tau^u = 1), \mathbf{p}^u) \leq \mathbf{0} \\
& \mathbf{x}^u(\tau^u = 0) = \mathbf{x}_0^u(\mathbf{p}^u) \\
& \dot{\mathbf{x}}^u(\tau^u) = t^{u,f}\mathbf{g}^{u,c}(\mathbf{x}^u(\tau^u), \mathbf{p}^u), \quad \forall \tau^u \in (0,1] \\
& v(t^{u,f}, \mathbf{p}^u) \leq 0 \\
& v(t^{u,f}, \mathbf{p}^u) = \max_{\tau^{u,s}} \min_{\mathbf{p}^m, t^{m,f}} \max_k g_k^{l,f}(\mathbf{x}^l(\tau^l = 1), \mathbf{p}^u, \mathbf{p}^m) \\
& \qquad \text{s.t.} \mathbf{x}^l(\tau^l = 0) = \mathbf{x}_0^l(\mathbf{x}^u(\tau^u = \tau^{u,s}), \mathbf{p}^u, \mathbf{p}^m, \tau^{u,s}) \\
& \qquad \dot{\mathbf{x}}^l(\tau^l) = (t^{m,f} - t^{u,f}\,\tau^{u,s})\mathbf{g}^{l,c} \\
& \qquad\qquad (\mathbf{x}^l(\tau^l), \mathbf{p}^u, \mathbf{p}^m), \quad \forall \tau^l \in (0,1] \\
& \mathbf{p}^u \in P^u \subset \mathbb{R}^{n_u}, \quad \mathbf{p}^l, \mathbf{p}^m \in P^l \subset \mathbb{R}^{n_l}, \quad t^{u,f} \geq 0, \quad t^{l,f} \geq \tau^{u,s}, \quad t^{m,f} \geq \tau^{u,s}, \\
& \tau^{u,s} \in [0,1].
\end{aligned}
\tag{42}
$$

This program has three levels and requires significant development compared to the methods described in Sect. 2. This development is outside of the scope of this article. Note that the three levels result from allowing operational degrees of freedom in the scenario operation and are not due to the different objective function. Note also that typically the maximum over a finite number of functions in the objective ($\max_k$) is reformulated with an auxiliary variable.

At the second step, the nominal operation is fixed and problem (41) for the scenario operation can be optimized as a function of a known transition time via parametric optimization. Simple discretization of the failure time is not rigorous for this problem because feasibility for the other points is not guaranteed. Developing an extension of the dynamic optimization algorithms from Sect. 3 to the parametric case is a major endeavor and outside the scope of this article.

4.2.3 Illustrative example

To illustrate the additional modeling capabilities by (42) in this subsection an extension of the car example by Abel and Marquardt [1] is considered. Recall that the nominal operation is to cover a distance of 300 m in minimal time. The acceleration/deceleration is the control variable, while the distance and the velocity are the state variables. The initial and final constraints are to start and end with zero velocity. The path constraints are bounds on velocity and acceleration. The scenario considered is a break failure down to 10% of the nominal deceleration value and the hard constraint is that the distance covered is less than 350 m. In the literature example a conservative approach is taken for the scenario mode, namely that the driver uses the full remainder of the break power. This elimination of all operational degrees of freedom results in the semi-infinite formulation (37). However, as a consequence of this overly conservative approach, if the break failure occurs early in time, the car never reaches the destination. The bilevel formulation allows a less conservative operation during failure, for instance by imposing the constraint that the distance covered must be between 300 and 350 m for all failure times. In addition, an objective for the scenario operation can be set, for instance minimization of the time required to stop the car or minimization of the final distance.

This is not shown in formulation (42). For this extended car problem formulation (42) becomes:

$$
\begin{aligned}
\min_{\mathbf{p}^u, t^{u,f}, w} \quad & t^{u,f} \\
\text{s.t. } & v^u(\tau^u = 0) = 0 \\
& d^u(\tau^u = 0) = 0 \\
& v^u(\tau^u = 1) = 0 \\
& d^u(\tau^u = 1) = 300 \\
& \quad \dot{v}^u(\tau^u) = \frac{4}{\pi} t^{u,f} \arctan(a^u(\mathbf{p}^u, \tau^u)) - t^{u,f} k_f (v^u(\tau^u))^2, \quad \forall \tau^u \in (0, 1] \\
& \quad \dot{d}^u(\tau^u) = t^{u,f} v^u(\tau^u), \quad \forall \tau^u \in (0, 1] \\
& \qquad 0 \le v^u(\tau^u) \le 10, \quad \forall \tau^u \in (0, 1] \\
& \qquad -2 \le a^u(\mathbf{p}^u, \tau^u) \le 2, \quad \forall \tau^u \in (0, 1] \\
& \qquad w \le 0 \\
& \qquad w = \max_{\tau^{u,s}} \min_{\mathbf{p}^l, t^{l,f}} \max \left\{ 300 - d^l(\tau^l = 1), d^l(\tau^l = 1) - 350 \right\} \\
& \qquad\qquad \text{s.t. } v^l(\tau^l = 0) = v^u(\tau^u = \tau^{u,s}) \\
& \qquad\qquad\quad d^l(\tau^l = 0) = d^u(\tau^u = \tau^{u,s}) \\
& \qquad\qquad\quad v^l(\tau^l = 1) = 0 \\
& \qquad\qquad\quad \dot{v}^l(\tau^l) = \frac{4}{\pi} (t^{l,f} - t^{u,f} \tau^{u,s}) \arctan(a^l(\mathbf{p}^l, \tau^l)) \\
& \qquad\qquad\qquad - (t^{l,f} - t^{u,f} \tau^{u,s}) k_f (v^l(\tau^l))^2, \quad \forall \tau^l \in (0, 1] \\
& \qquad\qquad\quad \dot{d}^l(\tau^l) = (t^{l,f} - t^{u,f} \tau^{u,s}) v^l(\tau^l), \quad \forall \tau^l \in (0, 1] \\
& \qquad\qquad\quad 0 \le v^l(\tau^l) \le 10, \quad \forall \tau^l \in (0, 1] \\
& \qquad\qquad\quad -0.2 \le a^l(\mathbf{p}^l, \tau^l) \le 2, \quad \forall \tau^l \in (0, 1] \\
& \qquad\qquad\quad t^{l,f} \ge \tau^{u,s} \\
& t^{u,f} \in [0, 1], \qquad \tau^{u,s} \in [0, 1] \\
& \mathbf{p}^u \in P^u \subset \mathbb{R}^{n_u}, \quad \mathbf{p}^l \in P^l \subset \mathbb{R}^{n_l}.
\end{aligned}
$$

5 Conclusions

An overview of recently developed dynamic and bilevel optimization algorithms is given. Formulations for bilevel dynamic optimization are also analyzed. A combination of the algorithms for bilevel and dynamic optimization presented can be applied to some of these problems, such as the special case of scenario-integrated dynamic optimization wherein the scenario operation has no operational degrees of freedom. Under some assumptions this results in a bilevel dynamic optimization formulation which is tractable through a combination of the algorithms described. In future work numerical case studies and applications will be performed. To that extent some modifications of the existing implementations for dynamic optimization is required. Another interesting alternative is the extension of proposals from the semi-infinite literature [8,9,17,28,35].

It would also be interesting to relax some of the assumptions made for the special case of scenario-integrated dynamic optimization mentioned above. In particular it would be interesting to consider the case that the failure time is dependent on the values of the state variables; this would result in a GSIP as opposed to SIP formulations. Moreover, in general the final time of the scenario mode may not be uniquely determined by a single constraint; it is unclear whether this would result in currently tractable optimization problems.

A formulation for scenario-integrated optimization is refined, wherein the scenario has operational degrees of freedom and possibly an objective. Unfortunately, this formulation results into a currently intractable program with three levels, namely an optimization program constrained by a max–min program embedded. The extension of the bilevel algorithm to this case is of interest.

Acknowledgements This work is based upon work supported by the National Science Foundation under Grant CTS-0521962.

References

1. Abel, O., Marquardt, W.: Scenario-integrated modeling and optimization of dynamic systems. AIChE J. **46**(4), 803–823 (2000)
2. Adjiman, C.S., Dallwig, S., Floudas, C.A., Neumaier, A.: A global optimization method, αBB, for general twice-differentiable constrained NLPs. I. Theoretical advances. Comput. Chem. Eng. **22**(9), 1137–1158 (1998)
3. Bard, J.F.: An algorithm for solving the general bilevel programming problem. Math. Oper. Res. **8**(2), 260–272 (1983)
4. Bard, J.F.: Practical Bilevel Optimization: Algorithms and Applications. Nonconvex Optimization and Its Applications. Kluwer Academic Publishers, Dordrecht (1998)
5. Ben-Tal, A.A., Nemirovski, A.: Robust optimization—methodology and applications. Math. Program. **92**(3), 453–480 (2002)
6. Berkovitz, L.D.: Convexity and Optimization in $\mathbb{R}^n$. Series Pure and Applied Mathematics. John Wiley & Sons Inc, New York (2002)
7. Bertsimas, D., Tsitsiklis, J.N.: Introduction to Linear Optimization. Athena Scientific, Belmont, Massachusetts (1997)
8. Bhattacharjee, B., Green, W.H. Jr., Barton, P.I.: Interval methods for semi-infinite programs. Comput. Optim. Appl. **30**(1), 63–93 (2005)
9. Bhattacharjee, B., Lemonidis, P., Green, W.H. Jr., Barton, P.I.: Global solution of semi-infinite programs. Math. Program. Ser. B **103**(2), 283–307 (2005)
10. Blankenship, J.W., Falk, J.E.: Infinitely constrained optimization problems. J. Optim. Theory Appl. **19**(2), 261–281 (1976)
11. Brusch, R., Schappelle, R.: Solution of highly constrained optimal control problems using nonlinear programming. AIAA J. **11**(2), 135–136 (1973)

12. Chachuat, B., Latifi, M.A.: A new approach in deterministic global optimization of problems with ordinary differential equations. In: Floudas, C.A., Pardalos, P.M., (eds.) Frontiers in Global Optimization, vol. 74 of Nonconvex Optimization and Its Applications, pp. 83–108. Kluwer Academic Publishers (2003)
13. Dempe, S.: Foundations of Bilevel Programming. Nonconvex Optimization and Its Applications. Kluwer Academic Publishers, Dordrecht (2002)
14. Esposito, W.R., Floudas, C.A.: Deterministic global optimization in nonlinear optimal control problems. J. Glob. Optim. **17**, 96–126 (2000)
15. Faisca, N.P., Dua, V., Rustem, B., Saraiva, P.M., Pistikopoulos, E.N.: Parametric global optimisation for bilevel programming. J. Glob. Optim. **38**(4), 609–623 (2007)
16. Falk, J.E., Hoffman, K.: A nonconvex max-min problem. Nav. Res. Logist. **24**(3), 441–450 (1977)
17. Floudas, C.A., Stein, O.: The adaptive convexification algorithm: a feasible point method for semi-infinite programming. SIAM J. Optim. **18**(4), 1187–1208 (2007)
18. Floudas, C.A., Gümüs, Z.H., Ierapetritou, M.G.: Global optimization in design under uncertainty: feasibility test and flexibility index problems. Ind. Eng. Chem. Res. **40**(20), 4267–4282 (2001)
19. Floudas, C.A., Akrotirianakis, I.G., Caratzoulas, S., Meyer, C.A., Kallrath, J.: Global optimization in the 21st century: advances and challenges. Comput. Chem. Eng. **29**(6), 1185–1202 (2005)
20. Grossmann, I.E.: Review of nonlinear mixed-integer and disjunctive programming techniques. Optim. Eng. **3**(3), 227–252 (2002)
21. Gümüs, Z.H., Floudas, C.A.: Global optimization of nonlinear bilevel programming problems. J. Glob. Optim. **20**(1), 1–31 (2001)
22. Harrison, G.W.: Dynamic models with uncertain parameters. In: Avula, X. (ed.) Proceedings of 1st International Conference Mathematical Modeling, vol. 1, pp. 295–304
23. Horst, R., Tuy, H.: Global Optimization: Deterministic Approaches. 3rd edn. Springer-Verlag, Berlin, Germany (1996)
24. Khalil, H.K.: Nonlinear Systems, 3rd edn. Prentice Hall, Upper Saddle River, NJ (2002)
25. Lee, C.K., Barton, P.I.: Global optimization of linear hybrid systems with varying transitions times. SIAM J. Control Optim. **47**(2), 791–816 (2008)
26. Lee, C.K., Singer, A.B., Barton, P.I.: Global optimization of linear hybrid systems with explicit transitions. Syst. Control Lett. **51**(5), 363–375 (2004)
27. Lemonidis, P.: Global optimization algorithms for semi-infinite and generalized semi-infinite programs. PhD thesis, Massachusetts Institute of Technology (2006)
28. Lemonidis, P., Barton, P.I.: Interval methods for generalized semi-infinite programs. In: International Conference on Parametric Optimization and Related Topics (PARAOPT VIII), Cairo, Egypt, Nov 27–Dec 1 2005
29. Lin, Y., Stadtherr, M.A.: Deterministic global optimization of nonlinear dynamic systems. AIChE J. **53**(4), 866–875 (2007)
30. Lin, Y., Stadtherr, M.A.: Validated solutions of initial value problems for parametric ODEs. Appl. Numer. Math. **57**(10), 1145–1162 (2007)
31. Luus, R., Cormack, D.E.: Multiplicity of solutions resulting from the use of variational methods in optimal control problems. Can. J. Chem. Eng. **50**, 309–311 (1972)
32. McCormick, G.P.: Computability of global solutions to factorable nonconvex programs: Part I—convex underestimating problems. Math. Program. **10**, 147–175 (1976)
33. Mitsos, A.: Man-portable power generation devices: product design and supporting algorithms. http://yoric.mit.edu/download/Reports/MitsosThesis.pdf. PhD thesis, Massachusetts Institute of Technology (2006)
34. Mitsos, A., Lemonidis, P., Barton, P.I.: Global solution of bilevel programs with a nonconvex inner program. J. Glob. Optim. **42**(4), 475–513 (2008)
35. Mitsos, A., Lemonidis, P., Lee, C.K., Barton, P.I.: Relaxation-based bounds for semi-infinite programs. SIAM J. Optim. **19**(1), 77–113 (2008)
36. Moore, R.E.: Methods and Applications of Interval Analysis. SIAM, Philadelphia, PA (1979)
37. 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. **7**, 449–465 (2001)
38. Nemhauser, G.L., Wolsey, L.A.: Integer and Combinatorial Optimization. Wiley-Interscience, New York (1999)
39. Oluwole, O.O., Bhattacharjee, B., Tolsma, J.E., Barton, P.I., Green, W.H. Jr.: Rigorous valid ranges for optimally reduced kinetic models. Combust. Flame **146**(1–2), 348–365 (2006)
40. Papamichail, I., Adjiman, C.S.: A rigorous global optimization algorithm for problems with ordinary differential equations. J. Glob. Optim. **24**, 1–33 (2002)

41. Pistikopoulos, E.N., Dua, V., Ryu, J.-H.: Global optimization of bilevel programming problems via parametric programming. In: Floudas, C.A., Pardalos, P.M. (eds.) Frontiers in Global Optimization, pp. 457–476. Kluwer Academic Publishers, Santorini, Greece (2003)
42. Shimizu, K., Ishizuka, Y., Bard, J.F.: Nondifferentiable and Two-Level Mathematical Programming. Kluwer Academic Publishers, Boston (1997)
43. Singer, A.B., Barton, P.I.: Global solution of optimization problems with parameter-embedded linear dynamic systems. J. Optim. Theory Appl. **121**(3), 613–646 (2004)
44. Singer, A.B., Barton, P.I.: Bounding the solutions of parameter dependent nonlinear ordinary differential equations. SIAM J. Sci. Comput. **27**(6), 2167–2182 (2006)
45. Singer, A.B., Barton, P.I.: Global optimization with nonlinear ordinary differential equations. J. Glob. Optim. **34**(2), 159–190 (2006)
46. Singer, A.B., Taylor, J.W., Barton, P.I., Green, W.H.: Global dynamic optimization for parameter estimation in chemical kinetics. J. Phys. Chem. A **110**(3), 971–976 (2006)
47. Tawarmalani, M., Sahinidis, N.V.: Convexification and Global Optimization in Continuous and Mixed-Integer Nonlinear Programming. Nonconvex Optimization and its Applications. Kluwer Academic Publishers, Boston (2002)
48. Teo, K.L., Goh, C.J., Wong, K.H.: A Unified Computational Approach to Optimal Control Problems. Pitman Monographs and Surveys in Pure and Applied Mathematics. John Wiley & Sons, New York (1991)
49. Tsang, T.H., Himmelblau, D.M., Edgar, T.F.: Optimal control via collocation and nonlinear programming. Int. J. Control **21**, 763–768 (1975)
50. Tuy, H., Migdalas, A., Varbrand, P.: A global optimization approach for the linear 2-level program. J. Glob. Optim. **3**(1), 1–23 (1993)
51. Tuy, H., Migdalas, A., Varbrand, P.: A quasi-concave minimization method for solving linear 2-level programs. J. Glob. Optim. **4**(3), 243–263 (1994)
52. Tuy, H., Migdalas, A., Hoai-Phuong, N.T.: A novel approach to bilevel nonlinear programming. J. Glob. Optim. **38**(4), 527–554 (2007)
53. Vicente, L.N., Calamai, P.H.: Bilevel and multilevel programming—a bibliography review. J. Glob. Optim. **5**(3), 291–306 (1994)
54. Walter, W.: Differential and Integral Inequalities. Springer-Verlag, Berlin, Germany (1970)
55. Williams, H.P.: Model Building in Mathematical Programming, 3rd edn. John Wiley and Sons, Chichester
56. Yunt, M., Chachuat, B., Mitsos, A., Barton, P.I.: Designing man-portable power generation systems for varying power demand. AIChE J. **54**(5), 1254–1269 (2008)
57. Zuhe, S., Neumaier, A., Eiermann, M.C.: Solving minimax problems by interval-methods. BIT **30**(4), 742–751 (1990)