



ELSEVIER

Available at  
**WWW.MATHEMATICSWEB.ORG**  
 POWERED BY SCIENCE @ DIRECT®

JOURNAL OF  
 COMPUTATIONAL AND  
 APPLIED MATHEMATICS

Journal of Computational and Applied Mathematics 156 (2003) 201–219

[www.elsevier.com/locate/cam](http://www.elsevier.com/locate/cam)

# Numerical solution for bounding feasible point sets

Peiliang Xu

*Disaster Prevention Research Institute, Kyoto University, Uji, Kyoto 611-0011, Japan*

Received 14 April 2002

## Abstract

Finding feasible points is important in optimization. There are currently two major classes of algorithms to deal with the problem of feasible points. The first class of algorithms (of local nature) is to find an approximate feasible point. Given a neighbourhood of an approximate feasible point, the second class of algorithms is to prove whether a feasible point exists inside this neighbourhood. To the best of our knowledge, no methods have been practically implemented to efficiently find the smallest boxes for bounding the feasible points defined by a system of nonlinear and nonconvex inequalities, unless the feasible set is convex. In this paper, we will present a numerical method to find the smallest boxes for bounding the feasible point sets defined by a nonlinear and nonconvex inequality and/or a system of nonlinear and nonconvex inequalities. Two examples have been synthetically constructed and used to show that the proposed numerical method can indeed correctly find all the smallest bounding boxes at any given accuracy efficiently. A brief comparison with relevant techniques will be discussed. Our method may also be thought of as the first solid theoretical basis for multisection and multisplitting in global optimization, when compared with those empirical ones in the literature.

© 2003 Elsevier Science B.V. All rights reserved.

**Keywords:** Feasible point set; Interval mathematics; Multisection; Multisplitting

## 1. Introduction

Consider the following system of equalities and inequalities:

$$g_i(\mathbf{x}) = 0, \quad i \in \mathbb{E}, \quad (1a)$$

$$g_i(\mathbf{x}) \leq 0, \quad i \in \mathbb{I}, \quad (1b)$$

where all the  $g_i(\mathbf{x})$  map  $\mathcal{R}^n$  into  $\mathcal{R}$  or parts of  $\mathcal{R}$ , the index sets  $\mathbb{E}$  for the equality constraints and  $\mathbb{I}$  for the inequality constraints satisfy  $\mathbb{E} \cup \mathbb{I} = \{1, 2, \dots, m\}$  and  $\mathbb{E} \cap \mathbb{I} = \emptyset$ . An equality can, in principle,

*E-mail address:* [pxu@rcep.dpri.kyoto-u.ac.jp](mailto:pxu@rcep.dpri.kyoto-u.ac.jp) (P. Xu).

be eliminated by representing one of the components of  $\mathbf{x}$  with the other  $(n - 1)$  free components. Alternatively, one may first use the method to be described in this paper to bound the feasible point set  $\mathbb{S}$  defined by the inequality system and then further to refine  $\mathbb{S}$  to satisfy the equality constraints. Thus we will confine ourselves to the inequality constraints (1b) in this paper. The vector form of (1b) is

$$\mathbf{g}(\mathbf{x}) \leq \mathbf{0}. \quad (2)$$

Problem (2) has been encountered often in science and engineering. Solving the inequality system (2) is of great mathematical interest by itself, which is as important as solving a system of equations (see e.g., [26,29,5,6]). Solving the inequality system (2) is almost always demanded, either explicitly or implicitly, in optimization and design (see e.g., [1]). As an example of optimization with great importance, we mention inverse problems, which are to find a quality solution by minimizing some measure of cost between the model and the measured data (see e.g., [30,11]). A nonlinear nonconvex inverse problem can also be reformulated as a constrained optimization model in order to obtain the global optimal solution(s) [34].

If a point  $\mathbf{x}$  satisfies all the inequality constraints of (2), it is said to be feasible; otherwise, it is said to be infeasible. By the solution to (2), we mean the set of all the feasible points that are implicitly determined by (2). If no feasible points satisfy (2), we say that problem (2) has no solution. Finding the set of feasible points is of fundamental importance in global optimization and provides the unique guarantee that the global solution has been or can be found. If all the functions  $g_i(\mathbf{x})$  are linear, then bounding the feasible point set of (2) can be exactly derived using interval analysis (see e.g., [25,9]). Unfortunately, many inverse problems, optimization and design in science and engineering are nonlinear and nonconvex. If some or all of the functions  $g_i(\mathbf{x})$  are nonlinear and nonconvex, there exists no efficient method or technique to correctly identify the feasible point set defined by nonlinear nonconvex constraints. For the methods of handling constraints in global optimization, the reader is referred to a recent review by Kearfott [19].

In this paper we assume that all the  $g_i(\mathbf{x})$  in (2) are nonlinear and nonconvex. There are a number of methods that are concerned with the feasible points of (2), which can be summarized as follows: (i) gradient or gradient-based methods (see e.g., [26,27,29,5,28]); (ii) trust-region and/or penalty function algorithms (see e.g., [6,28]); and (iii) interval analysis (see e.g., [10,9,21,12,20,16,17]) and approximation methods (see e.g., [15,31,14]). None of these methods can guarantee to correctly find or accurately bound the feasible point set of (2) efficiently, unless it is convex. The first two classes of methods are only to find a feasible point of (2). The gradient or gradient-based methods such as Newton and Gauss–Newton methods may be successfully used to find a feasible point if a starting point is sufficiently close to a region of the feasible points (see e.g., [29,5]), as in the case of using these methods in optimization. Because these methods are local in nature, they will generally not be able to find one feasible point in each region of the feasible points if the inequality system (2) gives rise to more than one region of the feasible points. If the feasible point set is convex, then a feasible point can be found from a sufficiently approximate starting point (see e.g., [29,5,2]). The trust-region algorithms were recently proposed in [6]. The basic idea is to recast the problem of solving (2) as a sequential trust-region optimization model. Although the trust-region methods could be used to find a feasible point, they may fail in some cases [6]. There is also no guarantee that they could find at least one feasible point in each of the disconnected regions of feasible points. More methods of these kinds for finding a feasible point can be found in [28].

Interval analysis may be used either to find a feasible point of (2) (see e.g., [12]) or to bound the feasible point set of (2) [10,24,9,21,33,19,16,17]. It was first proposed by Hansen and Sengupta [10] as a component of their optimization algorithm. The purpose of the method does not aim at finding a feasible point of (2), but instead, is to eliminate those certainly infeasible points of (2). The method may not work well if a given bounding parallelepiped box  $\mathbb{X}$  of  $\mathbf{x}$  is sufficiently large. In order to meet the need in engineering design, Kristinsdottir et al. [21] also discussed the problem of bounding the feasible points. By assuming a (given) feasible point, they then investigate two issues: (i) whether its  $\delta$  rectangular neighbourhood is also feasible; and (ii) find its maximum rectangular feasible neighbourhood. Obviously, their assumption is not acceptable in our case. Even worse is that a feasible region can be reported by their algorithm to be infeasible [21]. Kearfott [20] assumed an approximate feasible point and a small  $\delta$  neighbourhood around this point, and then used the interval Newton method to check whether a feasible point exists inside this small neighbourhood. Recently, Jaulin [16] and Jaulin et al. [17] also proposed an interval method to find the smallest box to bound the (connected or disconnected) feasible point region(s). This method is not generally applicable, because it depends on two rather restrictive assumptions: (i) for any  $g_i(\mathbf{x}) = y$  of (2), Jaulin [16] assumed that the inverse function for each component of  $\mathbf{x}$  is explicitly obtainable, namely,  $x_j = g_{ij}(x_1, x_2, \dots, x_{j-1}, x_{j+1}, \dots, x_n, y)$  for all  $j$ . If  $g_i(\mathbf{x})$  is nonlinear and nonconvex, it is generally impossible to satisfy this assumption; and (ii) Jaulin [16] further assumed that  $g_{ij}(\mathbb{X}_1, \mathbb{X}_2, \dots, \mathbb{X}_{j-1}, \mathbb{X}_{j+1}, \dots, \mathbb{X}_n, \mathbb{Y})$  can produce the sharpest (or minimal) interval, where  $\mathbb{X}_i$  ( $i = 1, 2, \dots, n$ ) is the interval of  $x_i$ , and  $\mathbb{Y}$  is that of  $y$ . Obviously, this second assumption is valid only for a very limited class of functions, monotonic and/or convex (see e.g., [9]), for example. If the second restrictive assumption is not satisfied, the bounding box found is not the smallest either, and may not make too much sense if the feasible point set consists of, say, two small subsets remotely separated. Neumaier [24] and Wolfe [33] used interval analysis to bound the feasible point set of nonlinear equality constraints by linearizing the nonlinear constraints, as is exactly what Hansen and Sengupta [10] first did with nonlinear inequality constraints. We would like to point out that our method can also be modified to find the bounding box(es) or enclosure for equality constraints, however. In the one-dimensional case, Xu [34] recast the nonlinear and nonconvex (2) as the problem of finding zeros of a function. It has been shown with examples that all the (disconnected) intervals of feasible points can be exactly obtained.

Approximation methods can also be used to correctly find the smallest box to bound the feasible point set, if the functions that define the feasible point set have some special features (convexity, for example) such that an affine function to cut off part of certainly infeasible points can be readily constructed (see e.g., [15,31,14]). This type of methods suffers two major drawbacks: (i) they depend on some assumptions which may fail to hold. If the feasible point set is defined by a set of nonlinear nonconvex inequalities, then the construction of an affine function can be as hard as the original feasibility problem (if not completely impossible). The convergence of these methods also require some conditions on the functions which may not hold generally (see e.g., [31]); and (ii) they may be quite inefficient computationally, since, at each iteration, a new affine function is constructed and added to the feasibility problem. Thus, approximation methods are either not generally applicable and/or may be computationally inefficient due to the growth of the feasibility problem with iterations.

Since the first two classes of methods cannot be used to find or bound the feasible point set of (2), they will not be investigated further in this paper. Although the method proposed in [10] (see also [9]) can be directly used for bounding the feasible point set defined by a set of nonlinear

nonconvex inequalities, it would often fail to reduce the original size of a given bounding box  $\mathbb{X}$  if  $\mathbb{X}$  is sufficiently large, as will become clear in Sections 2 and 4. The interval arithmetic method of Hansen [9] has already been implemented practically in [20], if an approximate feasible point and a small  $\varepsilon$  neighbourhood around it are further assumed. However, the main purpose of Kearfott [20] is not to find an accurate bounding box for the feasible points, but instead, is to check whether there exists a feasible point inside this  $\varepsilon$  neighbourhood. Without prior knowledge about the feasible point set, finding a good approximate feasible point may be as difficult as finding a feasible point itself.

The purpose of the paper is to propose a numerical method for finding the smallest bounding box(es) to bound the (disconnected) region(s) of the feasible points defined by the nonlinear and nonconvex inequality system (2). By a smallest bounding box  $\mathbb{S}$ , we mean that given a hyperplane  $x_i = c_i$ , if it cuts through  $\mathbb{S}$ , then it must intersect the equation of  $g(\mathbf{x}) = 0$  at least once inside  $\mathbb{S}$ . The paper is organized in the following. Section 2 will outline and briefly remark the method in [10,9] for later numerical comparisons. Since it is mainly proposed as a component of interval-based optimization algorithms, emphasis has been placed on efficiency but not on accuracy to bound the feasible point set [9]. In order to bound the feasible point set of (2) as accurately as possible, we will then modify the method of Hansen and Sengupta [10] for better performance, which is essentially equivalent to Neumaier [24] and Wolfe [33] in dealing with equality constraints. One may wonder why we will compare our method with the modified version of Hansen and Sengupta [10], because, among all the methods that may be used to find the bounding box(es) of feasible points, only Hansen and Sengupta [10] and its modified versions are generally applicable and can produce all the smallest bounding boxes of feasible points if one would have unlimited computing resource at his disposal. Section 3 is to present our numerical method for finding the smallest bounding box(es) to bound all the disconnected regions of the feasible points of (2). We will first discuss the case of one nonlinear and nonconvex inequality and then extend the results to the general case of systems of nonlinear and nonconvex inequalities in a natural manner. Section 4 will serve two purposes: (i) to demonstrate how the numerical method works with synthetic examples; and (ii) to compare it numerically with the modified versions of Hansen and Sengupta [10].

## 2. Bounding the feasible point set based on Taylor expansion

### 2.1. The Hansen–Sengupta’s method

Consider first the problem of one nonlinear inequality, namely,

$$g(\mathbf{x}) \leq 0, \quad \mathbf{x} \in \mathbb{X}, \quad (3)$$

where  $\mathbb{X}$  is a parallelepiped box with each component of  $\mathbf{x}$  bounded by  $\underline{x}_i$  from below and by  $\bar{x}_i$  from above. Here  $\underline{x}_i$  and  $\bar{x}_i$  are pre-determined. Applying the mean value theorem to the function  $g(\mathbf{x})$ , we have

$$g(\mathbf{x}) = g(\mathbf{y}) + (\mathbf{x} - \mathbf{y})^T \nabla g(\xi),$$

where  $\nabla g(\xi)$  is the vector of the first-order partial derivatives of the function  $g(\mathbf{x})$  at the point  $\xi \in \mathbb{X}$ . Since  $\nabla g(\xi) \in \nabla g(\mathbb{X})$ , (3) can be rewritten as

$$g(\mathbf{y}) + (\mathbf{x} - \mathbf{y})^T \nabla g(\mathbb{X}) \leq 0, \quad \mathbf{x} \in \mathbb{X}, \quad (4)$$

where  $\nabla g(\mathbb{X})$  is the interval vector of the first-order derivatives of  $g(\mathbf{x})$  computed within the box  $\mathbb{X}$ . Because use of (4) may run the risk of losing the feasible points of (3), Hansen [9] proposes to use

$$g(\mathbf{y}) + (\mathbf{x} - \mathbf{y})^T \nabla g(\mathbb{X}) > 0, \quad \mathbf{x} \in \mathbb{X}, \quad (5)$$

in order to reduce the size of  $\mathbb{X}$  or to bound the feasible point set. Since all the points  $\mathbf{x}$  that satisfy (5) are certainly not feasible, no feasible points of the original problem (3) will be eliminated by deleting the solutions to (5) from  $\mathbb{X}$  [9].

Since the linear interval inequality (5) is involved with  $n$  variables, we cannot solve it for the certainly infeasible points of (3) simultaneously for all the components of  $\mathbf{x}$  and delete them from  $\mathbb{X}$ . We have to start with one of the components to branch, say  $x_i$  of  $\mathbf{x}$  with the longest length of side or the largest difference between the upper and lower bounds. Thus we can rewrite (5) as follows:

$$g(\mathbf{y}) + \sum_{j=1, j \neq i}^n (\mathbb{X}_j - y_j) g'_j(\mathbb{X}) + (x_i - y_i) g'_i(\mathbb{X}) > 0. \quad (6a)$$

Since all the derivatives in (6a) are computed with  $\mathbb{X}$ , their interval widths may be too large. In order to sharpen the interval bounds for the derivatives, we can also use the following alternative inequality:

$$\begin{aligned} g(\mathbf{y}) + \sum_{j=1, j \neq i}^n (\mathbb{X}_j - y_j) g'_j(\mathbb{X}_1, \mathbb{X}_2, \dots, \mathbb{X}_j, y_{j+1}, \dots, y_n) \\ + (x_i - y_i) g'_i(\mathbb{X}_1, \mathbb{X}_2, \dots, \mathbb{X}_i, y_{i+1}, \dots, y_n) > 0, \end{aligned} \quad (6b)$$

(see [9]). (6) can be symbolically rewritten as follows:

$$\mathbb{U} + \mathbb{V}t > 0, \quad (7)$$

where  $g'_k(\cdot)$  is the derivative interval of  $g(\mathbf{x})$  with respect to  $x_k$  ( $k = 1, 2, \dots, n$ ),

$$\mathbb{U} = g(\mathbf{y}) + \sum_{j=1, j \neq i}^n (\mathbb{X}_j - y_j) g'_j(\mathbb{X}) = [\underline{u}, \bar{u}],$$

$$\mathbb{V} = g'_i(\mathbb{X}) = [\underline{v}, \bar{v}]$$

for the case of (6a), and

$$\mathbb{U} = g(\mathbf{y}) + \sum_{j=1, j \neq i}^n (\mathbb{X}_j - y_j) g'_j(\mathbb{X}_1, \mathbb{X}_2, \dots, \mathbb{X}_j, y_{j+1}, \dots, y_n),$$

$$\mathbb{V} = g'_i(\mathbb{X}_1, \mathbb{X}_2, \dots, \mathbb{X}_i, y_{i+1}, \dots, y_n)$$

for the case of (6b), and

$$t = x_i - y_i.$$

Let the solution set of  $t$  in (7) be denoted by  $\mathbb{T}_c$ , namely,

$$\mathbb{T}_c = \{t: u + vt > 0, u \in \mathbb{U}, v \in \mathbb{V}\}.$$

By direct analysis of (7) using the interval division rules (see e.g., [23,28,25,9]), we obtain

$$\mathbb{T}_c = \begin{cases} [-\underline{u}/\bar{v}, \infty) & \text{if } \underline{u} > 0, \underline{v} \geq 0 \text{ and } \bar{v} > 0, \\ [-\underline{u}/\underline{v}, \infty) & \text{if } \underline{u} \leq 0 \text{ and } \underline{v} > 0, \\ (-\infty, -\underline{u}/\bar{v}] & \text{if } \underline{u} \leq 0 \text{ and } \bar{v} < 0, \\ (-\infty, -\underline{u}/\underline{v}] & \text{if } \underline{u} > 0, \underline{v} < 0 \text{ and } \bar{v} \leq 0, \\ [-\underline{u}/\bar{v}, -\underline{u}/\underline{v}] & \text{if } \underline{u} > 0 \text{ and } \underline{v} < 0 < \bar{v}, \\ (-\infty, \infty) & \text{if } \underline{u} > 0 \text{ and } \underline{v} = \bar{v} = 0, \\ \text{empty set} & \text{if } \underline{u} \leq 0 \text{ and } \underline{v} \leq 0 \leq \bar{v}. \end{cases} \quad (8)$$

It is interesting to note that the solution  $\mathbb{T}_c$  is independent of the upper bound of  $\mathbb{U}$ ; this is nevertheless not obvious either in its original form (7) or from the definition of  $\mathbb{T}_c$ . We can now compute the complement set of  $\mathbb{T}_c$ , denoted by  $\mathbb{T}$ , as follows:

$$\begin{aligned} \mathbb{T} &= \{t: t \in (-\infty, \infty) \text{ and } t \notin \mathbb{T}_c\} \\ &= \begin{cases} (-\infty, -\underline{u}/\bar{v}] & \text{if } \underline{u} > 0, \underline{v} \geq 0 \text{ and } \bar{v} > 0, \\ (-\infty, -\underline{u}/\underline{v}] & \text{if } \underline{u} \leq 0 \text{ and } \underline{v} > 0, \\ [-\underline{u}/\underline{v}, \infty) & \text{if } \underline{u} > 0, \underline{v} < 0 \text{ and } \bar{v} \leq 0, \\ [-\underline{u}/\bar{v}, \infty) & \text{if } \underline{u} \leq 0 \text{ and } \bar{v} < 0, \\ (-\infty, -\underline{u}/\bar{v}] \cup [-\underline{u}/\underline{v}, \infty) & \text{if } \underline{u} > 0 \text{ and } \underline{v} < 0 < \bar{v}, \\ (-\infty, \infty) & \text{if } \underline{u} \leq 0 \text{ and } \underline{v} \leq 0 \leq \bar{v}, \\ \text{empty set} & \text{if } \underline{u} > 0 \text{ and } \underline{v} = \bar{v} = 0 \end{cases} \quad (9) \end{aligned}$$

(see also [10,9]). It should be pointed out that although all the points in  $\mathbb{T}_c$  are certainly infeasible, this does not mean that all the points in  $\mathbb{T}$  are feasible. In fact, as the complement set of  $\mathbb{T}_c$ ,  $\mathbb{T}$  will generally contain many (but certainly not all) infeasible points satisfying (7).

With the solution set  $\mathbb{T}$  of (9), we can then compute the reduced point set(s) of  $\mathbb{X}$ . The new interval of  $\mathbb{X}$  along the coordinate axis  $x_i$  is given as follows:

$$\mathbb{X}_i^n = \mathbb{X}_i \cap (\mathbb{T} + y_i), \quad (10)$$

where  $\mathbb{X}_i^n$  is the new interval for the component  $x_i$ . If  $\mathbb{T}$  consists of two parts, say  $\mathbb{T}_{i1}$  and  $\mathbb{T}_{i2}$  (compare the fifth row on the right-hand side of (9)), we then fathom  $\mathbb{X}$  into two disconnected boxes, which are computed by replacing  $\mathbb{T}$  with the respective intervals, namely,

$$\mathbb{X}_{i1}^n = \mathbb{X}_i \cap (\mathbb{T}_{i1} + y_i), \quad (11a)$$

$$\mathbb{X}_{i2}^n = \mathbb{X}_i \cap (\mathbb{T}_{i2} + y_i). \quad (11b)$$

The procedure to eliminate certainly infeasible points in this section is repeated for  $i = 1, 2, \dots, n$  and is supposed to result in  $m$  subboxes. Choose one of these boxes of reduced size, say the one with the longest side. Replace  $\mathbb{X}$  with this new subbox and then repeat the procedure to further eliminate certainly infeasible points.

To deal with the nonlinear inequality system (2), we first use Taylor series or similar expansion to derive the inequality of type (4) for each inequality constraint, and then collect them together to form the following interval linear system of inequalities:

$$\mathbb{A} \mathbf{t} > -\mathbf{g}(\mathbf{y}), \quad (12)$$

(see e.g., [9]), where  $\mathbb{A}$  is the interval matrix of  $m$  rows and  $n$  columns, with each element  $\mathbb{A}_{ij}$  being either equal to  $g'_{ij}(\mathbb{X})$  in the case of (6a) or  $g'_{ij}(\mathbb{X}_1, \mathbb{X}_2, \dots, \mathbb{X}_j, x_{j+1}, \dots, x_n)$  in the case of (6b). Here the interval  $g'_{ij}(\mathbb{S})$  stands for the derivative of the  $i$ th component of  $\mathbf{g}(\mathbf{x})$  with respect to  $x_j$  and is computed with the point set  $\mathbb{S}$ .

For the interval linear system of inequalities (12), we do not have the same explicit and elegant result as (9). In order to solve (12) without undue growth of intervals, Hansen and Sengupta [10] and Hansen [9] proposed applying preconditioning to  $\mathbb{A}$  first and then applying (9) to each derived linear inequality in order to eliminate the certainly infeasible points of (12). A preconditioning matrix  $\mathbf{B}$  may be obtained by inverting the centered matrix of  $\mathbb{A}$ . The matrix  $\mathbf{B}$  generally contains negative numbers. If it is left-multiplied to (12), the negative numbers of  $\mathbf{B}$  will result in uncertainty in the transformed inequality system, in the sense that we do not know any longer whether  $<$  or  $>$  should be applied to each of the transformed inequalities. It is thus a mandate to keep the present signs of inequalities, which, in turn, would only permit to derive a positive  $\mathbf{B}$  partially. Because of the preconditioning, the number of linear inequalities will increase by up to  $m$ —the same number of the original inequality system. For more technical details, the reader is referred to Hansen and Sengupta [10] and Hansen [9].

## 2.2. Modifying the Hansen–Sengupta’s method

For the nonlinear nonconvex inequality (3), if the given box  $\mathbb{X}$  is sufficiently large, then we would often encounter the situation that in (7),  $\underline{u} < 0 < \bar{u}$  and  $\underline{v} < 0 < \bar{v}$ . By comparing this scenario with the conditions of (9), we immediately conclude that  $\mathbb{T} = (-\infty, \infty)$ . This clearly indicates that the Hansen–Sengupta’s method generally does not eliminate any certainly infeasible points and thus cannot reduce the size of  $\mathbb{X}$  if the initial box is sufficiently large. The same can be said of the system of nonlinear and nonconvex inequalities (2).

It should become clear now that in order for the Hansen–Sengupta’s method to eliminate some certainly infeasible points, the given box  $\mathbb{X}$  to initialize the procedure described in the previous subsection has to be sufficiently small. Actually, the success of implementing the Hansen–Sengupta’s method numerically in [20] is exactly based on this assumption. Unfortunately, in practice, we often have some inequality constraints only, without any prior knowledge about the where-about of the feasible points. Thus the assumption of a sufficiently large initial box  $\mathbb{X}$  should be very reasonable.

In order to use the only method in the literature that has the potential of general applicability to bound the feasible points of (3), we have to iteratively bisect the starting box  $\mathbb{X}$  when the Hansen–Sengupta’s method fails to improve it. We can now assemble the Hansen–Sengupta’s method and



bisection to find the smallest possible boxes for bounding the feasible point set of (3). The algorithm is described as follows:

#### Algorithm I.

1. Given an initial box  $\mathbb{X}$ , initialize a problem list and assign  $\mathbb{X}$  to the list;
2. If the problem list is empty, terminate. Get a box  $\mathbb{Y}$  from the problem list. If the size of the box is smaller than a pre-determined  $\epsilon$ , store it as part of the solution and repeat this step;
3. Compute  $g(\mathbb{Y})$ . If  $g(\mathbb{Y}) \leq 0$  (or  $g(\mathbb{Y}) > 0$ ), then store  $\mathbb{Y}$  as part of the solution and rearrange the solution box (or delete  $\mathbb{Y}$ ) and go to Step 2;
4. Select one of the components of  $\mathbf{x}$ , say  $x_n$ , to start applying the Hansen–Sengupta’s method;
5. Compute  $\mathbb{T}$  of (9) with  $\mathbb{Y}$ . If  $\mathbb{T}$  improves reducing the size of  $\mathbb{Y}$ , and if  $\mathbb{T}$  is a single interval, replace  $\mathbb{Y}$  with the new one, and go to Step 3; otherwise, put one of the intervals into the problem list and use the other as  $\mathbb{Y}$ , and go to Step 3;
6. If  $\mathbb{T}$  fails to improve  $\mathbb{Y}$  and if not all the components have been checked, then select a new component and repeat applying the Hansen–Sengupta’s method (Step 4);
7. Bisect  $\mathbb{Y}$  into  $\mathbb{Y}_1$  and  $\mathbb{Y}_2$ . The simplest method is to bisect the side with the largest width (or length). Compute  $g(\mathbb{Y}_1)$  and  $g(\mathbb{Y}_2)$ . If both of the subboxes are infeasible, delete them and go to step 2; if one of them is infeasible, delete it. Replace  $\mathbb{Y}$  with the remaining half box and go to Step 5. If both of  $\mathbb{Y}_1$  and  $\mathbb{Y}_2$  are not infeasible, store one of them (say  $\mathbb{Y}_1$ ) into the problem list, replace  $\mathbb{Y}$  with the other, and then go to Step 3.

Compared with the original version of the Hansen–Sengupta’s method, this modified algorithm has brought in a new step of bisection. This step can be significant and crucial, since the Hansen–Sengupta’s method would probably not improve at all, if the initial box  $\mathbb{X}$  is sufficiently large and/or if the nonlinear inequality is highly oscillatory. Actually, this combination of linear Taylor expansion and bisection has been successfully applied in [24,33] to find the enclosure of solutions to nonlinear equations  $\mathbf{f}(\mathbf{x}) = \mathbf{0}$ , where  $\mathbf{f}$  is an  $m$ -dimensional vector of functions.

To solve the multiple inequality constraints (2), we can slightly modify Algorithm I. Actually, we can simply add the loop for each inequality to Algorithm I to deal with (2), as proposed in [10]. Additionally, we should slightly modify Step 3 of Algorithm I such that all  $g_i(\mathbb{Y})$  are computed and checked, since violation of any of these inequalities has already indicated that  $\mathbb{Y}$  is an infeasible point set. At the first steps of applying this modified algorithm, the preconditioning procedure will be probably not needed, since (12) would likely not result in any improvement on reducing the size of  $\mathbb{Y}$ .

### 3. The numerical method

In order to motivate our numerical method and have an impression on the performance of the Hansen–Sengupta’s method, let us start this section with a small illustrative example as follows:

$$g(x, y) = -10 \exp(-|x| - |y|) + \sin(xy) + 5 \quad (13a)$$



and

$$\mathbb{X} = \left\{ \begin{array}{c} [-10^8, 10^8] \\ [-10^8, 10^8] \end{array} \right\}. \quad (13b)$$

A rough estimate shows that the feasible points of the nonlinear inequality  $g(x, y) \leq 0$  must be inside the small box:

$$\mathbb{S} = \left\{ \begin{array}{c} [-1, 1] \\ [-1, 1] \end{array} \right\}, \quad (13c)$$

although the initial box  $\mathbb{X}$  is very large. A quick interval computation also shows that the range of the gradients of  $g(x, y)$  in  $\mathbb{X}$  is given by  $g'(\mathbb{X}) \approx \mathbb{X}$ . Even if the supposedly tighter expansion (6b) is used, the best possible ranges of the gradients we can expect are:

$$g'_x([-10^8, 10^8], 0) = g'_y(0, [-10^8, 10^8]) = [-10, 10].$$

It is trivial to show that the Hansen–Sengupta’s method cannot delete any certainly infeasible points from  $\mathbb{X}$ , although the smallest box to bound the feasible points should even be smaller than  $\mathbb{S}$ .

The consequences of Taylor-expanding the nonlinear inequality (3) obviously include: (i) that all the partial derivatives of first-order  $g'_i(\mathbf{x})$  ( $i=1, 2, \dots, n$ ) within a sufficiently large box  $\mathbb{X}$  are expected to almost always satisfy  $\underline{g'_i}(\mathbb{X}) < 0$  and  $\overline{g'_i}(\mathbb{X}) > 0$  for all  $i$ , where  $\underline{g'_i}(\mathbb{X})$  and  $\overline{g'_i}(\mathbb{X})$  are the lower and upper bounds of  $g'_i(\mathbb{X})$ , respectively; (ii) that the Taylor expansion will almost always substantially overestimate the range of  $g(\mathbf{x})$  in  $\mathbb{X}$  such that  $\underline{g}(\mathbb{X}) \leq 0$  and  $\overline{g}(\mathbb{X}) \geq 0$  if the box  $\mathbb{X}$  is sufficiently large. This should be immediately clear from (6), since the range of  $g(\mathbf{x})$  is extrapolated through the sum of  $\underline{g'_j}(\mathbb{X})(\mathbb{X}_j - y_j)$  ( $y_j \in \mathbb{X}_j$ ) in the case of (6a) or  $\underline{g'_j}(\mathbb{X}_1, \mathbb{X}_2, \dots, \mathbb{X}_j, y_{j+1}, \dots, y_n)(\mathbb{X}_j - y_j)$  in the case of (6b); and (iii) that as a result of (i) and (ii), we should generally expect that in (4),  $\underline{u} < 0$ ,  $\overline{u} > 0$ ,  $\underline{v} < 0$  and  $\overline{v} > 0$ , no matter what value of  $g(\mathbf{y})$  may take on. Thus for a nonlinear inequality in a sufficiently large box  $\mathbb{X}$ , the Hansen–Sengupta’s method will almost always fail to produce a smaller box to bound the feasible points.

Example (13) has clearly shown that the Taylor-series or similar expansion generally does not work well in eliminating the certainly infeasible points of a nonlinear nonconvex inequality constraint in a sufficiently large box. Thus we propose directly applying a numerical method to

$$g(\mathbf{x}) > 0, \quad \mathbf{x} \in \mathbb{X} \quad (14)$$

and obtain a set of certainly infeasible points. By eliminating this set of infeasible points from  $\mathbb{X}$ , we can then obtain a smaller box to bound the feasible points.

In order to find the solution to (14), namely, the smallest bounding box(es) to bound the feasible points of (14), we propose the following two-step numerical method: (i) quick estimate of the feasible point set of (14); and (ii) confirming and/or refining the results from the first step. The task of quickly estimating the solution of (14) can be recast as the problem of solving the one-dimensional interval inequality:

$$f(\mathbf{a}, t) > 0, \quad (15)$$

where  $\mathbf{a} \in \mathbb{A}$ ,  $t \in \mathbb{T}$ ,  $\mathbb{A}$  is a given interval vector and  $\mathbb{T}$  a given interval. For a nonlinear function  $f(\mathbf{a}, t)$ ,  $t$  generally intermingles with  $\mathbf{a}$ . Thus finding the exact solution to (15) may be difficult.

However, we can further simplify (15) significantly by replacing some or all of the intermingled (mixed) terms of  $\mathbf{x}$  such as  $\sin(x_i x_j)$  with their proper bounds, in particular, those mixed terms being small in  $\mathbb{X}$  but highly oscillatory (if any). We can then *quickly* solve the simplified one-dimensional inequality of type (15) numerically using the one-dimensional equation solver [9] or equivalently, the one-dimensional feasible point finder [34]. For the multivariate inequality (14), by treating one of the components  $x_i$  as  $t$  and the rest as  $\mathbf{a}$ , we can then solve (14) for all the components of  $\mathbf{x}$ . Using the tightened bounding box(es), we can then iteratively solve (14) until no improvement is possible. We will refer this procedure to eliminate those certainly infeasible points as the *quick solution approach* in the rest of this paper.

Similar approaches to the quick solution approach have been used in [13,32] to find the enclosures of zeros of a polynomial system. Hong and Stahl [13] proposed to first find the lower and upper bounding functions of  $f(\mathbf{a}, t)$ , and then use them to narrow the searching space or box. If  $g(\mathbf{x})$  is polynomial, finding the lower and upper bounding functions of  $g(\mathbb{X}_1, \mathbb{X}_2, \dots, \mathbb{X}_{i-1}, x_i, \mathbb{X}_{i+1}, \dots, \mathbb{X}_n)$  ( $i = 1, 2, \dots, n$ ) is rather straightforward without difficulty, as in [13]. For a general nonlinear non-convex function  $g(\mathbf{x})$ , it would be quite difficult to find the corresponding lower and upper bounding functions. One way to do so is to use the Taylor interval linear approximation, which is essentially equivalent to the Hansen–Sengupta’s method. The difference between our quick solution approach and tightening or narrowing of Hong and Stahl [13] and van Hentenryck et al. [32] is threefold: (i)  $f(\mathbf{a}, t)$  in this paper is not necessarily the upper bounding function of  $g(\mathbb{X}_1, \mathbb{X}_2, \dots, \mathbb{X}_{i-1}, x_i, \mathbb{X}_{i+1}, \dots, \mathbb{X}_n)$  for a given variable  $x_i$ , though they are related; (ii) after replacing all but one variable (say  $x_i$ ) with their corresponding intervals, we are still free to replace some of  $x_i$  in the derived univariate function  $g(\mathbb{X}_1, \mathbb{X}_2, \dots, \mathbb{X}_{i-1}, x_i, \mathbb{X}_{i+1}, \dots, \mathbb{X}_n)$  with  $\mathbb{X}_i$  in order to quickly find a function of type (15) for eliminating some certainly infeasible points. Since the derived function of type (15) is not lower-bounded, it can be more efficient, and (iii) we use the quick solution approach to the new problem of finding the smallest bounding boxes of feasible points, while Hong and Stahl [13] and van Hentenryck et al. [32] were concerned with solving the enclosures of zeros of a polynomial system.

We use example (13) to demonstrate the quick solution approach. By treating  $x$  as  $a$  and  $y$  as  $t$  in (13), and simplifying the intermingled term  $\sin(xy)$  over  $\mathbb{X}$  as  $[-1, 1]$ , we obtain the one-dimensional interval inequality of type (15) as follows:

$$-10[0, 1] \exp(-|y|) + [-1, 1] + 5 > 0 \quad (16)$$

the solution of which can be readily found to be  $|y| \in [0.9163, 10^8]$ . In the same manner, we can find the infeasible point set of (13) for the component  $x$ , namely,  $|x| \in [0.9163, 10^8]$ . Thus the bounding box for the feasible point set of (13) can be immediately obtained and given by

$$\mathbb{S} = \left\{ \begin{array}{l} [-0.9163, 0.9163] \\ [-0.9163, 0.9163] \end{array} \right\},$$

which is even much better than (13c). Iteration will further improve the above bounding box  $\mathbb{S}$ .

If there exist no intermingled terms in (14), the results obtained by repeating solving the interval inequality of type (15) will be final. Without loss of generality, we assume that there are intermingled terms among the components of  $\mathbf{x}$  in (14). In order to confirm and/or further improve the results from the first step, we propose the following three-component *recipe*: (i) slicing a given box  $\mathbb{Y}$  into a number of subboxes; (ii) using the original nonlinear function to compute the range of  $g(\mathbf{x})$  in a

given box  $\mathbb{Y}$ ; and (iii) deciding whether a box is feasible, not feasible or needs to be further sliced. Here the last two components should be clear by themselves. Nevertheless, we would like to note that unlike Hansen and Sengupta [10], Neumaier [24] and Wolfe [33] in the second component, we directly use the original nonlinear function but not its linear expansion to compute the range of  $g(\mathbf{x})$ .

We will now explain how to conduct multiple slicing in the first component. Usually, one would simply bisect a box  $\mathbb{Y}$  into two subboxes, often based on the length of a side, as seen in the literature (see e.g., [28,9,33]). The advantage of bisecting is twofold: (i) that the method is simple and easy to implement; and (ii) that it requires no other information about the function  $g(\mathbf{x})$ . Multiple slicing can be more efficient if the slicing is guided with the relevant information about  $g(\mathbf{x})$ . Some empirical multiple slicing methods have recently been proposed and investigated, for instance, in [9,3,4]. In this paper, we will use the one-dimensional feasible point finder in [34] as a solid theoretical basis to guide the multiple slicing or multisection. Without loss of generality, let us assign the  $(n-1)$  variables of  $g(\mathbf{x})$  (say  $x_1, x_2, \dots, x_{n-1}$ ) to some pre-determined values, namely,  $x_i = y_i$  ( $i = 1, 2, \dots, (n-1)$ ) and leave one free (say  $x_n \in \mathbb{Y}_n$ ), and assume that the equation

$$g(y_1, y_2, \dots, y_{n-1}, x_n) = 0 \quad (17)$$

has a number of solutions in  $\mathbb{Y}_n$ . Using the one-dimensional equation solver [9] or feasible point finder [34], we can then exactly separate the feasible and infeasible intervals of  $\mathbb{Y}_n$ . Thus we can readily slice  $\mathbb{Y}_n$  into a number of feasible and infeasible intervals accordingly. In the same manner, we can also multiply slice any component of  $\mathbf{x}$ . Different ways of multiple slicing may require significantly different computing times. The optimal way of multiple slicing is not known in advance, unfortunately. As a guide, we propose to choose the sliced subbox with the largest feasible interval from the previous slicing for further multiple slicing in order to quickly obtain a largest possible bounding subbox for the feasible points. A large subbox of the feasible points may also avoid frequent multiple slicing on the same component in the neighbouring sliced subboxes. If Eq. (17) has no solution for any free component  $x_i$ , then there can be two possibilities: (i)  $\mathbb{Y}$  may not be infeasible and we have to bisect  $\mathbb{Y}$  into two subboxes for further check; or (ii)  $\mathbb{Y}$  is already the smallest bounding box currently under check.

We put the quick solution approach and this three-component *recipe* together to construct Algorithm II for finding the smallest boxes to bound feasible point sets, which is briefly described as follows:

#### Algorithm II.

1. Given an initial box  $\mathbb{X}$ , initialize a problem list and assign  $\mathbb{X}$  to the list;
2. Iteratively solve the interval inequality of type (15) for all the components of  $\mathbf{x}$  and put the resulted bounding box(es) into the problem list;
3. If the problem list is empty, terminate. Get a box  $\mathbb{Y}$  from the problem list. If the size of the box is smaller than a pre-determined  $\epsilon$ , repeat this step;
4. Compute  $g(\mathbb{Y})$ . If  $g(\mathbb{Y}) \leq 0$  (or  $g(\mathbb{Y}) > 0$ ), then store  $\mathbb{Y}$  as part of the solution, rearrange the solution boxes (or delete  $\mathbb{Y}$ ) and go to Step 3;
5. If all the components of  $\mathbb{Y}$  cannot be sliced any more, store  $\mathbb{Y}$  as part of the solution, rearrange the solution boxes and then go to Step 3;

6. If the subbox  $\mathbb{Y}$  contains feasible sides, use the *quick solution* approach, namely, by repeated solving (15) for all the components of  $\mathbf{x}$ , in order to eliminate certainly infeasible points from this subbox. If the number of the obtained boxes is larger than one, keep one of these boxes as a new  $\mathbb{Y}$ , store all the others into the problem list, and then go to Step 4. Otherwise, choose any component without a feasible side for further slicing. Then use (one or both of) the following strategies:
  - the one-dimensional feasible point finder; and/or
  - local optimizers for the implicit function of reduced dimension derived from the inequality, as formulated by (19), to grow the feasible intervals, and accordingly, readjust the sliced results (subboxes) along this component. Keep one of these boxes as a new  $\mathbb{Y}$ , store all the others into the problem list, and then go to Step 4;
7. Use the one-dimensional equation solver or feasible point finder to separate the feasible and infeasible intervals of, say  $\mathbb{Y}_n$ , given  $x_i = y_i$  ( $i = 1, 2, \dots, (n-1)$ ). If there exists no solution to (17) for any component  $x_i$  and if  $g(\mathbf{y}) < 0$ , store  $\mathbb{Y}$  as part of the solution, rearrange the solution box and then go to Step 3;
8. If  $g(\mathbf{y}) > 0$  in Step 7, test whether any part of  $x_i < y_i$  or  $x_i > y_i$  can be eliminated. If  $\mathbb{Y}$  has been tightened, replace  $\mathbb{Y}$  with the improved one and go to Step 4; If tightening is not possible, bisect  $\mathbb{Y}$  into two subboxes. Put one of them into the problem list and then go to Step 4;
9. Slice  $\mathbb{Y}$  into multiple subboxes along  $\mathbb{Y}_n$  according to the separation of feasible and infeasible intervals. Use the one-dimensional feasible point finder to grow the feasible intervals along  $\mathbb{Y}_n$  and accordingly, readjust the multiply sliced subboxes. Keep one of these boxes as a new  $\mathbb{Y}$ , store all the others into the problem list, and then go to Step 4.

Some further explanations on the second strategy in Step 6 of Algorithm II may be appropriate. For a sliced subbox without a feasible side, we can use the quick solution approach or bisection to eliminate infeasible points. Thus without loss of generality, we assume that the sliced subbox  $\mathbb{Y}$  has a feasible side, say on  $x_1$ . In order to make a largest possible bounding box out of this feasible interval, we only need to find the maximum/minimum values of all the components other than  $x_1$ , satisfying the equality constraint:

$$g(\mathbf{x}) = 0. \quad (18)$$

In other words, we have to solve the following optimization problems:

$$\max/\min x_i \quad (19)$$

subject to the equality constraint (18), where  $i \neq 1$ . Finding the maximum/minimum values of a component  $x_i$  ( $i \neq 1$ ) within  $\mathbb{Y}$  is a global optimization problem and can be time-consuming (compare e.g., [22,8,9,18,7]). Since one of the most important purposes of bounding feasible point sets is to develop new methods/algorithms of global optimization, and since a global solution within  $\mathbb{Y}$  may likely not be a global solution in the largest possible but not yet known bounding box(es) with  $\mathbb{Y}$  as its subset, we will not use global optimization techniques to solve (19). On the other hand, a global optimization algorithm may produce the global solutions for  $x_i$  that are separated and unfavourably far away from the current feasible region of interest. In fact, the optimization formulation (19) is not an essential component of Algorithm II but can be helpful in reducing the numbers of bisection

and multiple slicings. Thus a local optimal solution to (19) is sufficient for our purpose. The size of a bounding box can then be grown by iteration. If the function  $g(\mathbf{x})$  is highly oscillatory, we may randomly sample a point over the feasible interval and find the smallest and largest possible feasible values for the components other than  $x_1$  using the one-dimensional zero point finder [34] in order to obtain a bounding (sub)box for the feasible points. One of the reviewers brought the attention of this author to the recently published book in [17], in which (19) is directly proposed to find the smallest bounding box. Due to the reasons mentioned in the introduction, this method can only be used to some rather limited cases.

For the system of multiple nonlinear and nonconvex inequality constraints (2), we can add an outer loop for each inequality of (2) to Algorithm II, as in [10] or [9]. As in Algorithm I, we should also slightly modify Step 4 of Algorithm II such that all  $g_i(\mathbb{Y})$  are computed and checked. This treatment of multiple inequality constraints by simply adding an outer loop can be far from satisfactory. The bounding box(es) for the feasible points obtained in this manner can be too large. In order to further improve the solution to the system of multiple nonlinear constraints (2), we propose replacing (2) with the following single nonlinear inequality constraint:

$$\sum_{i=1}^m \alpha_i |g_i(\mathbf{x})| \leq \varepsilon \quad (20a)$$

or

$$\sum_{i=1}^m \alpha_i g_i^2(\mathbf{x}) \leq \varepsilon, \quad (20b)$$

where  $\mathbf{x} \in \mathbb{X}$ ,  $\varepsilon$  is a pre-determined small positive constant, and

$$\alpha_i = \begin{cases} 0 & \text{if } g_i(\mathbf{x}) \leq 0, \\ 1 & \text{otherwise.} \end{cases}$$

The equivalence between (2) and (20) can be readily established. In order to satisfy (20a), for instance, we must have

$$g_i(\mathbf{x}) \leq \varepsilon, \quad i = 1, 2, \dots, m \quad (21)$$

since all  $\alpha_i \geq 0$ . Letting  $\varepsilon$  tend to zero, we see that (21) is turned to nothing but (2).

The difference between simply adding an outer loop and the equivalent treatment (20) to deal with the multiple inequality constraints (2) is significant. By adding an outer loop, as suggested in [10], we can only handle one inequality constraint after another. Mathematically, this is equivalent to activating one constraint at each run of the loop but turning off the constraints imposed by the other  $(m - 1)$  inequalities. Actually, adding an outer loop to either Algorithm I or II to handle multiple inequality constraints is different from and should result in a tighter bounding box than the violating-all-the-constraints formulation (12) given in [10] (see also [9]), since it is trivial to prove that the Hansen–Sengupta’s method only results in a subset of the infeasible points by adding an outer loop. When the proposed equivalence (20a) or (20b) is used, all the inequality constraints of (2) have been activated. Thus (20) should result in the tighter bounding box(es) for the feasible points of (2) than adding an outer loop.

#### 4. Synthetic examples and comparisons

In this section, we will implement, demonstrate and compare three methods for bounding the feasible points of nonlinear inequality constraints (2): (i) the Hansen–Sengupta’s method; (ii) the modified Hansen–Sengupta’s method; and (iii) the numerical method. The latter two methods are first proposed in this paper. We will also compare the performances of Hansen–Sengupta’s treatment of a nonlinear inequality system, namely, one inequality after another by adding an outer loop to either Algorithm I or II, and our equivalent reformulation (20). The experiments to be reported in this section were carried out with Borland C++ on a Toshiba Notebook Tecra 8000 (RAM 128 MB, Pentium II 400 MHz).

We have simulated two examples. The first example is based on (13) but is modified to create more regions of feasible points. More specifically, we have synthetically added two exponential functions to (13) and thus created two more major regions of feasible points. The modified example is given as follows:

$$\begin{aligned} g(x, y) = & -10 \exp(-|x| - |y|) - 7 \exp(-|x - 4| - |y|) \\ & -19 \exp(-|x + 10| - |y - 5|) + \sin(2xy) + 5 \leq 0 \end{aligned} \quad (22a)$$

and

$$\mathbb{X} = \left\{ \begin{array}{l} [-10^8, 10^8] \\ [-10^8, 10^8] \end{array} \right\}. \quad (22b)$$

Function (22a) is obviously nonlinear nonconvex and becomes highly oscillatory as  $x$  or  $y$  goes away from the origin. It is trivial to show that applying the Hansen–Sengupta’s method to (22) does not result in any improvement on the initial bounding box  $\mathbb{X}$ . The results by the modified Hansen–Sengupta’s and numerical methods are summarized in Table 1. The modified Hansen–Sengupta’s method can indeed produce the correct bounding boxes to bound the disconnected regions of feasible points at any pre-determined accuracy/resolution. However, the CPU time required to identify all the bounding boxes is inversely proportional to the pre-determined accuracy. For instance, the CPU times used to find the bounding boxes at the accuracy of 1.0E-3, 1.0E-5 and 1.0E-7 are respectively, equal to 6.857, 573.998 and 47592.479 seconds. Thus we may conclude that the modified Hansen–Sengupta’s method is only practically applicable to produce bounding boxes approximately or at low accuracy. The numerical approach is clearly capable of correctly identifying the bounding boxes at any given accuracy efficiently. The numerical method not only produces more accurately the bounding boxes by one order of magnitude than the modified Hansen–Sengupta’s method but also is faster by two orders of magnitude. The disconnected feasible regions of Example 1 are shown by shading in Fig. 1 at the resolution of  $(0.01 \times 0.01)$ . Also shown in Fig. 1 are the five smallest disconnected bounding boxes obtained by the numerical method.

The second example is composed of two nonlinear inequalities, namely,

$$g_1(x, y) = \left(\frac{x}{5.0}\right)^{2/3} + \left(\frac{y}{3.0}\right)^{2/3} - 1 \leq 0, \quad (23a)$$

Table 1

Accuracy, CPU times and the bounding boxes of feasible points for Example 1

Methods	Modified Hansen–Sengupta’s	Numerical
Accuracy	1.0E-7	1.0E-8
CPU times (s)	47592.479	429.019
Bounding boxes (1)	$\left\{ \begin{bmatrix} -11.533157920 & -8.588022289 \\ 3.727790651 & 6.443481909 \end{bmatrix} \right\}$	$\left\{ \begin{bmatrix} -11.5331579220 & -8.588022284 \\ 3.727790651 & 6.4434819140 \end{bmatrix} \right\}$
(2)	$\left\{ \begin{bmatrix} -10.072097939 & -9.847720183 \\ 3.499142362 & 3.609321709 \end{bmatrix} \right\}$	$\left\{ \begin{bmatrix} -10.072097944 & -9.847720178 \\ 3.499142361 & 3.609321714 \end{bmatrix} \right\}$
(3)	$\left\{ \begin{bmatrix} -0.705888994 & 0.748897368 \\ -0.705887215 & 0.705888995 \end{bmatrix} \right\}$	$\left\{ \begin{bmatrix} -0.705888990 & 0.748897342 \\ -0.705887190 & 0.705888977 \end{bmatrix} \right\}$
(4)	$\left\{ \begin{bmatrix} 3.973168403 & 4.033239578 \\ 0.441555518 & 0.585334261 \end{bmatrix} \right\}$	$\left\{ \begin{bmatrix} 3.973168396 & 4.033239586 \\ 0.441555601 & 0.585334223 \end{bmatrix} \right\}$
(5)	$\left\{ \begin{bmatrix} 3.558065417 & 4.434598505 \\ -0.381064462 & 0.181202394 \end{bmatrix} \right\}$	$\left\{ \begin{bmatrix} 3.558065414 & 4.434598514 \\ -0.381064461 & 0.181202357 \end{bmatrix} \right\}$

$$g_2(x, y) = \left( \frac{x - 5.0}{5.2} \right)^{2/3} + \left( \frac{y}{7.0} \right)^{2/3} - 1 \leq 0 \quad (23b)$$

and

$$\mathbb{X} = \left\{ \begin{bmatrix} -10^8, 10^8 \\ -10^8, 10^8 \end{bmatrix} \right\}. \quad (23c)$$

Although these two inequalities are rather simple, it is also trivial to show that the Hansen–Sengupta’s method cannot improve the initial bounding box  $\mathbb{X}$  of (23c), as in the case of Example 1. We then used the modified Hansen–Sengupta’s method and our numerical technique to find the bounding box of feasible points. The loop strategy has been applied to deal with multiple inequalities. We have also applied the numerical method to the equivalent representation (20a) of multiple inequality constraints in order to simultaneously handle the two inequalities (23a) and (23b) by setting  $\varepsilon$  to 1.0E-9. The accuracy, CPU times and the bounding boxes by the modified Hansen–Sengupta’s method, and the numerical techniques with loop strategy and by using the equivalent treatment (20a) of multiple inequality constraints are listed in Table 2. As in Example 1, the modified Hansen–Sengupta’s method can indeed produce the bounding box at any given accuracy. Unfortunately, the CPU times (not listed here) are again shown to be inversely proportional to the accuracy, which will definitely limit the practical applicability of the method. Although the two inequalities are rather simple, it still took almost 14 CPU hours to obtain the bounding box at the accuracy of 1.0E-7 (see Table 2). The numerical method with loop strategy successfully generates the same bounding box of higher accuracy but in almost no time (0.02 CPU seconds). The numerical method by using the



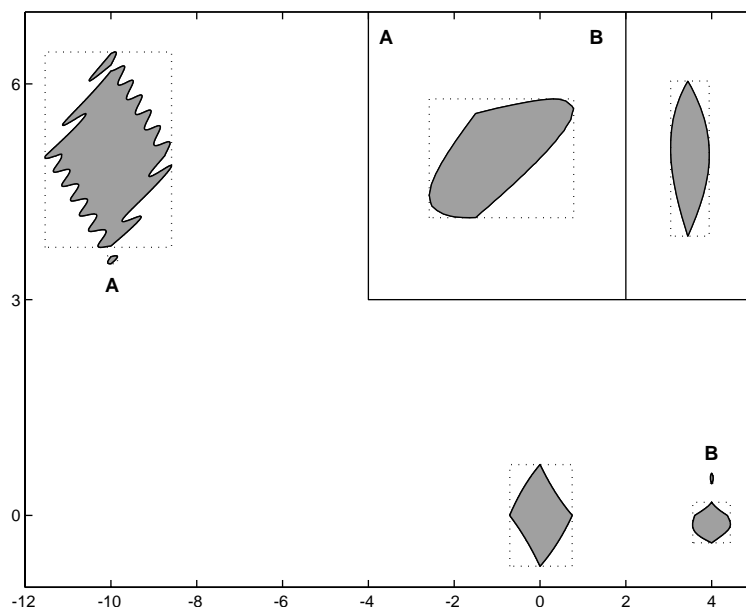


Fig. 1. The (disconnected) feasible regions (shaded areas) and bounding boxes of Example 1 shown in dotted lines. The feasible regions are produced from the function computation at the gridding points of  $(0.01 \times 0.01)$ . The bounding boxes are identified by the numerical method. Since two of the feasible regions and their bounding boxes, namely, A and B, are too small to be clearly visible, we have amplified these two regions and boxes by a factor of 15 and shown them at the upper-right corner.

Table 2

Accuracy, CPU times and the bounding boxes of feasible points for Example 2

Methods	Accuracy	CPU times (s)	Bounding boxes
Modified Hansen–Sengupta’s	1.0E-7	50049.974	$\left\{ \begin{bmatrix} -0.200000025 & 5.000000044 \\ -3.000000026 & 3.000000026 \end{bmatrix} \right\}$
Numerical (loop strategy)	1.0E-8	0.020	$\left\{ \begin{bmatrix} -0.200000008 & 5.000000009 \\ -3.000000008 & 3.000000008 \end{bmatrix} \right\}$
Numerical using (20a)	1.0E-8	0.102	$\left\{ \begin{bmatrix} -0.200000007 & 5.000000008 \\ -1.031647501 & 1.031647501 \end{bmatrix} \right\}$

equivalent representation (20a) has resulted in the correct, smallest bounding box, at the accuracy of 1.0E-8 and only in 0.102 seconds. It is also obvious from Table 2 that the loop strategy cannot produce the smallest possible bounding boxes for the feasible region, as was expected theoretically in Section 3. The bounding boxes from applying the numerical methods with loop strategy and by using (20a) are shown in Fig. 2 in dotted and thick dash-dotted lines, respectively.

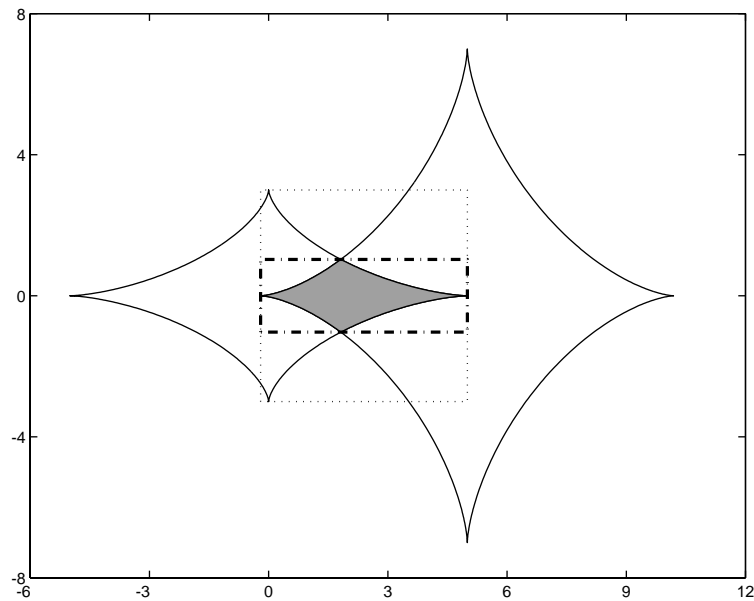


Fig. 2. The feasible region (shaded area) and bounding boxes of Example 2. The feasible region is produced by plotting the functions with Matlab at the resolution of 0.01. The bounding boxes by the numerical methods with loop strategy and through using (20a) are shown in dotted line and in thick-dash-dotted line, respectively.

## 5. Conclusions

Various algorithms have been proposed in the literature either for finding an approximate feasible point (see e.g., [26,29,5,12,6]) or for checking whether a feasible point exists inside a (small) box (see e.g., [9,20]). Although approximation methods can be used to correctly find the smallest box to bound the feasible point set defined by a set of convex inequalities, they may not be very efficient computationally due to the growth of the feasibility problem with iterations, and are even not generally applicable if the inequalities are nonlinear and nonconvex. The first method for finding bounding boxes to bound the disconnected feasible points defined by a nonlinear inequality or a system of nonlinear inequalities was proposed by Hansen and Sengupta [10] (see also [9]). The method has not been practically implemented for finding smallest possible bounding boxes of feasible points defined by a system of nonlinear nonconvex inequalities, though modified versions have been used to bound the constraints of equalities [24,33]. In this paper, we have shown with two examples that the Hansen–Sengupta’s method cannot result in any improvement on an initial bounding box, if it is sufficiently large. Given a feasible point, Kristinsdottir et al. [21] proposed a method: (i) to check whether its neighbourhood is also feasible; and (ii) to grow this feasible point into a maximal feasible region. The strategy proposed in [21] for answering the first question does not always succeed, since their algorithm may indicate that an interval is infeasible when it is really feasible and/or that a tolerance interval is feasible when it is really not. The strategy they proposed to answer the second question was to use a constant step size to grow the neighbourhood and then use interval mathematics to check whether the maximal feasible region has been found. Obviously,

the method is not able to find all disconnected feasible regions. For some other methods of handling inequalities, the reader is referred to a recent excellent review [18], which are not suitable to find the smallest bounding boxes, however.

We have proposed the first numerical method to find the smallest bounding boxes to bound the feasible points of a nonlinear and nonconvex inequality and/or a system of nonlinear and nonconvex inequalities. Two strategies, namely, the loop strategy as proposed in [10] (see also [9]) and the equivalent representation (20), have been used to deal with multiple nonlinear inequalities. Two examples have demonstrated that the proposed numerical algorithm can indeed correctly find the smallest bounding boxes for feasible points, to any given accuracy and efficiently. When comparing our method with that of Kristinsdottir et al. [21], we see that all the problems with the algorithm of Kristinsdottir et al. [21] have been completely circumvented. Our strategy to find the smallest bounding boxes also depends on the nature of a nonlinear inequality or a system of nonlinear inequalities. On the other hand, the correct solution to the second question of Kristinsdottir et al. [21] will generally lose many feasible points, while our method provides the smallest bounding box to bound all the feasible points in the same area. Since one of the core components in our numerical method is a new tool for multisection and multisplitting, this seems also to be the first such theoretically established technique, compared with empirical multisection in [9,3,4]. We have also modified the original Hansen–Sengupta’s method and shown that the modified method is able to find the smallest bounding boxes as well. The computation time it requires is, unfortunately, inversely proportional to the given accuracy/resolution, and thus would limit its practical applicability.

## Acknowledgements

The author would like to thank Prof. N.V. Thoai, Department of Mathematics, Trier University, for his great interest in and more than two hours of discussion on the content of this paper. Part of this work was supported by a Grant-in-Aid for Scientific Research (C13640422).

## References

- [1] D.P. Bertsekas, *Constrained Optimization and Lagrange Multiplier Methods*, Academic Press, New York, 1982.
- [2] J. Burke, S.-P. Han, A Gauss–Newton approach to solving generalized inequalities, *Math. Oper. Res.* 11 (1986) 632–643.
- [3] L.G. Casado, I. García, T. Csendes, A new multisection technique in interval methods for global optimization, *Computing* 65 (2000) 263–269.
- [4] A. Csallner, T. Csendes, M. Markót, Multisection in interval branch-and-bound methods for global optimization I: theoretical results, *J. Global Optim.* 16 (2000) 371–392.
- [5] J.W. Daniel, Newton’s method for nonlinear inequalities, *Numer. Math.* 21 (1973) 381–387.
- [6] J.E. Dennis, M. El-Alem, K. Williamson, A trust-region approach to nonlinear systems of equalities and inequalities, *SIAM J. Optim.* 9 (1999) 291–315.
- [7] C.A. Floudas, *Deterministic Global Optimization: Theory, Methods and Applications*, Kluwer Academic, Dordrecht, 2000.
- [8] E. Hansen, Global optimization using interval analysis: the multi-dimensional case, *Numer. Math.* 34 (1980) 247–270.
- [9] E. Hansen, *Global Optimization Using Interval Analysis*, Marcel Dekker, New York, 1992.

- [10] E. Hansen, S. Sengupta, Global constrained optimization using interval analysis, in: K.L. Nickel (Ed.), *Interval Mathematics*, Academic Press, New York, 1980, pp. 25–47.
- [11] G.T. Herman, H.K. Tuy, Image reconstruction from projections: an approach from mathematical analysis, in: P.C. Sabatier (Ed.), *Basic Methods of Tomography and Inverse Problems*, Adam Hilger, Bristol, 1987, pp. 1–124.
- [12] H. Hong, Heuristic search and pruning in polynomial constraints satisfaction, *Ann. Math. Artif. Intellig.* 19 (1997) 319–334.
- [13] H. Hong, V. Stahl, Safe starting regions by fixed points and tightening, *Computing* 53 (1994) 323–335.
- [14] R. Horst, P.M. Pardalos, N.V. Thoai, *Introduction to Global Optimization*, 2nd Edition, Kluwer Academic Publishers, Dordrecht, 2000.
- [15] R. Horst, H. Tuy, *Global Optimization: Deterministic Approaches*, 3rd Edition, Springer, Berlin, 1996.
- [16] L. Jaulin, Interval constraint propagation with application to bounded-error estimation, *Automatica* 36 (2000) 1547–1552.
- [17] L. Jaulin, M. Kieffer, O. Didrit, É. Walter, *Applied Interval Analysis*, Springer, London, 2001.
- [18] R.B. Kearfott, *Rigorous Global Search: Continuous Problems*, Kluwer Academic, Dordrecht, 1996.
- [19] R.B. Kearfott, A review of techniques in the verified solution of constrained global optimization problems, in: R.B. Kearfott, V. Kreinovich (Eds.), *Applications of Interval Computations*, Kluwer Academic Publishers, Dordrecht, 1996, pp. 23–60.
- [20] R.B. Kearfott, On proving existence of feasible points in equality constrained optimization problems, *Math. Prog.* 83 (1998) 89–100.
- [21] B.P. Kristinsdottir, Z.B. Zabinsky, T. Csendes, M.E. Tuttle, Methodologies for tolerance intervals, *Interval Comput.* 3 (1993) 133–147.
- [22] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, E. Teller, Equation of state calculations by fast computing machines, *J. Chem. Phys.* 21 (1953) 1087–1092.
- [23] R.E. Moore, *Interval Analysis*, Prentice-Hall, Englewood Cliffs, NJ, 1966.
- [24] A. Neumaier, The enclosure of solutions of parameter-dependent systems of equations, in: R.E. Moore (Ed.), *Reliability in Computing*, Academic Press, London, 1988, pp. 269–286.
- [25] A. Neumaier, *Interval Methods for Systems of Equations*, Cambridge University Press, Cambridge, 1990.
- [26] B.T. Polyak, Gradient methods for solving equations and inequalities, *USSR Comput. Math. Math. Phys.* 4 (1964) 17–32.
- [27] B.N. Pshenichnyi, Newton's method for the solution of systems of equalities and inequalities, *Math. Notes Acad. Sci. USSR* 8 (1970) 827–830.
- [28] H. Ratschek, J. Rokne, *New Computer Methods for Global Optimization*, Ellis Horwood, Chichester, 1988.
- [29] S.M. Robinson, Extension of Newton's method to nonlinear functions with values in a cone, *Numer. Math.* 19 (1972) 341–347.
- [30] A.N. Tikhonov, V.Y. Arsenin, *Solutions of Ill-posed Problem*, Wiley, New York, 1977.
- [31] H. Tuy, *Convex Analysis and Global Optimization*, Kluwer Academic Publishers, Dordrecht, 1998.
- [32] P. van Hentenryck, D. Mcallester, D. Kapur, Solving polynomial systems using a branch and prune approach, *SIAM J. Numer. Anal.* 34 (1997) 797–827.
- [33] M.A. Wolfe, An interval algorithm for constrained global optimization, *J. Comput. Appl. Math.* 50 (1994) 605–612.
- [34] P.L. Xu, A hybrid global optimization method: the one-dimensional case, *J. Comput. Appl. Math.* 147 (2002) 301–314.