I&EC research
Industrial & Engineering Chemistry Research
Article
pubs.acs.org/IECR

Semi-Infinite Optimization with Implicit Functions

Matthew D. Stuber and Paul I. Barton*

Process Systems Engineering Laboratory, Dept. of Chemical Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, United States

ABSTRACT: In this work, equality-constrained bilevel optimization problems, arising from engineering design, economics, and operations research problems, are reformulated as an equivalent semi-infinite program (SIP) with implicit functions embedded, which are defined by the original equality constraints that model the system. Using recently developed theoretical tools for bounding implicit functions, a recently developed algorithm for global optimization of implicit functions, and a recently developed algorithm for solving standard SIPs with explicit functions to global optimality, a method for solving SIPs with implicit functions embedded is presented. The method is guaranteed to converge to ϵ-optimality in finitely many iterations given the existence of a Slater point arbitrarily close to a minimizer. Besides the Slater point assumption, it is assumed only that the functions are continuous and factorable and that the model equations are once continuously differentiable.

■ INTRODUCTION

Many engineering design feasibility and reliability problems give rise to optimization programs whose feasible sets are parametrized. This is because it is often of great interest to study performance and/or safety of engineering systems under parametric uncertainty. Particularly, it is important to study the performance and/or safety in the face of the worst case, giving rise to equality-constrained bilevel programs of the following form:

$$\begin{aligned} f^* = \min_{\mathbf{x}} & f(\mathbf{x}) \\ \text{s.t. } & 0 \geq \max_{\tilde{\mathbf{y}},\mathbf{p}} g(\mathbf{x}, \tilde{\mathbf{y}}, \mathbf{p}) \\ & \text{s.t. } \mathbf{h}(\mathbf{x}, \tilde{\mathbf{y}}, \mathbf{p}) = \mathbf{0} \\ & \mathbf{x} \in X = \{\mathbf{x} \in \mathbb{R}^{n_x}: \mathbf{x}^L \leq \mathbf{x} \leq \mathbf{x}^U\} \\ & \mathbf{p} \in \mathbf{P} = \{\mathbf{p} \in \mathbb{R}^{n_p}: \mathbf{p}^L \leq \mathbf{p} \leq \mathbf{p}^U\} \\ & \tilde{\mathbf{y}} \in D_y \subset \mathbb{R}^{n_y} \end{aligned} \tag{1}$$

It is assumed that the objective function $f: D_x \to \mathbb{R}$ and the inequality constraint function $g: D_x \times D_y \times D_p \to \mathbb{R}$ are continuous and are factorable in the sense that they are composed from elementary arithmetic operations and transcendental functions. The equality constraints of eq 1 are the system of equations representing a steady-state model of the system of interest:

$$\mathbf{h}(\mathbf{x}, \tilde{\mathbf{y}}, \mathbf{p}) = \mathbf{0} \tag{2}$$

with $\mathbf{h}: D_x \times D_y \times D_p \to \mathbb{R}^{n_y}$, also assumed factorable and continuously differentiable on its domain with $D_x \subset \mathbb{R}^{n_x}$, $D_p \subset \mathbb{R}^{n_p}$ open. Due to the complexity of the models to be considered in this problem, the bilevel formulation eq 1 is intractable, or even impossible to solve.

It is proposed in this work that, under some relatively mild assumptions, the equality constraints are used to eliminate $\tilde{\mathbf{y}}$ and eq 1 is reformulated as an equivalent semi-infinite program (SIP) without equality constraints. If a $\tilde{\mathbf{y}}$ exists that satisfies eq 2 for each $(\mathbf{x},\mathbf{p}) \in X \times P \subset D_x \times D_p$, then it defines an implicit function of $(\mathbf{x},\mathbf{p})$, expressed as $\mathbf{y}(\mathbf{x},\mathbf{p})$. It must be assumed that there exists at least one implicit function $\mathbf{y}: X \times P \to Y$ such that $\mathbf{h}(\mathbf{x},\mathbf{y}(\mathbf{x},\mathbf{p}),\mathbf{p}) = \mathbf{0}$, $\forall(\mathbf{x},\mathbf{p}) \in X \times P$ with $Y \subset D_y$. Conditions guaranteeing uniqueness of $\mathbf{y} \in Y$ are given by the *semilocal implicit function theorem*.[1] Given the existence of an implicit function $\mathbf{y}$ (and its uniqueness in Y), the equality constraints can be eliminated and the program eq 1 can be expressed as

$$\begin{aligned} \min_{\mathbf{x}} & f(\mathbf{x}) \\ \text{s.t. } & 0 \geq \max_{\mathbf{p}} g(\mathbf{x}, \mathbf{y}(\mathbf{x}, \mathbf{p}), \mathbf{p}) \\ & \mathbf{x} \in X = \{\mathbf{x} \in \mathbb{R}^{n_x}: \mathbf{x}^L \leq \mathbf{x} \leq \mathbf{x}^U\} \\ & \mathbf{p} \in P = \{\mathbf{p} \in \mathbb{R}^{n_p}: \mathbf{p}^L \leq \mathbf{p} \leq \mathbf{p}^U\} \end{aligned} \tag{3}$$

Furthermore, the inner maximization program can be expressed as

$$\max_{\mathbf{p} \in P} g(\mathbf{x}, \mathbf{y}(\mathbf{x}, \mathbf{p}), \mathbf{p}) \leq 0 \Leftrightarrow g(\mathbf{x}, \mathbf{y}(\mathbf{x}, \mathbf{p}), \mathbf{p}) \leq 0, \quad \forall\, \mathbf{p} \in P \tag{4}$$

where the latter constraint is referred to as the (implicit) semi-infinite constraint. The following SIP, equivalent to the original bilevel program in eq 1, can then be formulated:

$$\begin{aligned} f^* = \min_{\mathbf{x}} & f(\mathbf{x}) \\ \text{s.t. } & g(\mathbf{x}, \mathbf{y}(\mathbf{x}, \mathbf{p}), \mathbf{p}) \leq 0, \quad \forall\, \mathbf{p} \in P \\ & \mathbf{x} \in X = \{\mathbf{x} \in \mathbb{R}^{n_x}: \mathbf{x}^L \leq \mathbf{x} \leq \mathbf{x}^U\} \\ & P = \{\mathbf{p} \in \mathbb{R}^{n_p}: \mathbf{p}^L \leq \mathbf{p} \leq \mathbf{p}^U\} \end{aligned} \tag{5}$$

Received: July 21, 2014
Revised: September 26, 2014
Accepted: December 11, 2014
Published: December 11, 2014

For a chemical engineering application, $\tilde{\mathbf{y}}$ may represent internal state variables, such as composition, determined by an equation of state or some other physics, the variables $\mathbf{x}$ may represent design variables such as chemical reactor dimensions or some pipe lengths, and $\mathbf{p}$ may represent uncertain model parameters such as reaction rate constants. In this case, f may represent some economic objective related to sizing and g may represent a critical performance and/or safety constraint. The global solution (if one exists) will correspond to the worst-case realization of uncertainty and address the question of optimal design under uncertainty. Alternatively, $\mathbf{x}$ may represent uncertainty in the system or environment and $\mathbf{p}$ may represent the controls. The function f may then represent some metric of uncertainty and g may again represent a performance and/or safety constraint. In that case, the global solution (if one exists) will correspond to the worst-case realization of uncertainty for which there exists a control setting such that the system meets specification. This formulation addresses the question of feasibility of the design as well as the determination of the maximum allowable uncertainty realization such that the design remains feasible.

In the next section, a background and literature review of semi-infinite programming is given. Then, the previous work on standard SIPs is extended to SIPs with implicit functions embedded including the full statement of the algorithm for solving such SIPs. Following the presentation of the algorithm, three numerical examples are presented and analyzed.

■ BACKGROUND

It has long been known that bilevel programs and SIPs can be utilized to address problems that are inherently uncertain. Kwak and Haug[2] addressed the question of optimal design under uncertainty using the bilevel formulation in eq 1 with first-order (affine) approximations of f, g, and $\mathbf{h}$. A stochastic version of the program in eq 1 was considered by Malik and Hughes[3] for optimal design of chemical processes under uncertainty. In Halemane and Grossmann,[4] the equality constraints were manipulated to give an explicit function of the control and uncertainty variables and a special case of the SIP in eq 5, known as a max−min, min−max, or "minimax" problem. This approach was explored further by Swaney and Grossmann[5] with application to addressing the feasibility problem. Two algorithms were presented by Swaney and Grossmann[6] for solving the problem under certain convexity requirements. This formulation was further explored by Stuber and Barton,[7] with general nonconvex g as in eq 5, in order to address the question of robust feasibility of engineering designs. A rigorous finite ϵ-convergent deterministic global optimization algorithm was presented that was based on the algorithm of Bhattacharjee et al.[8] The algorithm relies on the ability to solve the equality constraints approximately for an implicit function of the control and uncertainty variables using the successive-substitution fixed-point iteration[7] and so is limited to certain systems. Halemane and Grossmann[4] explained that the SIP formulation leads to an optimization problem that is not necessarily differentiable. In order to maintain differentiability, a special case of the bilevel formulation (eq 1), equivalent to that given by Halemane and Grossmann,[4] was further explored by Floudas et al.[9] The authors presented a rigorous deterministic algorithm, relying on twice-differentiability of $\mathbf{h}$ and g.[9] Ostrovsky and co-workers[10−12] developed algorithms for bounding solutions of the feasibility problem as formulated by Halemane and Grossmann[4] and Swaney and Grossmann.[5] The algorithms rely on iteratively solving nonlinear programming problems under similar certain convexity requirements. Therefore, the applicability is limited to such systems considered by Halemane and Grossmann[4] and Swaney and Grossmann.[5] Kwak and Haug[2] briefly discussed the relationship between the min−max problem formulation[3−7,9] and the bilevel formulation. In Mitsos et al.,[13] the authors presented an algorithm to address the bilevel formulation with a nonconvex inner program. However, their algorithm was limited in that it could only handle inequality constrained problems and so cannot be applied to eq 1. This paper will focus on solving the general SIP in eq 5, without relying on approximations of f, g, and $\mathbf{h}$, such as in Kwak and Haug,[2] with application to the min−max problems that show up in economics and design feasibility problems considered in the aforementioned articles,[3−7,9] only requiring continuity of f and g, once-differentiability of $\mathbf{h}$, the existence of a Slater point arbitrarily close to a minimizer, and the existence of a unique implicit function $\mathbf{y} \in Y$ such that $\mathbf{h}(\mathbf{x},\mathbf{y}(\mathbf{x},\mathbf{p}),\mathbf{p}) = \mathbf{0}$, $\forall(\mathbf{x},\mathbf{p}) \in X \times P$.

Solving SIPs with explicit functions (e.g., without implicit functions embedded as in eq 5), referred to as explicit SIPs herein, has been an active area of research for many years. An overview of the previous application of explicit SIPs to real-world problems with theoretical results and available methods can be found in previous works.[14−16] Contributions that have specific relevance to this work are summarized below.

Blankenship and Falk[17] present a cutting-plane algorithm for approximating solutions to explicit SIPs, which amounts to solving two nonlinear programs (NLPs), to global optimality in the general case, at each iteration. Their algorithm generates a sequence of (not necessarily feasible) points that converge to the solution of the SIP in the limit.[17] Under appropriate convexity assumptions, their algorithm converges finitely[17] to a feasible solution. Their method is applicable to SIPs in general and they make specific mention of the application to the max−min problem. The max−min problem is further explored by Falk and Hoffman[18] for general nonconvex functions. The cutting-plane algorithm relies on the techniques of discretization and what is called local reduction, which is a technique for theoretically describing (locally) the SIP feasible region with finitely many constraints.[19] Most SIP algorithms employ these techniques in various ways.[16]

Zuhe et al.[20] presented a method based on interval analysis for solving explicit min−max problems, again, which are special instances of explicit SIPs. Their method is applicable to min-max problems with twice continuously differentiable explicit functions. Interval analysis was used to dynamically exclude regions of the search space guaranteed not to contain solutions.[20] It was suggested that, using the properties of interval analysis and generalized bisection, their method converges in finitely many iterations.[20] Bhattacharjee et al.[21] applied interval analysis to the general case of explicit SIPs in order to construct what is called the inclusion-constrained reformulation, which is a valid restriction of the original explicit SIP. This idea was used further in the first algorithm for generating SIP-feasible points finitely, that relies on the inclusion-constrained reformulation.[8] A lower-bounding procedure that relies on McCormick's convex and concave relaxations[22] and discretization was introduced.[8] Together with the inclusion-constrained reformulation and the branch-and-bound (B&B) framework, Bhattacharjee et al. were able to solve SIPs to global optimality with guaranteed finite ϵ-optimal

DOI: 10.1021/ie5029123
Ind. Eng. Chem. Res. 2015, 54, 307−317

convergence.[8] As previously mentioned, this algorithm was employed by Stuber and Barton[7] to solve implicit max-min problems cast as implicit SIPs. Due to the overestimation of inclusion functions and the fact that the size of the upper- and lower-bounding problems grow rapidly with depth in the branch-and-bound tree,[8] this algorithm can be ineffective at solving implicit SIPs modeling more complex processes.

Stein and Still[23] solved explicit SIPs, with g convex, as a Stackelberg game using an interior-point method. By convexity of g, they were able to exploit the first-order optimality conditions to characterize the solution set of the inner program and solve equivalent finite nonlinear programs.[23] Floudas and Stein[24] used a similar idea and constructed concave relaxations of g on P using αBB.[25] They then replaced the inner program with its KKT optimality conditions and solved the resulting finite nonlinear program with complementarity constraints.[24] By doing so, the resulting program is a restriction of the original explicit SIP, and therefore, upon solution, it generates SIP-feasible points.[24] This idea was concurrently discussed by Mitsos et al.,[26] where they also considered a technique closely related to the inclusion-constrained reformulation[8,21] but instead used interval analysis to further construct Mc-Cormick-based concave relaxations[22] of g on P to restrict the inner program and generate SIP-feasible points finitely.

More recently, Mitsos[27] developed an algorithm based on the ideas of Blankenship and Falk[17] that relies on a new restriction technique for the upper-bounding procedure that requires the right-hand side of the semi-infinite constraint to be perturbed from zero. This formulation results in solving at least three NLP subproblems to global optimality at each iteration, in the general case, and the computational results reported[27] are quite promising. The key contribution is the novel upper-bounding procedure that is guaranteed to generate SIP-feasible points after finitely many iterations. It is stated explicitly that the algorithm only requires continuity of f and g and the existence of a Slater point arbitrarily close to a SIP minimizer, "provided the functions can be handled by the NLP solver".[27] Therefore, this algorithm *could* be applied to solve the SIP in eq 5 while handling the equality constraints directly, without requiring the introduction of the implicit function, by formulating each nonconvex subproblem as an equality-constrained global optimization problem. However, this strategy is not advisable since the algorithm would then require the number of variables in the upper- and lower-bounding subproblems to increase with each iteration. Thus, these subproblems become increasingly more expensive to solve with each iteration. However, this algorithm is a promising candidate for the global solution of SIPs with implicit functions embedded; the focus of this paper.

With the exception of Stuber and Barton,[7] all of the aforementioned methods were developed to solve explicit SIPs (or explicit min−max programs). For clarity, it is worth mentioning that all of the previously developed methods that are guaranteed to generate a rigorous SIP-feasible point in finitely many iterations rely on the existence of a Slater point or sequence of Slater points arbitrarily close to an SIP minimizer. If one were to simply reformulate eq 1 as an SIP and handle the equality constraints, $\mathbf{h}$, directly as a series of inequality constraints, as is commonly done in global optimization, the Slater point assumption would be violated. In other words, the rigorous methods are simply not applicable to programs of the form of eq 1. The major complication with formulating the bilevel program in eq 1 as the SIP in eq 5, is that an implicit function $\mathbf{y}$, which may not have a closed algebraic form, becomes embedded within the semi-infinite constraint g. Therefore, $\mathbf{y}$ (and g) cannot be evaluated directly but instead must be approximated using a numerical method, such as Newton's method or some other fixed-point iteration. In order to modify previously developed methods that rely on relaxations of the inner program, it must be possible to construct relaxations of $g(\cdot,\mathbf{y}(\cdot,\mathbf{p}),\mathbf{p})$ on X, $\forall \mathbf{p} \in P$. However, in order to relax $g(\cdot,\mathbf{y}(\cdot,\mathbf{p}),\mathbf{p})$ on X, $\forall \mathbf{p} \in P$, convex and concave relaxations of the implicit function $\mathbf{y}(\cdot,\mathbf{p})$ on X must be calculable. As previously mentioned, this has been achieved for problems in which the implicit function $\mathbf{y}$ could be approximated using the successive-substitution fixed-point iteration.[7] The theoretical details of these relaxations were presented by Scott et al.[28] This work will improve on the previous results of Stuber and Barton[7] and consider solving SIPs with more general implicit functions embedded that can be approximated using *any* available method, such as Newton's method, instead of being restricted to the successive-substitution case. This work will make use of a modified version of the algorithm developed by Mitsos,[27] where the solution of each of the (implicit) subproblems will be performed using the novel relaxation techniques and global optimization algorithm developed in a recently published article.[29]

In that article,[29] theoretical developments were made to construct convex and concave relaxations of more general implicit functions. The construction of these relaxations are analogous in many ways to how interval bounds can be calculated for implicit functions using (parametric) interval Newton-type methods. By applying (parametric) interval Newton-type methods[1,7,30,31] to a function $\mathbf{h}$, under certain conditions, an interval can be calculated that bounds a unique root, $\mathbf{y}$, of $\mathbf{h}$ over the set $X \times P$. Taking these bounds as *initial* relaxations of $\mathbf{y}$, they can be iteratively refined using the methods developed by Stuber et al.[29] to produce convex and concave relaxations of $\mathbf{y}$ on $X \times P$. As a result, global optimization of implicit functions was developed.[29] It should be noted that although global optimization of implicit functions will be relied upon in this paper, the intricate theoretical details of constructing relaxations of implicit functions and the workings of the B&B algorithm will not be necessary as these developments will be called upon simply as part of the external optimization subroutines.

In the next section, the global optimization algorithm for SIPs with implicit functions embedded is discussed. The application to min−max and max−min problems is made explicit, immediately following the statement of the algorithm. Finally, three numerical examples are given that illustrate the solution of implicit SIPs to global optimality.

■ GLOBAL SOLUTION OF SIPS WITH IMPLICIT FUNCTIONS EMBEDDED

The global optimization algorithm for implicit SIPs is based entirely on the cutting-plane algorithm presented by Mitsos,[27] which itself is based on the algorithm developed by Blankenship and Falk[17] but with a novel upper-bounding procedure. The algorithm, as applied to explicit SIPs is guaranteed to produce SIP-feasible points after finitely many iterations under the assumption that there exists a Slater point arbitrarily close to a minimizer.[27] As previously mentioned, the algorithm relies on the ability to solve three nonconvex NLP subproblems to global optimality at each iteration. The three

DOI: 10.1021/ie5029123
Ind. Eng. Chem. Res. 2015, 54, 307−317

subproblems are discussed below specialized to the case of implicit SIPs.

Lower-Bounding Problem. The lower-bounding procedure comes from a simple relaxation technique based on the adaptive discretization procedure originally described by Blankenship and Falk.[17] The SIP is reduced to an implicit NLP, again by considering only a finite number of constraints corresponding to realizations of $\mathbf{p} \in P^{\mathrm{LBD}}$ with $P^{\mathrm{LBD}} \subset P$ a finite set. The lower-bounding problem is formulated as

$$\begin{aligned} f^{\mathrm{LBD}} &= \min_{\mathbf{x}} f(\mathbf{x}) \\ &\text{s.t. } g(\mathbf{x}, \mathbf{y}(\mathbf{x}, \mathbf{p}), \mathbf{p}) \le 0, \quad \forall\, \mathbf{p} \in P^{\mathrm{LBD}} \\ &\mathbf{x} \in X = \{\mathbf{x} \in \mathbb{R}^{n_x}: \mathbf{x}^L \le \mathbf{x} \le \mathbf{x}^U\} \end{aligned} \tag{6}$$

In order to guarantee $f^{\mathrm{LBD}} \le f^*$, the lower-bounding problem must be solved to global optimality.

Inner Program. The inner program, stated explicitly in eqs 3 and 4, which is equivalent to the semi-infinite constraint, defines the SIP feasible region. Thus, given a candidate $\bar{\mathrm{x}} \in X$, feasibility can be determined by solving the inner (in general nonconvex) program:

$$\bar{g}(\bar{\mathbf{x}}) = \max_{\mathbf{p} \in P} g(\bar{\mathbf{x}}, \mathbf{y}(\bar{\mathbf{x}}, \mathbf{p}), \mathbf{p}) \tag{7}$$

The point $\bar{\mathrm{x}}$ is feasible in the original SIP given in eq 5 if $\bar{g}(\bar{\mathrm{x}}) \le 0$. Therefore, in order to determine feasibility of a candidate $\bar{\mathrm{x}}$, the inner program (eq 7) must be solved to global optimality for the general case.

Upper-Bounding Problem. The upper-bounding problem comes from perturbing the right-hand side of the semi-infinite constraint away from zero by a parameter $\epsilon^g > 0$, referred to as the restriction parameter,[27] and reducing the SIP to an implicit NLP by only considering a finite number of constraints corresponding to realizations of $\mathbf{p} \in P^{\mathrm{UBD}}$, where $P^{\mathrm{UBD}} \subset P$ is a finite set. The upper-bounding problem is formulated as

$$\begin{aligned} f^{\mathrm{UBD}} &= \min_{\mathbf{x}} f(\mathbf{x}) \\ &\text{s.t. } g(\mathbf{x}, \mathbf{y}(\mathbf{x}, \mathbf{p}), \mathbf{p}) \le -\epsilon^g, \ \forall\, \mathbf{p} \in P^{\mathrm{UBD}} \\ &\mathbf{x} \in X = \{\mathbf{x} \in \mathbb{R}^{n_x}: \mathbf{x}^L \le \mathbf{x} \le \mathbf{x}^U\} \end{aligned} \tag{8}$$

As mentioned by Mitsos,[27] the upper-bounding problem (eq 8) must be solved to global optimality in order for the algorithm to solve the original SIP (eq 5) to global optimality. However, a valid upper bound, $f^{\mathrm{UBD}} \ge f^*$, can be obtained by solving eq 8 locally for $\bar{\mathrm{x}}$ and verifying that it is feasible in the original SIP (eq 5).

Algorithm. In this section, the algorithm used for solving globally SIPs with implicit functions embedded to guaranteed ϵ-optimality is given. Again, as presented, this algorithm is an adaptation of the algorithm given by Mitsos[27] to SIPs with implicit functions embedded. Finite convergence of the algorithm for explicit SIPs was previously proven.[27] The results proven by Mitsos[27] extend directly to the implicit SIP algorithm provided that finite convergence of each implicit NLP subproblem can be guaranteed. The latter result was proven by Stuber et al.[29] The assumptions on which the SIP algorithm relies upon are stated explicitly in the following.

Assumption 1.

(a) The functions $f: D_x \to \mathbb{R}$ and $g: D_x \times D_y \times D_p \to \mathbb{R}$ are factorable[28] and continuous on their domains.

(b) Derivative information $\nabla_y h_i$, $i = 1,...,n_y$ is available and is factorable, say by automatic differentiation.[32,33]

(c) There exists $\mathbf{y}: X \times P \to D_y$ such that $\mathbf{h}(\mathbf{x},\mathbf{y}(\mathbf{x},\mathbf{p}),\mathbf{p}) = \mathbf{0}$, $\forall (\mathbf{x},\mathbf{p}) \in X \times P$, and an interval $Y \subset D_y$ is available such that $\mathbf{y}(X,P) \subset Y$ and $\mathbf{y}(\mathbf{x},\mathbf{p})$ is unique for every $(\mathbf{x},\mathbf{p}) \in X \times P$.

(d) A matrix $\mathbf{\Psi} \in \mathbb{R}^{n_y \times n_y}$ is known such that $A \equiv \mathbf{\Psi} J_y(X,Y,P)$ satisfies $0 \notin A_{ii}$ for all i, where J_y is an inclusion monotonic interval extension of the Jacobian matrix of $\mathbf{h}$, $\mathbf{J}_y$, on $X \times Y \times P$.

(e) There exists a point $\mathbf{x}^S \in X$ with $g(\mathbf{x}^S,\mathbf{y}(\mathbf{x}^S,\mathbf{p}),\mathbf{p}) < 0, \forall \mathbf{p} \in P$ such that $f(\mathbf{x}^S) - f^* < \epsilon_{\mathrm{tol}}$.

Assumption 1(a−d) are essentially required by the external subroutines for constructing convex and concave relaxations for global optimization of implicit functions.[29] Assumption 1(c) can be satisfied by applying parametric interval-Newton methods[1,7,30,31] and their key theoretical results. For Assumption 1(d), the matrix $\mathbf{\Psi}$ is a preconditioning matrix and has been the focus of many research articles. The application to interval-Newton methods is discussed by Kearfott,[34] among others. The interval-valued matrix A can be calculated efficiently by taking natural interval extensions[1,35] and thus satisfying Assumption 1(d). Assumption 1(e) is the ϵ_{tol}-optimal SIP-Slater point condition, which guarantees that a sequence of feasible points can be generated by the algorithm. Altogether, satisfying Assumption 1 guarantees that the following SIP algorithm terminates in finitely many iterations with a certificate of optimality and a rigorous ϵ_{tol}-optimal feasible point.[27,29] The algorithm for semi-infinite optimization with implicit functions embedded is presented in the following.

Algorithm 1 (Global Optimization Algorithm for Implicit SIPs)

1. (Initialization)
   (a) Set LBD = $-\infty$, UBD = $+\infty$, $\epsilon_{\mathrm{tol}} > 0$, $k := 0$.
   (b) Set initial parameter sets $P^{\mathrm{LBD}} = P^{\mathrm{LBD},0}$, $P^{\mathrm{UBD}} = P^{\mathrm{UBD},0}$.
   (c) Set initial restriction parameter $\epsilon^{g,0} > 0$ and $r > 1$.
2. (Termination) Check UBD − LBD $\le \epsilon_{\mathrm{tol}}$.
   (a) If true, $f^* :=$ UBD, terminate.
   (b) Else $k := k + 1$.
3. (Lower-Bounding Problem) Solve the lower-bounding problem (eq 6) to global optimality.
   (a) Set LBD $:= f^{\mathrm{LBD}}$, set $\bar{\mathrm{x}}$ equal to the optimal solution found.
4. (Inner Program) Solve the inner program (eq 7) to global optimality.
   (a) If $g(\bar{\mathrm{x}}, \mathrm{y}(\bar{\mathrm{x}},\bar{\mathrm{p}}), \bar{\mathrm{p}}) = \bar{g}(\bar{\mathrm{x}}) \le 0$, set $\mathrm{x}^* := \bar{\mathrm{x}}$, UBD $:= f(\bar{\mathrm{x}})$, terminate algorithm.
   (b) Else, add $\bar{\mathrm{p}}$ to P^{LBD}.
5. (Upper-Bounding Problem) Solve the upper-bounding problem (8) to global optimality.
   (a) If feasible:
      i. Set $\bar{\mathrm{x}}$ equal to the optimal solution found and solve the inner program (eq 7) to global optimality.
      ii. If $\bar{g}(\bar{\mathrm{x}}) < 0$:
         A. If $f(\bar{\mathrm{x}}) \le$ UBD, set UBD $:= f(\bar{\mathrm{x}})$, $\mathrm{x}^* := \bar{\mathrm{x}}$.
         B. Set $\epsilon^{g,k+1} := \epsilon^{g,k}/r$, go to 2.
      iii. Else $(\bar{g}(\bar{\mathrm{x}}) \ge 0)$, add $\bar{\mathrm{p}}$ to P^{UBD}, go to 2.
   (b) Else (infeasible), set $\epsilon^{g,k+1} := \epsilon^{g,k}/r$, go to 2.

DOI: 10.1021/ie5029123
Ind. Eng. Chem. Res. 2015, 54, 307−317

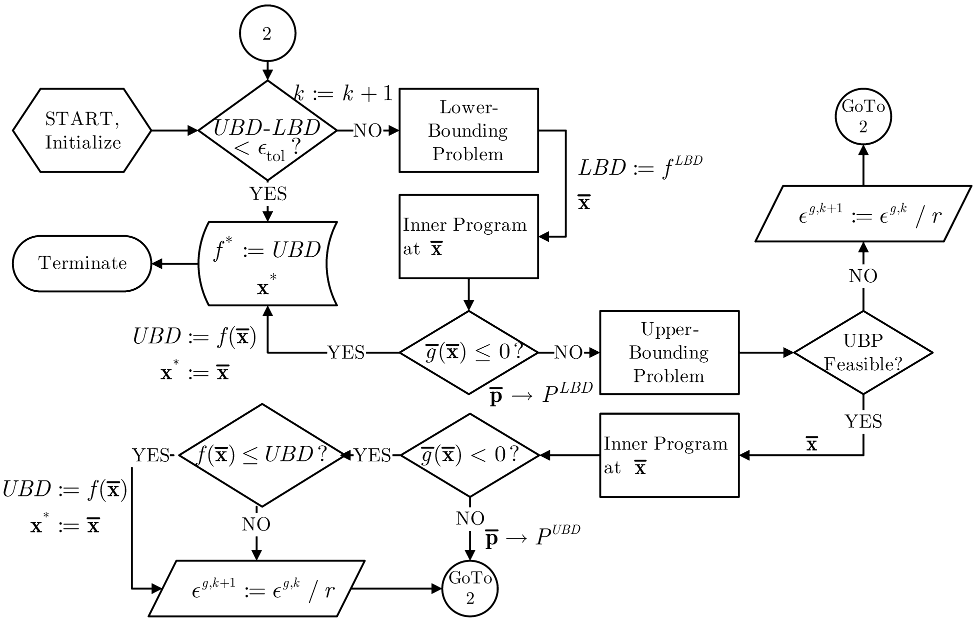

Figure 1. Flowchart for Algorithm 1.

It should be noted that the subproblems can only be solved finitely to within some chosen tolerances. In order to guarantee that the SIP algorithm is rigorous, the convergence tolerances for the subproblems must be set such that they are lower than ϵ_{tol}.

In order to better visualize Algorithm 1, the flowchart is given in Figure 1.

APPLICATION TO MAX–MIN AND MIN–MAX PROBLEMS

Constrained min–max problems of the form:

$$\begin{aligned} &\min_{\mathbf{x}\in X} \max_{\mathbf{p}\in P, \hat{\mathbf{y}}\in Y} G(\mathbf{x}, \hat{\mathbf{y}}, \mathbf{p}) \\ &\text{s.t. } \mathbf{h}(\mathbf{x}, \hat{\mathbf{y}}, \mathbf{p}) = \mathbf{0} \end{aligned} \tag{9}$$

and constrained max–min problems of the form:

$$\begin{aligned} &\max_{\mathbf{x}\in X} \min_{\mathbf{p}\in P, \hat{\mathbf{y}}\in Y} G(\mathbf{x}, \hat{\mathbf{y}}, \mathbf{p}) \\ &\text{s.t. } \mathbf{h}(\mathbf{x}, \hat{\mathbf{y}}, \mathbf{p}) = \mathbf{0} \end{aligned} \tag{10}$$

can also be solved using Algorithm 1. The min–max case results in solving the implicit program:

$$G^* = \min_{\mathbf{x}\in X} \max_{\mathbf{p}\in P} G(\mathbf{x}, \mathbf{y}(\mathbf{x}, \mathbf{p}), \mathbf{p}) \tag{11}$$

which can be reformulated as an implicit SIP by introducing a variable $\eta \in H \subset \mathbb{R}$, with H a compact interval, and writing:

$$\begin{aligned} &\min_{\mathbf{x}\in X, \eta\in H} \eta \\ &\text{s.t. } \eta \geq \max_{\mathbf{p}\in P} G(\mathbf{x}, \mathbf{y}(\mathbf{x}, \mathbf{p}), \mathbf{p}). \end{aligned} \tag{12}$$

Using the relationship given by eq 4, and setting $g(\mathbf{x},\mathbf{y}(\mathbf{x},\mathbf{p}),\mathbf{p},\eta) = G(\mathbf{x},\mathbf{y}(\mathbf{x},\mathbf{p}),\mathbf{p}) - \eta$, the following SIP can be written:

$$\begin{aligned} &\min_{\mathbf{x}\in X, \eta\in H} \eta \\ &\text{s.t. } g(\mathbf{x}, \mathbf{y}(\mathbf{x}, \mathbf{p}), \mathbf{p}, \eta) \leq 0, \quad \forall\, \mathbf{p} \in P \end{aligned} \tag{13}$$

which is equivalent to the implicit SIP in eq 5. The implicit SIP algorithm can be applied directly to this problem without any modification by setting $n_x := n_x + 1$ and treating η as the $n_x + 1$ component of $\mathbf{x}$. Here, an optimal solution value of $\eta^* \leq 0$ implies that $G^* \leq 0$, and alternatively, $\eta^* > 0$ implies $G^* > 0$.

The constrained max–min problem reformulation is slightly different. This case results in solving

$$G^* = \max_{\mathbf{x}\in X} \min_{\mathbf{p}\in P} G(\mathbf{x}, \mathbf{y}(\mathbf{x}, \mathbf{p}), \mathbf{p}) \tag{14}$$

Again, the variable $\eta \in H \subset \mathbb{R}$ is introduced and eq 14 is written as

$$\begin{aligned} &\max_{\mathbf{x}\in X, \eta\in H} \eta \\ &\text{s.t. } \eta \leq \min_{\mathbf{p}\in P} G(\mathbf{x}, \mathbf{y}(\mathbf{x}, \mathbf{p}), \mathbf{p}) \end{aligned} \tag{15}$$

which can be written as

$$\begin{aligned} &\max_{\mathbf{x}\in X, \eta\in H} \eta \\ &\text{s.t. } \eta \leq G(\mathbf{x}, \mathbf{y}(\mathbf{x}, \mathbf{p}), \mathbf{p}), \forall\, \mathbf{p} \in P \end{aligned} \tag{16}$$

or equivalently as

$$\begin{aligned} &\min_{\mathbf{x}\in X, \eta\in H} -\eta \\ &\text{s.t. } g(\mathbf{x}, \mathbf{y}(\mathbf{x}, \mathbf{p}), \mathbf{p}, \eta) \leq 0, \forall\, \mathbf{p} \in P \end{aligned} \tag{17}$$

by using the identity $g(\mathbf{x},\mathbf{y}(\mathbf{x},\mathbf{p}),\mathbf{p},\eta) = \eta - G(\mathbf{x},\mathbf{y}(\mathbf{x},\mathbf{p}),\mathbf{p})$. Now, the implicit SIP algorithm can be applied without modification by again setting $n_x{:} = n_x + 1$ and treating η as the $n_x + 1$ component of $\mathbf{x}$. Now, analogous to the min–max case, an optimal solution value of $\eta^* \leq 0$ implies that $G^* \leq 0$ and $\eta^* > 0$ implies $G^* > 0$.

DOI: 10.1021/ie5029123
Ind. Eng. Chem. Res. 2015, 54, 307–317

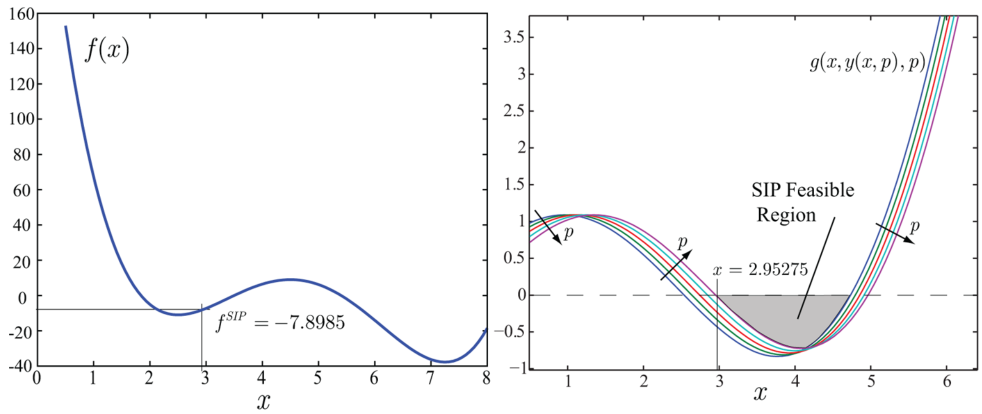

Figure 2. Objective function and implicit semi-infinite constraint for Example 1.

■ NUMERICAL EXAMPLES

Example 1. Consider the following illustrative example with $n_x = n_p = n_y = 1$:

$$f(x) = (x - 3.5)^4 - 5(x - 3.5)^3 - 2(x - 3.5)^2 + 15(x - 3.5)$$

$$h(x, \hat{y}, p) = \hat{y} - (x - x^3/6 + x^5/120)/\sqrt{\hat{y}} - p = 0$$

$$g(x, \hat{y}, p) = \hat{y} + \cos(x - p/90) - p \leq 0, \quad \forall\, p \in P$$

$$x \in X = [0.5, 8.0]$$

$$p \in P = [80, 120]$$

The objective function and implicit semi-infinite constraint are shown in Figure 2. An interval $Y = [68.8,149.9]$, guaranteed to contain a unique implicit function $y{:}P \times X \to Y$ was obtained using the parametric interval-Newton method.[1,7] This illustrative example was chosen because the SIP feasible set and the objective function are obviously nonconvex and the objective function has a suboptimal local minimum that is SIP-feasible.

Example 2. Consider the robust design of an isothermal flash separator under uncertainty. We wish to verify robust operation of a proposed design in the face of the worst-case realization of uncertainty. The flash separator is designed to separate a ternary mixture of *n*-butane, *n*-pentane, and *n*-hexane, with molar fractions of 0.5, 0.4, and 0.1, respectively. The separator is designed to create a vapor product stream with no more than 0.05 mol-fraction of *n*-hexane. To do so, it is designed to operate at 85 °C and a pressure no greater than 5100 torr (6.80 bar). It is expected that during operation, the vessel temperature, error in the thermocouple reading, or both, may vary by as much as ±5 °C. For this system, there are six unknowns: the compositions of the vapor and liquid streams. Three species balance equations and three phase-behavior equations can be written, resulting in a dimensionality $n_y = 6$. However, an alternative, and equivalent, model formulation with $n_y = 1$ can be formulated by writing the stream composition model equations in terms of the cut fraction $\hat{\alpha}$:

$$h(\tau, \hat{\alpha}, p) = \sum_i \frac{z_i(K_i(\tau, p) - 1)}{(K_i(\tau, p) - 1)\hat{\alpha} + 1} = 0$$

where τ will be the temperature (uncertain) variable, the cut fraction, $\hat{\alpha}$, is defined as the fraction of the feed that leaves in the vapor stream (internal state variable), p is the vessel pressure which can be controlled in order to mitigate fluctuations in τ, K_i is the vapor−liquid equilibrium coefficient for the i^{th} chemical component, and z_i is the mole-fraction of chemical component i in the feed. Solving $h(\tau,\hat{\alpha},p) = 0$ for $\hat{\alpha}$ defines the cut fraction as an implicit function of temperature and pressure, $\alpha{:}T \times P \to Y$. Any value $\alpha \notin [0,1]$ is nonphysical so the interval $Y = [0,1]$ was considered. For this system, the vapor−liquid equilibrium coefficient can be calculated as

$$K_i(\tau, p) = \frac{p_i^{\text{sat}}(\tau)}{p}$$

for each chemical component i with

$$\log_{10} p_i^{\text{sat}}(\tau) = A_i - \frac{B_i}{C_i + \tau}$$

with τ in °C and p_i^{sat} in torr. The Antoine coefficients A_i, B_i, C_i are available in Table 1.

Table 1. Antoine Coefficients for the Ternary Mixture in Example 2[36]

ex. two antoine coefficients				
i	A_i	B_i	C_i	temp. range (°C)
1: *n*-butane	7.00961	1022.48	248.145	−138.29−152.03
2: *n*-pentane	7.00877	1134.15	238.678	−129.73−196.5
3: *n*-hexane	6.9895	1216.92	227.451	−95.31−234.28

For robust design problems, one must consider the worst-case realization of uncertainty and examine if there exists a control setting that allows the design to still meet the performance and/or safety specification. This problem can be formulated mathematically as a max−min problem:

$$\max_{\tau \in T} \min_{p \in P} G(\tau, \alpha(\tau, p), p)$$

$$T = [80, 90]$$

$$P = [4400, 5100]$$

which, if $G(\tau^*,\alpha(\tau^*,p^*),p^*) \leq 0$, the design is robustly feasible, or simply, for the worst-case realization of uncertainty, there exists a control setting such that the system meets specification. The lower bound on the control variable comes from a requirement that there are two phases present in the separator at all times (i.e., any lower pressure will flash all of the liquid into the vapor phase). According to the previous discussion, this problem can be reformulated as an implicit SIP:

DOI: 10.1021/ie5029123
Ind. Eng. Chem. Res. 2015, 54, 307−317

$$\min_{\tau\in T,\eta\in H} \ -\eta$$
$$\text{s.t. } \eta - G(\tau, \alpha(\tau, p), p) \le 0, \quad \forall\, p \in P$$

with $H = [-1,1]$. The performance specification can be written as

$$G(\tau, \alpha(\tau, p), p) = \frac{z_3K_3(\tau, p)}{(K_3(\tau, p) - 1)\alpha(\tau, p) + 1} - 0.05 \le 0$$

which comes from material balances on the system. Figure 3 shows G plotted against τ.

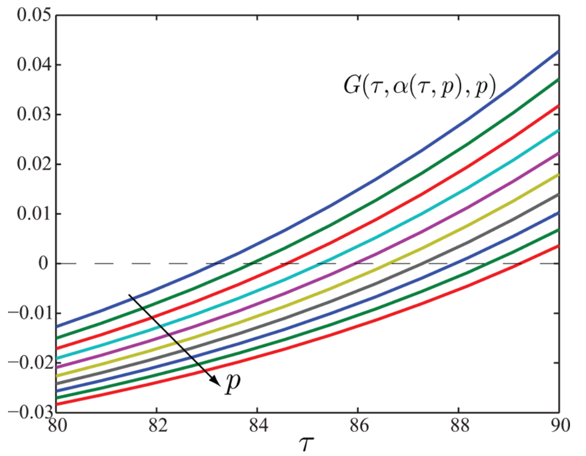

Figure 3. Design constraint function for Example 2.

Example 3. Consider the optimal design of a continuous-stirred tank reactor (CSTR) for the chlorination of benzene, shown in Figure 4. The reactions taking place are

$$C_6H_6 + Cl_2 \xrightarrow{k_1} C_6H_5Cl + HCl$$
$$C_6H_5Cl + Cl_2 \xrightarrow{k_2} C_6H_4Cl_2 + HCl$$

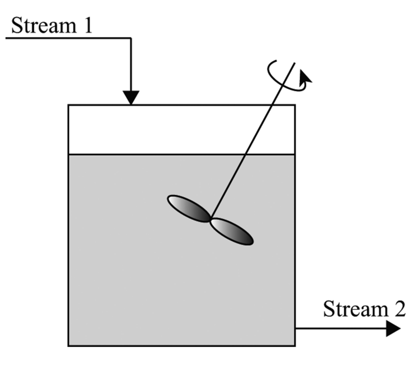

Figure 4. Continuous-stirred tank reactor for Example 3.

where the rate constants k_1 and k_2 (h^{-1}), as well as the feed flow rate F_1 (kmol/h), will be considered as uncertain parameters, $\mathbf{p} = (k_1\ k_2\ F_1)^T$. The design variable will be the reactor volume (m^3), $x = v$. The reaction kinetics can be considered to be first-order with respect to benzene and chlorobenzene and the reactions are irreversible.[37] For simplicity, A will denote C_6H_6, B will denote C_6H_5Cl, and C will denote $C_6H_4Cl_2$. Therefore, there are a total of four unknowns: the composition (mole-fractions) of the product stream and the product stream flow rate in terms of A, B, and C, $\hat{\mathbf{y}} = (y_A\ y_B\ y_C\ F_2)^T$. In this formulation, $n_x = 1$, $n_y = 4$, and $n_p = 3$. Note that F_1 and F_2 are the flow rates (kmol/h) in terms of the chemical species A, B, and C only. The model equations are then

$$\mathbf{h}(x, \hat{\mathbf{y}}, \mathbf{p}) = \begin{pmatrix} y_{A,1}p_3 - \hat{y}_1\hat{y}_4 - xr_1 \\ y_{B,1}p_3 - \hat{y}_2\hat{y}_4 + x(r_1 - r_2) \\ y_{C,1}p_3 - \hat{y}_3\hat{y}_4 + xr_2 \\ 1 - \hat{y}_1 - \hat{y}_2 - \hat{y}_3 \end{pmatrix} = \mathbf{0} \tag{18}$$

with $y_{i,1}$ as the mole-fraction of chemical specie i in the feed stream, and the reaction rates r_1 and r_2 are given by

$$r_1 = p_1\hat{y}_1/(\hat{y}_1V_A + \hat{y}_2V_B + \hat{y}_3V_C)$$
$$r_2 = p_2\hat{y}_2/(\hat{y}_1V_A + \hat{y}_2V_B + \hat{y}_3V_C)$$

with V_i as the molar volumes of chemical specie i: $V_A = 8.937 \times 10^{-2}$ m^3/kmol, $V_B = 1.018 \times 10^{-1}$ m^3/kmol, $V_C = 1.13 \times 10^{-1}$ m^3/kmol. The feed was taken to be pure benzene.

For this particular system, the design objective is to minimize the reactor volume while satisfying the performance constraint that at least 22 kmol C_6H_5Cl/h is produced:

$$\min_{x\in X} x$$
$$\text{s.t. } 22 - y_2(x, \mathbf{p})y_4(x, \mathbf{p}) \le 0, \quad \forall\, \mathbf{p} \in P$$

The uncertainty interval will be $P = [0.38,0.42] \times [0.053,0.058] \times [60,70]$, the design interval will be $X = [10,20]$. From the parametric interval-Newton method,[1,7] an interval $Y = [0.15,0.85] \times [0.3,0.65] \times [0.0,0.12] \times [60,70]$ was calculated that encloses the implicit function $\mathbf{y}:X \times P \to Y$ such that $\mathbf{h}(x,\mathbf{y}(x,\mathbf{p}),\mathbf{p}) = \mathbf{0}$, $\forall(x,\mathbf{p}) \in X \times P$.

EXPERIMENTAL CONDITIONS AND RESULTS

Algorithm 1 was implemented in C++. Each NLP subproblem was solved using the algorithm for global optimization of implicit functions,[29] which was also implemented in C++ and utilizes the library MC++.[38] The algorithm for global optimization of implicit functions relies on the ability to solve convex nonsmooth subproblems. This is because convex and concave relaxations of implicit functions[29] are in general nonsmooth. For this task, the nonsmooth bundle solvers PBUN and PBUNL[39] were utilized with default settings for the NLP lower-bounding problems, and the objective function was evaluated at NLP feasible points to obtain valid upper bounds on the NLP. Since the constrained bundle solver (PBUNL) can only handle affine constraints, affine relaxations of the convex constraints with respect to reference points must be calculated. In other words, once a nonconvex program is convexified by constructing relaxations of the nonconvex objective function and nonconvex constraints, the newly constructed convex constraints must be further relaxed by constructing affine underestimating functions that PBUNL can handle. The hierarchy of information flow for global optimization of implicit functions is shown in Figure 5.

Two sets of experiments were conducted.

Case 1: A single reference point—taken as the midpoint of X—was used to construct affine relaxations of constraints.

Case 2: Three reference points—the lower bound, the midpoint, and the upper bound of X—were used to construct affine relaxations of the constraints and used simultaneously.

DOI: 10.1021/ie5029123
Ind. Eng. Chem. Res. 2015, 54, 307−317

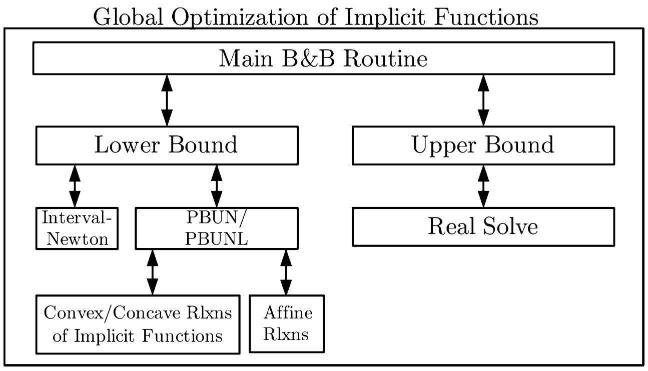

Figure 5. Hierarchy of information flow for global optimization of implicit functions.[29]

The numerical experiments were performed using a PC with an Intel Core2 Quad 2.66 GHz CPU operating Linux. For each example, absolute and relative convergence tolerances of 10^{-7} and 10^{-5}, respectively, were used for the NLP subproblems unless otherwise noted.

Example 1. For the SIP algorithm, each constraint set was initialized as empty, $\epsilon^{g,0} = 0.9$, $r = 2.0$, and $\epsilon_{tol} = 10^{-4}$. For each set of experiments, the implicit SIP algorithm was applied and an approximate global optimal solution with an objective function value of $f^* = -7.8985$ at $x^* = 2.95275$. Convergence was observed in three iterations for each case and took 0.211 s for Case 1 and 0.24815 s for Case 2. For this example, the algorithm terminates after the lower-bounding problem furnishes a SIP-feasible point (Step 4a of the algorithm) and so the parameter r does not affect the performance of the implicit SIP algorithm.

In an effort to explore the behavior of the algorithm, the NLP subproblem algorithm absolute and relative convergence tolerances were relaxed to 10^{-6} and 10^{-4}, respectively. Interestingly, for both cases, the SIP algorithm does not terminate with the lower-bounding problem furnishing an SIP-feasible point but instead terminates at Step 2 of the algorithm. This suggests that spending more time solving the NLP subproblems to higher accuracy benefits the total algorithm runtime by helping to accelerate locating a global optimal SIP-feasible point. The convergence results are plotted in Figure 6. Qualitatively, the algorithm performed similarly to the explicit SIP algorithm by Mitsos[27] where the solution time and number of iterations rapidly decreased with increasing reduction parameter. This behavior readily plateaued after a value of around $r = 8$. As expected, Case 2 exhibited higher computational cost without reducing the overall solution time or number of iterations.

Example 2. For the implicit SIP algorithm, each constraint set was initialized as empty, $\epsilon^{g,0} = 0.9$, $r = 2.0$, and $\epsilon_{tol} = 10^{-4}$. For both Case 1 and Case 2, the implicit SIP algorithm was applied and an approximate global optimal solution was obtained with $\eta^* = 3.6165 \times 10^{-3}$, $\tau^* = 90$ °C, $p^* = 5100$ torr. For both cases, the algorithm terminates in three iterations after the lower-bounding problem furnishes an SIP-feasible point. Thus, as previously mentioned, the parameter r has no effect on the performance of the algorithm. Relaxing the NLP convergence tolerances did not have the same effect on the behavior of the algorithm as it did in Example 1. Case 1 converged just after 0.381 s whereas Case 2 took 0.667 s. This suggests that the choice of the reference point for linearizing the constraints for the NLP subproblems was not very important and choosing more than one reference point just increased the computational complexity and, therefore, the solution time.

Returning to the idea of robust design, since $\eta^* > 0$, the flash separator design is not robust. However, as can be seen from Figure 3, if the design can be improved such that the temperature (or thermocouple reading) may only vary by ±4 °C, the design appears to be robust. This result was verified by the implicit SIP algorithm converging after three iterations and 0.386 s to an approximate optimal solution with $\eta^* = -1.015 \times 10^{-3}$.

Example 3. For the SIP algorithm, each constraint set was initialized as empty, $\epsilon^{g,0} = 0.9$, and $\epsilon_{tol} = 10^{-4}$. For Case 2, the implicit SIP algorithm was applied and an approximate global optimal solution was obtained with $f^* = x^* = 10.1794$ m^3, $\mathbf{p}^* = (0.38\ 0.058\ 60)^T$. Therefore, in order to produce at least 22 kmol/h of chlorobenzene, taking into account uncertainty in the input flow rate and the reaction rate constants, the reactor volume must be 10.1794 m^3. The reader is reminded that, similar to Example 2, this example is looking to solve the problem of robust design under uncertainty. However, there is a key difference in the formulation, between the two examples. Example 2 is seeking a "yes" or "no" answer to whether or not

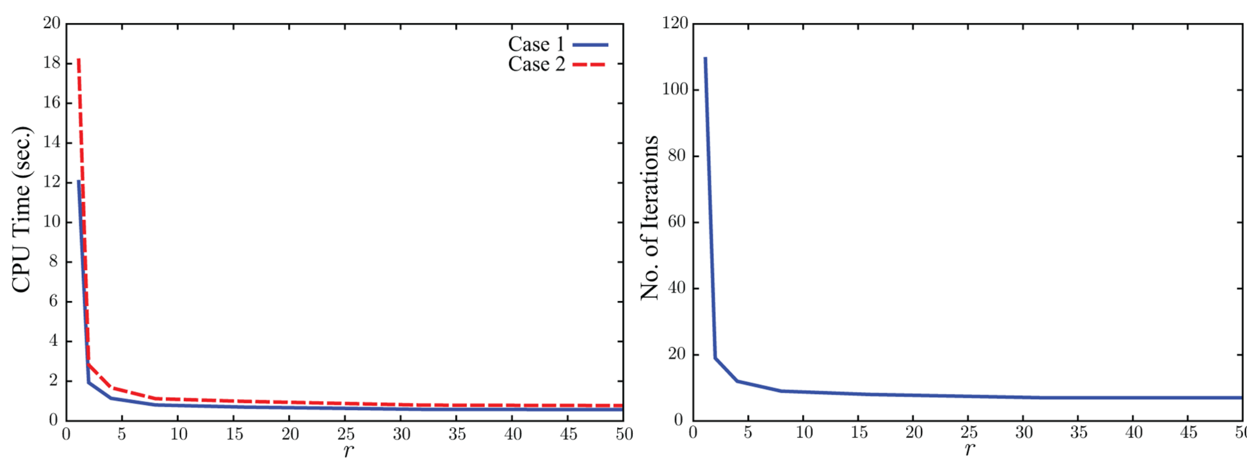

Figure 6. Computational effort in terms of the solution time (left) and the number of iterations (right) the algorithm takes to solve Example 1 versus the reduction parameter r. Note that the number of iterations is the same for both cases.

DOI: 10.1021/ie5029123
Ind. Eng. Chem. Res. 2015, 54, 307−317

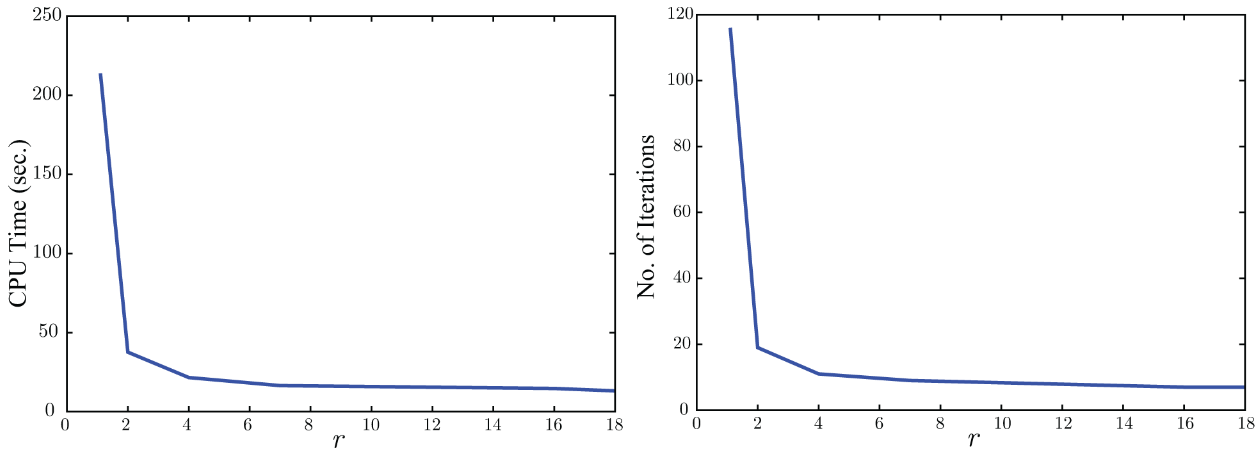

Figure 7. Computational effort in terms of the number of solution time (left) and the number of iterations (right) the algorithm takes to solve Case 2 of Example 3 versus the reduction parameter r.

the process is robust to the worst-case realization of uncertainty. Alternatively, this example is seeking to identify the actual size of the reactor such that the process will perform as desired in the face of the worst-case realization of uncertainty. Note that the worst-case realization of uncertainty is exactly what is to be expected; in order to have the least amount of chlorobenzene in the product stream, k_1 should be the smallest value it can take, k_2 should be the largest it can take, and the least amount of benzene should be fed to the reactor.

For a value of $r = 18$, the algorithm converges in 7 iterations and 13.17 s. The performance of the algorithm for Case 2 can be found in Figure 7. Similar to Example 1, a small value for r resulted in the implicit SIP algorithm taking many iterations to converge. As r was increased, the number of iterations required to converge, as well as the total solution time dropped drastically and plateaued. A parameter value of $r = 18$ reduced the solution time and number of iterations by 94% over $r = 1.1$. Again, qualitatively, this behavior is similar to what Mitsos[27] demonstrated. In his explicit SIP examples, the number of iterations was reduced from about 80 to between 5 and 20, depending on the example.

For this example, Case 1 failed to converge within 200 iterations of the algorithm. This result is simply a consequence of using PBUNL which only accepts affine constraints. In this case, since the affine constraints are being constructed with reference to the midpoint of X, the solver apparently fails to ever return a point that is feasible in the original SIP.

■ CONCLUSION

A method was presented for reformulating equality-constrained bilevel programs as SIPs with embedded implicit functions, requiring that

(a) all functions involved are continuous and factorable,
(b) derivative information on the equality constraint function is available and is factorable,
(c) there exists at least one solution $\mathbf{y}$ to the system of equations in eq 2 for every $(\mathbf{x},\mathbf{p}) \in X \times P$, and an interval Y can be found that bounds an isolated solution,
(d) an appropriate matrix for preconditioning the interval-Jacobian can be calculated such that their product has nonzero diagonal elements, and
(e) there exists a Slater point arbitrarily close to a SIP minimizer.

To solve the resulting implicit SIP, the global optimization algorithm developed by Mitsos[27] has been adapted. The algorithm relies on the ability to solve three nonconvex implicit NLP subproblems to global optimality. This is performed utilizing the relaxation methods and the deterministic algorithm for global optimization of implicit functions which were developed by Stuber et al.[29] The algorithm developed by Stuber et al.[29] relies on the ability to solve nonsmooth lower-and/or upper-bounding problems at each iteration. This can be done using any available nonsmooth optimization algorithm or using the calculated subgradient information to construct affine relaxations and transform the problem into a linear program and solved using any efficient LP optimization algorithm. For this paper, the nonsmooth bundle solvers PBUN and PBUNL[39] were utilized. Note that the requirements (b) and (c) are only due to current limitations of the algorithm for global optimization of implicit functions. The requirements (d) and (e) imply that the SIP is feasible and (a) and (e) are required for guaranteed ϵ-optimal convergence of the original explicit SIP algorithm[27] after finitely many iterations. Altogether, these requirements guarantee ϵ-optimal convergence of Algorithm 1.

As a proof-of-concept, three numerical examples were presented that illustrate the global solution of implicit SIPs using this algorithm. The first example illustrated the solution of a simple numerical system that fits the implicit SIP form given in eq 5. This problem is interesting because it is easy to visualize the nonconvexity of the functions and identify the SIP-feasible suboptimal local minimum. The second example was an engineering problem of robust design under uncertainty, originally cast as a constrained max-min problem. It was then reformulated as an implicit SIP of the form in eq 5 and solved using the implicit SIP algorithm. Under the original design conditions and uncertainty interval, the design was not robustly feasible. After altering the design such that the uncertainty interval was reduced, the design was found to be robust. The third example was an engineering problem of optimal design of a chemical reactor considering uncertainty in the kinetic parameters and feed rate which was formulated as an SIP. The examples chosen offered varying levels of complexity as well as size which allowed various features and behavior of the implicit SIP algorithm to be explored.

Due to the limitations of the PBUNL solver, only affine constraints could be used. Since the implicit semi-infinite constraint is almost surely nonlinear, affine relaxations must be

DOI: 10.1021/ie5029123
Ind. Eng. Chem. Res. 2015, 54, 307−317

constructed. For the numerical examples, two sets of experiments were conducted: one using a single reference point for constructing affine relaxations of the constraints and another using three reference points for constructing affine relaxations of the constraints and using them all simultaneously. The first method was hypothesized to be advantageous since it required less computational effort to calculate the constraints. Alternatively, the second method was hypothesized to be advantageous since using multiple reference points results in better approximations of the constraints, which in turn may speed up convergence of the overall algorithm. For Experiments 1 and 2, it was observed that Case 2 offered no benefit over Case 1 and only added computational complexity. However, for Example 3, Case 1 failed to converge after 200 iterations. This was likely due to the affine relaxations of the semi-infinite constraint not being very tight, resulting in PBUNL failing to find a solution that is feasible in the original SIP. It was found that for this problem, Case 2 and its multiple reference points for calculating affine relaxations were clearly superior with the algorithm converging after 7 iterations and 13.17 s.

Lastly, the generalization of eq 5 with the semi-infinite constraint $g \in \mathbb{R}$ to $\mathbf{g} \in \mathbb{R}^{n_g}$ will be discussed. For this case, eq 5 would take the form

$$\begin{aligned} f^* = \ & \min_{\mathbf{x}} f(\mathbf{x}) \\ & \text{s. t. } g_j(\mathbf{x}, \mathbf{y}(\mathbf{x}, \mathbf{p}), \mathbf{p}) \leq 0, j = 1, ..., n_g, \quad \forall\, \mathbf{p} \in P \\ & \mathbf{x} \in X = \{\mathbf{x} \in \mathbb{R}^{n_x}: \mathbf{x}^L \leq \mathbf{x} \leq \mathbf{x}^U\} \\ & P = \{\mathbf{p} \in \mathbb{R}^{n_p}: \mathbf{p}^L \leq \mathbf{p} \leq \mathbf{p}^U\} \end{aligned} \tag{19}$$

The explicit case was discussed by Mitsos.[27] The inner program would of course need to be replaced with n_g inner programs: one for each $j = 1,...,n_g$ and instead of a single constraint index set for the upper- and lower-bounding problems there should be a set P_j^{UBD} and P_j^{LBD} for each $j = 1,...,n_g$ constraints, respectively. Furthermore, instead of a single restriction $\epsilon^{g,k}$, there should be one for each constraint, $\epsilon_j^{g,k}$, $j = 1,...,n_g$. As noted by Mitsos,[27] since the constraints can be sufficiently different from one another, having a single constraint index set each for the lower-bounding problem and the upper-bounding problem would unnecessarily increase the problem size and number of constraints of the subproblems. Investigation into other strategies is still an open research objective.

■ AUTHOR INFORMATION

Corresponding Author

*E-mail: pib@mit.edu.

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

The authors give special thanks to Prof. Alexander Mitsos, RWTH Aachen, for providing information about the SIP algorithm on which this work was based. The authors acknowledge the Chevron University Partnership Program for funding this work through the MIT Energy Initiative.

■ REFERENCES

(1) Neumaier, A. *Interval Methods for Systems of Equations*; Cambridge University Press: Cambridge, 1990.
(2) Kwak, B. M.; Haug, J. E. J. Optimum design in the presence of parametric uncertainty. *J. Optim. Theory Appl.* **1976**, *19*, 527−546.
(3) Malik, R. K.; Hughes, R. R. Optimal design of flexible chemical processes. *Comput. Chem. Eng.* **1979**, *3*, 473−485.
(4) Halemane, K. P.; Grossmann, I. E. Optimal process design under uncertainty. *AIChE J.* **1983**, *29*, 425−433.
(5) Swaney, R. E.; Grossmann, I. E. An index for operational flexibility in chemical process design. Part I: Formulation and theory. *AIChE J.* **1985**, *31*, 621−630.
(6) Swaney, R. E.; Grossmann, I. E. An index for operational flexibility in chemical process design. Part II: Computational algorithms. *AIChE J.* **1985**, *31*, 631−641.
(7) Stuber, M. D.; Barton, P. I. Robust simulation and design using semi-infinite programs with implicit functions. *Int. J. Reliability Safety* **2011**, *5*, 378−397.
(8) Bhattacharjee, B.; Lemonidis, P.; Green, W. H., Jr.; Barton, P. I. Global solution of semi-infinite programs. *Math. Program* **2005**, *103*, 283−307.
(9) Floudas, C. A.; Gumus, Z. H.; Ierapetritou, M. G. Global optimization in design under uncertainty: Feasibility test and flexibility index problems. *Ind. Eng. Chem. Res.* **2001**, *40*, 4267−4282.
(10) Ostrovsky, G. M.; Volin, Y. M.; Barit, E. I.; Senyavin, M. M. Flexibility analysis and optimization of chemical plants with uncertain parameters. *Comput. Chem. Eng.* **1994**, *18*, 755−767.
(11) Ostrovsky, G. M.; Volin, Y. M.; Golovashkin, D. V. Optimization problem of complex system under uncertainty. *Comput. Chem. Eng.* **1998**, *22*, 1007−1015.
(12) Ostrovsky, G. M.; Achenie, L. E. K.; Wang, Y. A new algorithm for computing process flexibility. *Ind. Eng. Chem. Res.* **2000**, *39*, 2368−2377.
(13) Mitsos, A.; Lemonidis, P.; Barton, P. I. Global Solution of bilevel programs with a nonconvex inner program. *J. Global Optim.* **2008**, *42*, 475−513.
(14) Hettich, R.; Kortanek, K. O. Semi-infinite programming: Theory, methods, and applications. *SIAM Rev.* **1993**, *35*, 380−429.
(15) Lopez, M.; Still, G. Semi-infinite programming. *Eur. J. Optim. Res.* **2007**, *180*, 491−518.
(16) *Semi-Infinite Programming*; Reemtsen, R., Rückmann, J.-J., Eds.; Kluwer Academic Publishers: Boston, 1998.
(17) Blankenship, J. W.; Falk, J. E. Infinitely constrained optimization problems. *J. Optim. Theory Appl.* **1976**, *19*.
(18) Falk, J. E.; Hoffman, K. A nonconvex max−min problem. *Naval Res. Logistics Q.* **1977**, *24*, 441−450.
(19) Reemtsen, R.; Görner, S. In *Semi-Infinite Programming (Nonconvex Optimization and Its Applications)*; Reemtsen, R., Rückmann, J.-J., Eds.; Kluwer Academic Publishers: Boston, 1998.
(20) Zuhe, S.; Neumaier, A.; Eiermann, M. C. Solving Minimax Problems by Interval Methods. *BIT Numerical Mathematics* **1990**, *30*, 742−751.
(21) Bhattacharjee, B.; Green, W. H., Jr.; Barton, P. I. Interval methods for semi-infinite programs. *Comput. Optim. Appl.* **2005**, *30*, 63−93.
(22) McCormick, G. P. Computability of global solutions to factorable nonconvex programs: Part I—Convex underestimating problems. *Math. Program* **1976**, *10*, 147−175.
(23) Stein, O.; Still, G. Solving semi-infinite optimization problems with interior point techniques. *SIAM J. Control Optim.* **2003**, *42*, 769−788.
(24) Floudas, C. A.; Stein, O. The adaptive convexification algorithm: A feasible point method for semi-infinite programming. *SIAM J. Optim.* **2007**, *18*, 1187−1208.
(25) Adjiman, C. S.; Floudas, C. A. Rigorous convex underestimators for general twice-differentiable problems. *J. Global Optim.* **1996**, *9*, 23−40.

DOI: 10.1021/ie5029123
Ind. Eng. Chem. Res. 2015, 54, 307−317

(26) Mitsos, A.; Lemonidis, P.; Lee, C. K.; Barton, P. I. Relaxation-based bounds for semi-infinite programs. *SIAM J. Optim.* **2008**, *19*, 77−113.
(27) Mitsos, A. Global optimization of semi-infinite programs via restriction of the right-hand side. *Optimization* **2011**, *60*, 1291−1308.
(28) Scott, J. K.; Stuber, M. D.; Barton, P. I. Generalized McCormick relaxations. *J. Global Optim.* **2011**, *51*, 569−606.
(29) Stuber, M. D.; Scott, J. K.; Barton, P. I. Convex and concave relaxations of implicit functions. *Optim. Methods Software* **2014**, DOI: 10.1080/10556788.2014.924514.
(30) Hansen, E.; Walster, G. W. *Global Optimization Using Interval Analysis*, 2nd ed.; Marcel Dekker: New York, 2004.
(31) Hansen, E. R.; Greenberg, R. I. Interval Newton methods. *Appl. Math.Comput.* **1983**, *12*, 89−98.
(32) Bendtsen, C.; Stauning, O. *FADBAD, a Flexible C++ Package for Automatic Differentiation*; 1996.
(33) Griewank, A.; Walther, A. *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*, 2nd ed.; SIAM: Philadelphia, PA, 2008.
(34) Kearfott, R. B. Preconditioners for the Interval Gauss−Seidel Method. *SIAM J. Numer. Anal.* **1990**, *27*, 804−822.
(35) Moore, R. E. *Methods and Applications of Interval Analysis*; SIAM: Philadelphia, 1979.
(36) Yaws, C. L.; Narasimhan, P. K.; Gabbula, C. *Yaws' Handbook of Antoine Coefficients for Vapor Pressure*, 2nd electronic ed.; Knovel: New York, 2009.
(37) Kokossis, A. C.; Floudas, C. A. Synthesis of isothermal reactor−separator−recycle systems. *Chem. Eng. Sci.* **1991**, *46*, 1361−1383.
(38) Chachuat, B. *MC++: A Versatile Library for McCormick Relaxations and Taylor Models*. Available from http://www3.imperial.ac.uk/people/b.chachuat/research.
(39) Luksan, L.; Vlcek, J. Algorithms for non-differentiable optimization. *ACM Trans. Math. Software* **2001**, *27*, 193−213.

DOI: 10.1021/ie5029123
Ind. Eng. Chem. Res. 2015, 54, 307−317