Noname manuscript No.
(will be inserted by the editor)

Some Observations on Exclusion Regions in Branch and Bound Algorithms

Ralph Baker Kearfott

Received: date / Accepted: date

Abstract In branch and bound algorithms for constrained global optimization, an acceleration technique is to construct regions $\boldsymbol{x}^*$ around local optimizing points $\check{x}$, then delete these regions from further search. The result of the algorithm is then a list of those small regions in which all globally optimizing points must lie. If the constructed regions are too small, the algorithm will not be able to easily reject adjacent regions in the search, while, if the constructed regions are too large, the set of optimizing points is not known accurately. We briefly review previous methods of constructing boxes about approximate optimizing points. We then derive a formula for determining the size of a constructed solution-containing region, depending on a small radius ϵ, and of constructing a containing box $\boldsymbol{X}^* \supset \boldsymbol{x}^*$ such that all points in $\boldsymbol{X}^* \setminus \boldsymbol{x}^*$ are proven to be infeasible, without the need to actually process them in the branch and bound algorithm. The construction differs in its motivation and concept from previous methods of constructing such boxes $\boldsymbol{X}^*$. It may be possible to use this technique to reduce the large amount of processing branch and bound algorithms typically require to fathom regions adjacent to optimizing points, and to obtain more accurate bounds on solution sets.

Keywords cluster problem, backboxing, epsilon-inflation, complete search, branch and bound, interval computations

1 Introduction and Notation

Consider the global optimization problem:

$$\boxed{\begin{array}{l}\text{minimize } \varphi(x)\\ \text{subject to } C(x) = (c_1(x), \ldots c_{m_1}(x)) = 0,\\ \text{where } \varphi : \mathbb{R}^n \to \mathbb{R} \text{ and } c_i : \mathbb{R}^n \to \mathbb{R}, i = 1, \ldots m_1.\end{array}} \tag{1}$$

Department of Mathematics, University of Louisiana at Lafayette, U.L. Box 4-1010, Lafayette, LA 70504-1010 USA
Tel.: 337-482-5270
Fax: 337-482-5346
E-mail: rbk@louisiana.edu

Problem (1) is not general, since it does not include inequality or bound constraints, a generalization we will consider in later work.

Depending on application and context, the minimum value φ^* and a single optimizing point x^* are sought, or all globally optimizing points are sought. It is well-known that the general problem (1) is NP-hard, but application-specific structure can often be used. Sometimes, such structure allows us to construct practical algorithms that compute good approximations to globally optimizing points, while, in other cases, the only practical procedure is to employ heuristics that yield low values of φ that may not be near a global optimum. Examples of the latter include evolutionary algorithms.

The literature on global optimization has exploded over the past several decades, in books, application-specific journals, and on the web. This journal is focussed exclusively on the subject. An extensive web page is at [15].

When usable problem structure is unknown or absent, and especially when an actual global optimum, not merely a low feasible value for φ, is desired, complete search methods, as defined in [14], are used. Such methods employ a version of the well-known branch and bound paradigm for continuous global optimization; the general structure is as in Algorithm 1.

```
   Input : An initial region $\boldsymbol{x}^{(0)}$ (generally a hyper-rectangle, or box), the objective $\varphi$,
           the constraints $C$, a domain stopping tolerance $\epsilon_d$, and a limit $M$ on the
           maximum number of boxes to be allowed to be processed.
   Output: OK = true and the best upper bound $\overline{\varphi}$ for the global optimum, and the list $\mathcal{C}$
           within which all optimizing points must lie, if the algorithm completed with
           less than $M$ boxes, and OK = false if the algorithm could not complete.
 1 Initialize the list $\mathcal{L}$ of boxes to be processed to consist of the single box $\boldsymbol{x}^{(0)}$;
 2 Determine an upper bound $\overline{\varphi}$ on the global optimum;
 3 $i \leftarrow 1$;
 4 while $\mathcal{L} \neq \emptyset$ do
 5     $i \leftarrow i + 1$;
 6     if $i > M$ then return OK = false;
 7     ;
 8     Remove a box $\boldsymbol{x}$ from $\mathcal{L}$;
 9     Determine if $\boldsymbol{x}$ is not infeasible, and if it is not proven to be infeasible, determine a
       lower bound $\underline{\varphi}$ on $\varphi$ over the feasible part of $\boldsymbol{x}$;
10     if $\boldsymbol{x}$ is infeasible or $\underline{\varphi} > \overline{\varphi}$ then return to Step 8;
11     ;
12     Possibly compute a better upper bound $\overline{\varphi}$;
13     if a scaled radius diam of $\boldsymbol{x}$ satisfies diam($\boldsymbol{x}$) $< \epsilon_d$ then
14         Store $\boldsymbol{x}$ in $\mathcal{C}$;
15         Return to Step 8;
16     else
17         Split $\boldsymbol{x}$ into two or more boxes;
18         Put each of the sub-boxes into $\mathcal{L}$;
19         Return to Step 8;
20     end
21 end
22 return OK = true, $\overline{\varphi}$, and $\mathcal{C}$ (possibly empty);
```

Algorithm 1: General Branch and Bound Structure

A huge amount of literature exists for Algorithm 1 in the context of mixed integer linear programs, etc. Some general introductions in the context of general continuous global optimization, including when mathematically rigorous bounds on the optimum and optimizer are desired, are [6], [14], and [23]. A few of the notable software implementations are described in [18], [22], [23], [8, 9], and [19].

Generally, upper bounds $\overline{\varphi}$ can be computed by evaluating φ at carefully computed feasible points, although this can be problematical for constrained problems when mathematical rigor is sought ([6, §4.2.4], [7], and [12]). Lower bounds $\underline{\varphi}$ can be computed with interval arithmetic or by solving relaxations, and may be enhanced with constraint propagation techniques. However, if the upper bound $\overline{\varphi}$ is inaccurate or corresponds to a non-global local optimum, or perhaps more commonly if the lower bounds $\underline{\varphi}$ are not sufficiently sharp or even if an isolated global optimizer lies near a corner of the box containing it, adjacent boxes cannot be rejected in Step 11 of Algorithm 1; this is known as the *cluster problem* [11], [21], [20]. The phenomenon is such that, for particular ways of computing $\overline{\varphi}$ and $\underline{\varphi}$, the number of boxes that cannot be rejected near an isolated global optimizer[1] does not decrease as ϵ_d is decreased.

Clustering of non-rejected boxes around isolated global optimizers can be avoided by constructing a box $\boldsymbol{x}^*$ with radius[2] significantly larger than ϵ_d about a point thought to be an accurate representation of the optimizer, and eliminating that region from the search[3]. An important question is how to construct $\boldsymbol{x}^*$, specifically, how wide to make it. An early, abstract analysis of this question in the context of nonlinear systems of equations[4] appears in [5]; there, the box $\boldsymbol{x}^*$ was constructed with radius proportional to $\sqrt{\epsilon_d}$. Also, in the nonlinear systems of equations context, there is a history of construction of boxes $\boldsymbol{x}^*$ about approximate solutions $\check{x}$ in which an interval Newton method $\boldsymbol{N}(F;\boldsymbol{x}^*,\check{x})$ has $\boldsymbol{N}(F;\boldsymbol{x}^*,\check{x}) \subset \boldsymbol{x}^*$, thus proving existence and possibly uniqueness of solutions to the system $F(x) = 0$ (where F maps a subset of $\mathbb{R}^n$ into $\mathbb{R}^n$). In [17], Rump proposes a method to circumscribe $\check{x}$ with a box $\boldsymbol{x}^*$ of as large a radius as possible, within which an interval Newton method guarantees uniqueness of a solution to $F(x) = 0$, and within which there is a box $\check{\boldsymbol{x}}^* \subseteq \boldsymbol{x}^*$ such that $\check{\boldsymbol{x}}^*$ is as small as possible, subject to an interval Newton method proving that $\check{\boldsymbol{x}}^*$ contains a solution. Mayer surveys such "ϵ-inflation" techniques in [13]. Schichl and Neumaier provide an interesting analysis along these lines, for nonlinear systems of equations, in the context of branch and bound algorithms, in [21], and in the context of unconstrained global optimization in [20]. An interesting aspect of the analysis of Schichl and Neumaier is that it uses higher-order information, so has the potential to handle cases where the Jacobian matrix of the system is singular.

A more recent work containing a practical procedure successful for verifying feasibility on an extensive test set is [2]. In contrast to that work, our motivation here is to construct a box $\boldsymbol{x}^*$ such that boxes bordering $\boldsymbol{x}^*$ can be easily eliminated in a branch-and-bound algorithm, in contrast to constructing a small $\boldsymbol{x}^*$ within which a feasible point is known to exist.

[1] Non-isolated global optimizers, such as, for example, the optimizers of the unconstrained problem $\varphi(x) = (x_2 - x_1^2)^2$, pose additional different problems to branch and bound algorithms.

[2] in a scaled infinity norm

[3] One method of modifying the list $\mathcal{L}$ once an $\boldsymbol{x}^*$ has been constructed appears in [6, §4.3.1]. That method produces at most $2n$ additional boxes surrounding the optimizer-containing box.

[4] This analysis also dealt primarily with the case where solutions happen to lie near a boundary or vertex.

In constrained global optimization, interval Newton methods can be used in various ways to construct exclusion boxes $\boldsymbol{x}^*$ containing global minimizers. For example, the Kuhn–Tucker or Fritz John conditions may be used as the system of equations. However, in many cases of interest, these systems are ill-conditioned or singular at optimizers, sometimes provably so [12]; this forces the process to either fail or the boxes $\boldsymbol{x}^*$ to be too small to avoid the cluster problem. Alternately, a system of equations may be formed directly from the approximately active constraints and a selected subspace of variables as in [6, §5.2.4] and [7], but problems occurring in practical applications often have linearly dependent constraints, with more active constraints than variables [10].

In our GlobSol software [8, 9], we avoid ill-conditioning problems with Kuhn–Tucker or Fritz John equations by constructing $\boldsymbol{x}^*$ roughly according to the ideas in [5] with size proportional to $\sqrt{\epsilon_d}$ and scaled according to the partial derivatives of the objective and constraints, as in [10, Formula (13)]. However, in general, small boxes $\check{\boldsymbol{x}}^* \subset \boldsymbol{x}^*$ cannot be constructed such that $\check{\boldsymbol{x}}^*$ is verified to contain a critical point or local optimizer. Thus, without verification with an interval Newton method, the list $\mathcal{C}$ Algorithm 1 generates is only guaranteed to contain all globally optimizing points, the individual boxes $\boldsymbol{x}^* \in \mathcal{C}$ may not contain optimizing points; in return, we are freed to adjust the radii of the constructed $\boldsymbol{x}^*$ to avoid clustering or otherwise improve the efficiency within the branch and bound process.

This work is within this context of heuristically adjusted box sizes of boxes $\boldsymbol{x}^*$ to be placed in $\mathcal{C}$. If the radius of $\boldsymbol{x}^*$ is too small, clustering will occur, resulting in a large number of boxes in $\mathcal{C}$ and possibly in the inability of the algorithm to complete within M steps. On the other hand, if the radius[5] of $\boldsymbol{x}^*$ is larger than it needs to be to avoid clustering, the bounds on optimizers will not be tight. In Section 2, we present a method of constructing a box $\boldsymbol{x}^*$ and a box $\boldsymbol{X}^* \supset \boldsymbol{x}^*$ such that each point in $\boldsymbol{X}^* \setminus \boldsymbol{x}^*$ is infeasible[6]. The method is motivated by imagining small boxes of radius ϵ about $\boldsymbol{x}^*$, presumably produced by some branch and bound algorithm. Two illustrative examples are given. We have tried the technique on a test set from the literature, and report the results in in Section 3, while we draw conclusions and propose how the analysis can be used in Section 4.

2 Ideas and Algorithms for Setting the Exclusion Region Size

Given an approximate optimizing point $\check{x}$, we will construct a box $\boldsymbol{x}^*$ about $\check{x}$ and a box $\boldsymbol{X}^*$ containing $\boldsymbol{x}^*$ such that every point in the shell $\boldsymbol{X}^* \setminus \boldsymbol{x}^*$ is known to be infeasible. As indicated in the introduction, such constructions have a long history, perhaps beginning with work of Rump, as in [17]. The novelty of our construction is that, given an ϵ, we determine the widths of $\boldsymbol{x}^*$ by imagining $\boldsymbol{X}^* \setminus \boldsymbol{x}^*$ to consist of boxes whose coordinate widths are all ϵ. Our motivation for this approach is imagining $\boldsymbol{x}^*$ to have been deleted from the search space in a branch and bound algorithm and, absent other acceleration techniques, boxes of width ϵ adjacent to

[5] The *radius* of an interval is half the width of the interval, while we may take the radius of a box (where a box is an interval vector) to be the maximum of the radii of its component intervals.

[6] We do not explicitly deal with the objective φ in this work or with transforming the problem and objective into an appropriate form.

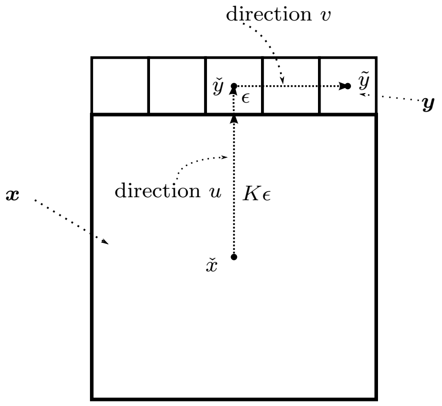

Fig. 1 Analysis of box rejection on a boundary

the solution box would have been produced by the branch and bound algorithm; with our construction, $\boldsymbol{X}^*$ would be eliminated without branching and bounding.

We summarize our actual goal and assumptions.

Goal: Given an approximate optimizing point $\check{x}$ and an $\epsilon > 0$, construct a box $\boldsymbol{x}^*$ and a containing box $\boldsymbol{X}^*$, $\boldsymbol{x}^* \subset \boldsymbol{X}^*$, such that some of the coordinates of $\boldsymbol{X}^*$ extend outward distance ϵ from $\boldsymbol{x}^*$ and every point in $\boldsymbol{X}^* \setminus \boldsymbol{x}^*$ is proven to be infeasible.

Assumptions: The constraints C are continuously differentiable, and mathematically rigorous bounds on the partial derivatives of the components of c over any box $\boldsymbol{x}$ are computable.

2.1 Expansion Factors for the Constraints

Refer to Figure 1 in this analysis. As illustrated, we consider the expansion box $\boldsymbol{x}$ with radius $K\epsilon$, centered at the approximate global optimizer $\check{x}$, we consider a face of this box in the direction u from $\check{x}$, we wish to analyze when a box $\boldsymbol{y}$ of radius ϵ at $\tilde{y}$ will be rejected due to an equality constraint $c(x)$ with $c : \mathbb{R}^n \to \mathbb{R}$, and we decompose $\tilde{y} - \check{x}$ into a component $\check{y} - \check{x}$ of length $(K+1)\epsilon$ in the direction of u and a component $\tilde{y} - \check{y}$ in the direction v orthogonal to u, and consider an arbitrary $y \in \boldsymbol{y}$. We have

$$\begin{aligned} c(y) &= c(\check{x}) + [c(\check{y}) - c(\check{x})] + [c(\tilde{y}) - c(\check{y})] + [c(y) - c(\tilde{y})] \\ &= c(\check{x}) + \nabla_u c(\xi)(K+1)\epsilon + \nabla_v c(\eta)(\tilde{y} - \check{y}) + \nabla c(\zeta)(y - \tilde{y}), \end{aligned} \tag{2}$$

where ∇_u is the scalar directional derivative in the direction u, ∇_v is the gradient in the $(n-1)$-dimensional subspace orthogonal to u, and ∇ is the full n-dimensional gradient, ξ is on the line between $\check{x}$ and $\check{y}$, η is on the line between $\check{y}$ and $\tilde{y}$, and ζ is on the line between $\tilde{y}$ and y. Now, define μ, $\mathcal{M}$, M, L, and δ such that

$$\begin{array}{ll} \mu \le \min_{t\in[0,1]} |\nabla_u c(\check{x}+t(\check{y}-\check{x}))|, & \mathcal{M} \ge \max_{t\in[0,1]} \|\nabla_v c(\check{y}+t(\tilde{y}-\check{y}))\|, \\ M \ge \max_{\|y-\check{x}\|\le\sqrt{(K+2)^2+(L+1)^2}\epsilon} \|\nabla c(x)\|, & L \ge \|\tilde{y}-\check{y}\|/\epsilon, \text{ and } \delta \ge |c(\check{x})|/\epsilon. \end{array} \tag{3}$$

In the computations to follow, all of these norms are taken to be $\|\cdot\|_2$. Observe the minimum possible L is proportional to the coordinate widths of $\boldsymbol{x}$ in directions perpendicular to u. Also, if the constraints were exactly satisfied at $\check{x}$, we can choose $\delta = 0$; it will become apparent we will want δ to be small in relation to the minimum box radius ϵ in any case.

Using (3) in (2) gives

$$|c(y)| \ge (K+1)\epsilon\mu - L\epsilon\mathcal{M} - \epsilon M - \epsilon\delta. \tag{4}$$

Thus, (4) implies $|c(y)| > 0$ for $y \in \boldsymbol{y}$ provided we can choose K such that

$$K > \frac{L\mathcal{M} + M + \delta - \mu}{\mu} \quad \text{provided } \mu > 0 \text{ and } L\mathcal{M} + M + \delta - \mu > 0. \tag{5}$$

For example, if c is linear and u is in the direction of ∇c, $\mathcal{M}$ can be chosen to be 0, M and μ can be chosen to be equal, and any K will do provided δ is sufficiently small.

If we assume $L \le \gamma K + a$ for some constants γ and a and we assume we can bound μ, $\mathcal{M}$, and M with interval arithmetic, we may solve (5) for K to obtain

$$K > \frac{a\mathcal{M} + M + \delta - \mu}{\mu - \gamma\mathcal{M}} \quad \text{provided } \mu > \gamma\mathcal{M} \text{ and } a\mathcal{M} + M + \delta - \mu > 0. \tag{6}$$

These conditions only take account of overestimation when bounding c over $\boldsymbol{y}$ using interval evaluations if μ, M, and $\mathcal{M}$ are bounded using interval computations. In practice, μ, $\mathcal{M}$, and M can be computed with interval evaluations over a box with radii based on $K_{\max}\epsilon$, where $K_{\max}$ is the maximum allowable value for K, and the computed K is rejected if $K > K_{\max}$.

We summarize these computations with the following.

Proposition 1 *Assume the constraints of (1) are continuously differentiable, assume that a maximum extent $K_{\max}$ has been chosen, assume a direction u parallel to a coordinate direction, say the i-th, has been chosen, assume c is any linear combination of the component constraints of C and assume μ, $\mathcal{M}$, M, and L satisfy*

$$\begin{array}{ll} \mu \le \min_{\|x-\check{x}\|_\infty \le K_{\max}\epsilon} |\nabla_u(x)|, & \mathcal{M} \ge \max_{\|x-\check{x}\|\le K_{\max}\epsilon} \|\nabla_v(x)\|, \\ M \ge \max_{\|x-\check{x}\|\le K_{\max}]]\epsilon} \|\nabla c(x)\|, & \delta \ge |c(\check{x})|/\epsilon, \end{array}$$

where ∇_v is the gradient of c with the coordinate parallel to u excluded. and set $\gamma = \sqrt{n-1}$, $a = 3\sqrt{n-1}$, and $L = \sqrt{n-1}\,(K_{\max}+3)$. Assume that a box $\boldsymbol{x}^*$ centered at $\check{x}$ is constructed such that the widths in each coordinate direction are equal to $K\epsilon$, where K is computed either to (5) (with associated condition), or (6) (with associated condition), and $K+3 < K_{\max}$. Then the box of radius ϵ in the $i-th$ coordinate direction e_i and radius $(K+3)\epsilon$ in the other coordinate directions and centered on $\check{x}+\sigma(K+1)\epsilon e_i$, $\sigma = \pm 1$, is infeasible.

Proof (of Proposition 1) The proposition follows from the definitions (3), the assumptions in the proposition, the derivation of (5) and (6) from the first-order expansions of c, and Proposition 2 below. □

We have assumed $L = \gamma K + a$ for some γ and a, but how did we know what γ and a should be? If each coordinate radius of $\boldsymbol{x}^*$ is equal[7], what can L be for $\boldsymbol{x}^* \subset \mathbb{R}^n$? Along these lines, it is logical to extend a face at $x_i = \check{x}_i + \sigma K\epsilon$, $\sigma = 1$ or -1, beyond $x_j \in [\check{x}_j - K\epsilon, \check{x}_j + K\epsilon]$, $j \neq i$, to $x_j \in [\check{x}_j - (K+1)\epsilon, \check{x}_j + (K+1)\epsilon]$; this facilitates construction of a cube $\boldsymbol{x}^*$ is contained in a cube $\boldsymbol{X}^*$ such that all points in $\boldsymbol{X}^* \setminus \boldsymbol{x}^*$ are infeasible; see Figure 2.

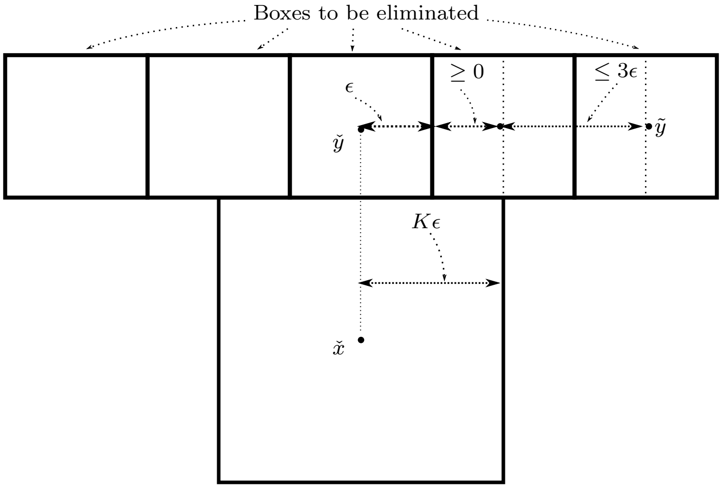

Fig. 2 Analysis of border elimination

Proposition 2 *Suppose we wish to enclose the box $\boldsymbol{x}^*$ whose i-th coordinate extents are $\boldsymbol{x}_i^* = [\check{x}_i - K\epsilon, \check{x}_i + K\epsilon]$ within a box $\boldsymbol{X}^*$ constructed by covering each coordinate face of $\boldsymbol{x}^*$ with boxes of radii ϵ and extending outward from each coordinate face one box in each direction beyond boxes sharing a part of a face of $\boldsymbol{x}^*$; see Figure 2. Adequate γ and a for $L \leq \gamma K + a$ to describe $\tilde{y} \in \boldsymbol{X}^* \setminus \boldsymbol{x}^*$ are $\gamma = \sqrt{n-1}$ and $a = 3\sqrt{n-1}$, i.e. we may take*

$$L = \sqrt{n-1}\,(K_{\max} + 3)\,.$$

Proof (of Proposition 2) Refer to Figure 2. $\boldsymbol{X}^* \setminus \boldsymbol{x}^*$ consists of $2n$ hyper-rectangles with overlapping edges, where each hyperrectangle has an i-th coordinate fixed at $\boldsymbol{z}_i = \check{y}_i = \check{x}_i + \sigma(K+1)\epsilon$ for $\sigma = 1$ or $\sigma = -1$ and other coordinates $\boldsymbol{z}_j$, $j \neq i$ with

$$\boldsymbol{z}_j \subseteq [\check{x}_j - (K+3)\epsilon, \check{x}_j + (K+3)\epsilon] = [\check{y}_j - (K+3)\epsilon, \check{y}_j + (K+3)\epsilon].$$

The farthest a small box center $\tilde{y}$ can be from $\check{y}$ in such a hyper-rectangle $\boldsymbol{z}$ is when it is in the center of a box with coordinate radii ϵ located in a corner of

[7] We may modify this later by using scaled distances or by locally scaling the constraints.

the hyper-rectangle. Each coordinate of such a box center lies at distance at most $(K+3)\epsilon$ from the corresponding coordinate of $\check{y}$. The conclusion of the proposition therefore follows from the Pythagorean theorem. □

We now give an example illustrating the value of these techniques, followed by an example illustrating the techniques' limitations.

Example 1

$$\begin{aligned} &\min x_1 \\ &\text{subject to: } c(x) = x_1^2 - x_2^2 - 1 = 0, \\ &\qquad\qquad (x_1, x_2) \in ([-2,2],[-2,2]). \end{aligned}$$

The unique global optimizer for this problem is at $\check{x} = (-1, 0)$, at which $\varphi(\check{x}) = -1$. Here $\nabla c(\check{x}) = (-2, 0)^T$, so this constraint may be most useful in eliminating boxes bordering faces of x perpendicular to the first coordinate direction, that is, a promising u is $u = (\pm 1, 0)$; let's do the analysis with $u = (1, 0)^T$. For simplicity, suppose the box $\boldsymbol{x}^*$ about $\check{x}$ has been constructed to have equal widths in both coordinate directions, so it is of the form

$$\boldsymbol{x}^* = ([-1 - K\epsilon, -1 + K\epsilon], [-K\epsilon, K\epsilon]).$$

(Note that, in this example, this corresponds to Figure 1 and not Figure 2 or Proposition 2.) As in practice, we compute μ, $\mathcal{M}$, and M with interval arithmetic over a box with coordinate radii $K_{\max}\epsilon \gg K\epsilon$. In this example, let's take $\epsilon = 10^{-6}$. If we assume $\boldsymbol{x}^*$ has equal widths in all directions, then we may take $\gamma = 1$; also, we may take $a = 0$ and $\delta = 0$. If we assume $K < 100$, we may evaluate $\nabla_u c$, $\nabla_v c$ and ∇c over the box

$$([-1 - 10^{-4}, -1 + 10^{-4}], [-10^{-4}, 10^{-4}])$$

to obtain

$$\nabla c \in ([-2.0002, -1.9997], [-0.0002, 0.0002])^T,$$

and we may take $\mu = 1.9997$, $\mathcal{M} = 0.0002$, $M = \sqrt{2.0002^2 + 0.0002^2} \le 2.0003$. Since $\mu = 1.9997 > 0.0002 = \gamma\mathcal{M}$, we may employ (6) to obtain

$$K \ge \frac{2.0003 - 1.9997}{1.9997 - 0.0002}.$$

Since this fraction is bounded by 0.003, $K = 1$ will do.

Indeed, with $K = 1$, the single box with radius $\epsilon = 10^{-6}$ that covers the face $(-1 + 10^{-6}, [-10^{-6}, 10^{-6}])$ of $\boldsymbol{x}^*$ is

$$\boldsymbol{y} = ([-1 + 10^{-6}, -1 + 3 \cdot 10^{-6}], [-10^{-6}, 10^{-6}]),$$

and a naive interval evaluation of c within Intlab (see [16]) gives

$$c(\boldsymbol{y}) \subset [-0.6, -0.199999] \times 10^{-5},$$

so, indeed, $0 \notin c(\boldsymbol{y})$.

Example 2 Consider

$$\begin{aligned} &\min(x_1 - x_2)^2 \\ &\text{subject to: } c(x) = x_2 - e^{x_1 x_2} = 0, \\ &\qquad\qquad (x_1, x_2) \in ([-2, 2], [-2, 2]). \end{aligned}$$

This problem has a unique global optimizer at $\check{x} = (0, 1)$, at which $\varphi(\check{x}) = 1$. As in Example 1, let us construct $\boldsymbol{x}^*$ with equal coordinate widths as

$$\boldsymbol{x}^* = ([-1 - K\epsilon, -1 + K\epsilon], [-K\epsilon, +K\epsilon]),$$

so the minimum possible L has $L \le K - 1/2 < K$. We have $\nabla c = (-x_2 e^{x_1 x_2}, 1 - x_1 e^{x_1 x_2})^T$, both of whose coordinates have equal magnitude at $(0, 1)$. If we choose $u = (0, 1)^T$, we will be looking to reject boxes lining the face $([-K\epsilon, K\epsilon], 1 + K\epsilon)$ of $\boldsymbol{x}^*$. If we assume each box on this face to be rejected has radius in each direction equal to ϵ (see Figure 1), the minimum possible L has $L \le K - 1/2 < K$, so we may take $\gamma = 1$ and $a = 0$. Furthermore, $|\nabla_u c|$ is constantly equal to 1 on the line between $\check{x}$ and $\check{y}$, so $\mu = 1$ will do. Also, since we are working in $\mathbb{R}^2$, v is the single direction $(1, 0)^T$ (to within algebraic sign), and, along the line between $\check{y}$ and $\tilde{y}$,

$$\nabla_v c = -x_2 e^{x_1 x_2} \in -(1 + (K + 1)\epsilon) e^{(1 + (K+1)\epsilon)[-K\epsilon, K\epsilon]},$$

so

$$\|\nabla_v c\| \le (1 + (K + 1)\epsilon) e^{(1 + (K+1)\epsilon)K\epsilon} \le 1.36$$

if $K\epsilon \le 0.1$ and $\epsilon \le 0.1$, in which case $\mathcal{M} = 1.36$ will do. Finally, if $y \in \boldsymbol{y}$, y will be inside a box of radius $(K + 2)\epsilon$ in each coordinate direction and centered at $\check{x}$, and, if $\| \cdot \| = \| \cdot \|_\infty$, M can be taken as $(1 + (K + 2)\epsilon) e^{((1 + (K+2)\epsilon)(K+2)\epsilon} \le 1.23 = M$ if $(K + 2)\epsilon \le 0.1$. Combining all of this and using (5) gives $K > (1.23K + 1.36 - 1)/1$. There is no such K, and the analysis fails in this case. (Alternatively, using (6), $\mu = 1 < \gamma\mathcal{M} = 1.23$, and the condition cannot be applied.)

The analysis failed for Example 2 because ∇_v was not small in relation to ∇_u. In general, if the problem is dominated by the constraints, a particular constraint would be most effective at eliminating boxes on faces of $\boldsymbol{x}^*$ most nearly orthogonal to that constraint, as was illustrated in Example 1. Here,we say that a constraint c is orthogonal to a face

$$\boldsymbol{x}^*_{\underline{i}} = (\boldsymbol{x}^*_1, \ldots, \boldsymbol{x}^*_{i-1}, \underline{x}^*_i, \boldsymbol{x}^*_{i+1}, \ldots, \boldsymbol{x}^*_n) \quad \text{or} \quad \boldsymbol{x}^*_{\bar{i}} = (\boldsymbol{x}^*_1, \ldots, \boldsymbol{x}^*_{i-1}, \overline{x}^*_i, \boldsymbol{x}^*_{i+1}, \ldots, \boldsymbol{x}^*_n)$$

provided $\nabla c(\check{x})$ (and hence all values of ∇c in a sufficiently small box around $\check{x}$) is approximately a scalar multiple of the i-th coordinate vector.

2.2 Using the Objective Values

There are various possibilities for constructing expansion factors based on the objective. Different approaches are warranted for problems of different types. However, if the problem is constraint-dominated, it may be appropriate to convert all constraints to equality constraints, as in [2], and to similarly replace the objective by a single slack variable, so the objective is equal to one of the coordinate variables, with known lower and upper bounds. This is the approach we adopt in this work; we will investigate alternative approaches later.

If we have a linear objective equal to one of the coordinate variables, we simply construct coordinate bounds for that variable with lower bound computed with constraint propagation and upper bound computed as the best upper bound on the global optimum. The best upper bound can then be propagated into the constraints.

2.3 Combining the Constraint Expansion Factors

In this section, we propose handling equality-constrained problems by combining estimates as in Sections 2.1 and 2.2 to use all constraints to analyze all faces of $\boldsymbol{x}^*$.

Example 2 illustrates that, even with problems with a single constraint, it may not be possible to construct a box around an optimizing point in a way that all adjacent boxes with radii ϵ are rejected. This can be due to the way the problem is posed or to the angles between the null space of the constraint gradients and the coordinate directions. In some cases, this can be mitigated by taking a linear combination of the constraints (equivalent to preconditioning the m by n matrix of constraint gradients) and using the fact that

$$\text{if, for some } \{\alpha_i\}_{i=1}^{m_1}, \quad \sum_{i=1}^{m_1} \alpha_i c_i(x) \neq 0 \quad \text{then } c_i(x) \neq 0 \text{ for at least one } i. \tag{7}$$

We will use (7) in our proposed method for computing the expansion factor K. We present some definitions to aid our exposition.

Definition 1 Let $C(x) = (c_1(x), \ldots, c_{m_1}(x))$ represent the equality constraints in the optimization problem (1). Then the *optimal constraint combination* at an optimizing point $\check{x}$ corresponding to coordinate direction i is that constraint combination

$$\tilde{c}_i = \sum_{j=1}^{m_1} \alpha_j c_j(x) \quad \text{which minimizes } \|(\nabla C(\check{x}))\alpha - e_i\|_2$$

over the parameters $\alpha = (\alpha_1, \ldots, \alpha_{m_1})$, where $\nabla C(\check{x})$ is the n by m_1 matrix whose i-th column is the gradient of c_i at $\check{x}$ and where $e_i \in \mathbb{R}^n$ is the i-th coordinate vector.

Choosing $\tilde{c}_i$ according to Definition 1 gives us the linear combination of constraints such that when u is the i-th coordinate direction, μ is large and $\mathcal{M}$ is small, making the right member of (4) likely to be positive. This brings us to

Definition 2 The *expansion factor for the i-th coordinate* in direction $p \in \{-1, 1\}$ at an optimizing point $\check{x}$ is defined to be $K_{i,p}$, where $K_{i,p}$ is the constraint expansion factor for $\tilde{c}_i$ in the i-th coordinate direction (where $\tilde{c}_i$ replaces the constraint c in the development in Section 2.1), computed through (5) and possibly with (6) if $\mu > \gamma\mathcal{M}$, provided the computed $K_{i,p}$ corresponds to a region in which the bounds in (3) are valid; if the computed $K_{i,p}$ would lie outside those bounds, $K_{i,p}$ is defined to be ∞.

The bounds in (3) can generally be computed by interval evaluations over a suitably large box, where the size of such a box can be determined if a maximum practical value of K can be given.

We now consider combining the coordinate expansion factors $K_{i,p}$ to determine a K valid for as many coordinate directions as we can.

Definition 3 Suppose a maximum practical constraint expansion factor $K_{\max}$ has been determined. The *combined constraint expansion factor* $\overline{K}$ is then defined as

$$\overline{K} = \begin{cases} \max\limits_{\substack{1 \le i \le n, p \in \{-1,1\} \\ K_{i,p} < K_{\max}}} K_{i,p} & \text{if } \exists K_{i,p} < K_{\max}, \\ \\ K_{\max} & \text{otherwise.} \end{cases} \tag{8}$$

In fact, successful computation a combined constraint expansion factor $\overline{K}$ allows us to remove shells of depth ϵ bordering $\boldsymbol{x}^*$ from the search region without subdividing them with any branch and bound process:

Theorem 1 *Suppose we have Problem (1) (that is, the problem is formulated so there are only equality constraints), suppose $\check{x}$ is an optimizing point of this problem, suppose the combined constraint expansion factor $\overline{K}$ has been constructed for a given ϵ according to Definition 3, and suppose a hypercube $\boldsymbol{x}^*$ has been constructed with center at $\check{x}$ and equal radius $\overline{K}\epsilon$ in each coordinate direction. With this $\boldsymbol{x}^*$, suppose i is any coordinate index such that $K_{i,p} < K_{\max}$. Then $\tilde{c}_i$ is infeasible at every point y such that $|y_j - \check{x}_j| \le \overline{K}\epsilon$ for $j \ne i$ and $\overline{K}\epsilon \le |y_i - \check{x}_i| \le (\overline{K}+1)\epsilon$.*

In other words, the conclusion of Theorem 1 states that the two shells of depth 2ϵ bordering the two i-th faces of $\boldsymbol{x}^*$ with $x_i = \check{x}_i \pm \overline{K}\epsilon$ are infeasible, for each i with expansion factor for the i-th coordinate less than $K_{\max}$.

Proof (of Theorem 1) The conclusion follows directly from the definition of $\overline{K}$ (Definition 3) and Formulas 5 or 6. □

Computation of $\overline{K}$ and utilization of Theorem 1 with Proposition 2 can in many cases allow us to eliminate portions of the search space bordering some of the faces of solution boxes $\boldsymbol{x}^*$ without actually performing a branch and bound algorithm and testing a large number of such bordering boxes for infeasibility. Also, observing the sizes of $\overline{K}$ and $K_{\max}$ and the number of faces of $\boldsymbol{x}^*$ for which $K_{i,p} < \infty$ provides a gauge of how difficult the remaining branch and bound search is likely to be.

For example, suppose $n = 10$ and $K = 32$; in that case, it would take $2^{45} \approx 3.5 \times 10^{13}$ boxes of uniform radius ϵ in all coordinate directions to cover a face of $\boldsymbol{x}^*$. Fortunately, many problems are formulated so each constraint depends only on a small number of variables, so the subdivision may be done in a low-dimensional subspace. Taking advantage of this may render some otherwise intractable problems practical to solve.

3 Experimental Results

We programmed computation of $\overline{K}$ within our GlobSol [9] user environment. We applied the computation to the problems from [10][8]. We coded routines within

[8] We used these problems (1) since they represent an independent test set, and (2) since we had these particular problems in a form we could easily manipulate.

Table 1 Success statistics for Theorem 1.

Problem	n	m_1	$\overline{K}$	N_q
ex14.1.1	13	9	0.6	12
ex14.1.8	13	9	0.0	0
ex14.2.1	22	17	0.0	0
ex14.2.2	17	13	0.0	0
ex14.2.4	22	17	0.0	0
ex14.2.5	17	13	0.0	0
ex14.2.6	22	17	0.0	0
ex14.2.8	17	13	0.0	0
ex14.2.9	17	13	0.0	0
ex2.1.4	23	16	0.0	0
ex5.2.2.1	31	25	0.1	6
ex5.2.2.2	31	25	0.1	6
ex5.2.4	28	21	0.2	18
ex7.3.3	13	9	3.6	4
ex8.1.7	21	16	5.0	6
ex9.1.2	22	20	0.0	32
ex9.1.7	66	57	5.3	36
ex9.1.9	51	39	4.0	30
ex9.2.4	21	19	0.2	24
harker	42	28	0.0	0
haverly	30	24	0.1	4
house	19	14	4.3	8
immun	44	29	0.0	4
qp5	219	140	0.0	0
sambal	19	11	0.0	0
sample	16	11	0.0	4

GlobSol to automatically add slack variables and convert the problem's internal representation to one in which the objective consists of a slack variable and all constraints are equality constraints. We used $K_{\max} = 10$ and $\epsilon = 10^{-3}$, and we recorded $\overline{K}$ and the number of faces for which $K_{i,p} < K_{\max}$. We used only those problems for which the approximate optimizer (IPOPT [24]) gave a reasonable feasible point. We ran the experiments on an Intel i7-3820-based computer with an up-to-date Ubuntu 14.04 operating system and the gcc compiler suite. The additional processor time beyond computing the point $\check{x}$ within IPOPT was not significant for these problems, and the processor time to compute $\check{x}$ was not significant except for problem qp5.

In reporting our results, we use the following term.

Definition 4 A *qualified face* of a box $\boldsymbol{x}^*$ constructed according to Theorem 1 is a face with x_i constant upon which $K_{i,j_i} < K_{\max}$. All other faces are said to be *unqualified.*

The results appear in Table 1. In this table, N_q gives the number of faces that were found to be qualified. It can be seen from Table 1 that, although not successful for all problems, the computations underlying Theorem 1 are possibly a valuable adjunct to branch and bound algorithms to global optimization. For example, for ex9.1.7, 2/3 of the faces of $\boldsymbol{x}^*$ were found to be qualified with $\overline{K} = 1.5$. This allows us to eliminate shells of significant volume around $\check{x}$ in which it is known there are no global optimizers, without subdividing in the branch and bound algorithm.

4 Conclusions and Future Work

We have studied the process of elimination of boxes surrounding optimizer-containing boxes within branch and bound processes. We have proposed a formula for constructing an optimum-containing box and eliminating shells surrounding such optimum-containing boxes. This process shows promise for improving branch and bound algorithms by computing a minimal box around which an adjacent region can be eliminated. Regarding the actual shells adjacent particular faces of the optimum-containing box, the method is potentially more powerful than the "shaving" processes in Numerica [22], [3], and elsewhere, since the term $\nabla c(\eta)(y - \tilde{y})$ in (2) represents an increment over a small box rather than an entire bordering shell.

Future work can include the following.

1. Develop and test an algorithm to handle unqualified edges and to eliminate shells beyond adjacent faces of $\check{x}$, working outward from the faces rather than inward through a branch-and-bound algorithm, and utilizing the structure gained from Theorem 1.
2. Compare the process to adaptive shaving.
3. Evaluate the process within a branch and bound algorithm.

Acknowledgement

I thank the referees for their exceptionally careful reading and critique, and I apologize for the inconsistencies in the original submission. The refereeing considerably improved the exposition, clarity, and mathematical correctness.

References

1. Tibor Csendes, editor. *Developments in Reliable Computing: Papers presented at the International Symposium on Scientific Computing, Computer Arithmetic, and Validated Numerics, SCAN-98, in Szeged, Hungary*, volume 5(3) of *Reliable Computing*, Dordrecht, Netherlands, 1999. Kluwer Academic Publishers.
2. Ferenc Domes and Arnold Neumaier. Rigorous verification of feasibility. *Journal of Global Optimization*, Online First, March 2014.
3. Alexandre Goldsztejn and Frédéric Goualard. Box consistency through adaptive shaving. In Sung Y. Shin, Sascha Ossowski, Michael Schumacher, Mathew J. Palakal, and Chih-Cheng Hung, editors, *Proceedings of the 2010 ACM Symposium on Applied Computing (SAC), Sierre, Switzerland, March 22-26, 2010*, pages 2049–2054. ACM, 2010.
4. Jürgen Herzberger, editor. *Topics in Validated Computations: proceedings of IMACS-GAMM International Workshop on Validated Computation, Oldenburg, Germany, 30 August–3 September 1993*, volume 5 of *Studies in Computational Mathematics*, Amsterdam, The Netherlands, 1994. Elsevier.
5. Ralph Baker Kearfott. Abstract generalized bisection and a cost bound. *Mathematics of Computation*, 49(179):187–202, July 1987.

6. Ralph Baker Kearfott. *Rigorous Global Search: Continuous Problems.* Number 13 in Nonconvex Optimization and its Applications. Kluwer Academic Publishers, Dordrecht, Netherlands, 1996.
7. Ralph Baker Kearfott. On proving existence of feasible points in equality constrained optimization problems. *Mathematical Programming*, 83(1, Ser. A):89–100, 1998.
8. Ralph Baker Kearfott. GlobSol: History, composition, and advice on use. In Christian Bliek et al., editors, *Global optimization and constraint satisfaction. First international workshop on global constraint optimization and constraint satisfaction, COCOS 2002, Valbonne-Sophia Antipolis, France, October 2 -4, 2002.* Springer-Verlag, Berlin, Germany / Heidelberg, Germany / London, UK / etc., 2003.
9. Ralph Baker Kearfott. GlobSol user guide. *Optimization Methods and Software*, 24(4-5):687–708, 2009.
10. Ralph Baker Kearfott. On rigorous upper bounds to a global optimum. *Journal of Global Optimization*, 59(2–3):459–476, July 2014.
11. Ralph Baker Kearfott and Kaisheng Du. The cluster problem in multivariate global optimization. *Journal of Global Optimization*, 5:253–265, 1994.
12. Ralph Baker Kearfott, Sowmya Muniswamy, Yi Wang, Xinyu Li, and Qian Wang. On smooth reformulations and direct non-smooth computations in global optimization for minimax problems. *Journal of Global Optimization*, 57(4):1091–1111, December 2013.
13. Günter Mayer. Epsilon-inflation in verification algorithms. *J. Comput. Appl. Math.*, 60(1-2):147–169, 1995.
14. Arnold Neumaier. Complete search in continuous global optimization and constraint satisfaction. In A. Iserles, editor, *Acta Numerica 2004*, pages 271–369. Cambridge University Press, 2004.
15. Arnold Neumaier. Global optimization (extensive web site), 2013. `http://www.mat.univie.ac.at/~neum/glopt.html`.
16. Siegfried M. Rump. INTLAB–INTerval LABoratory. In Csendes [1], pages 77–104. URL: `http://www.ti3.tu-harburg.de/rump/intlab/`.
17. Siegfried Michael Rump. Verification methods for dense and sparse systems of equations. In Herzberger [4], pages 63–136.
18. Nikolaos Vasili Sahinidis. BARON: A general purpose global optimization software package. *Journal of Global Optimization*, 8(2):201–205, 1996.
19. Hermann Schichl. *Mathematical Modeling and Global Optimization.* Cambridge monographs on applied and computational mathematics. Cambridge University Press, 2008.
20. Hermann Schichl, Mihály Csaba Markót, and Arnold Neumaier. Exclusion regions for optimization problems, 2013. accepted for publication; preprint available at `http://www.mat.univie.ac.at/~neum/ms/exclopt.pdf`.
21. Hermann Schichl and Arnold Neumaier. Exclusion regions for systems of equations. *SIAM Journal on Numerical Analysis*, 42(1):383–408, 2004.
22. Pascal Van Hentenryck, Laurent Michel, and Yves Deville. *Numerica: A Modeling Language for Global Optimization.* The MIT Press, Cambridge, MA, USA,

1997.

23. Ronald John Van Iwaarden. *An Improved Unconstrained Global Optimization Algorithm*. PhD thesis, University of Colorado at Denver, 1996.
24. Andreas Wächter. Homepage of IPOPT, 2002. `https://projects.coin-or.org/Ipopt`.