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

First Order Rejection Tests For Multiple-Objective Optimization

Alexandre Goldsztejn · Ferenc Domes · Brice Chevalier

the date of receipt and acceptance should be inserted later

Abstract Three rejection tests for multi-objective optimization problems based on first order optimality conditions are proposed. These tests can certify that a box does not contain any local minimizer, and thus it can be excluded from the search process. They generalize previously proposed rejection tests in several regards: Their scope include inequality and equality constrained smooth or non-smooth multiple objective problems. Reported experiments show that they allow quite efficiently removing the cluster effect in mono-objective and multi-objective problems, which is one of the key issues in continuous global deterministic optimization.

Keywords Multi-objective deterministic global optimization · first order optimality conditions · interval analysis · branch and bound algorithm · cluster effect

1 Introduction

We consider the nonlinear multi-objective problem of minimizing $f(x)$ subject to the constraints $g(x) \leq 0$ and $h(x) = 0$, where $f : \mathbb{R}^n \to \mathbb{R}^m$, $g : \mathbb{R}^n \to \mathbb{R}^p$ and $h : \mathbb{R}^n \to \mathbb{R}^q$ are Lipschitz continuous near every point of $\mathbb{R}^n$ (inequalities holding component wise). First oder optimality conditions, like Karush-John or Karush-Kuhn-Tucker conditions (see e.g. [4,26,36,29]), play a key role in the theory and practice of solving such nonlinear problems by standard numerical analysis methods. Karush-John conditions state that for any local Pareto optimal (or efficient, or non-dominated) solutions x_* of the previous problem, there exist multipliers

A. Goldsztejn
CNRS, LINA (UMR 6241), France, E-mail: Alexandre.Goldsztejn@gmail.com

F. Domes
Université de Nantes, LINA (UMR 6241), Nantes, France, E-mail: Ferenc.Domes@univie.ac.at

B. Chevalier
Université de Nantes, France, E-mail: Brice.Chevalier@gmail.com

$0 \leq \lambda \in \mathbb{R}^m$, $0 \leq r \in \mathbb{R}^p$ and $s \in \mathbb{R}^q$ not all 0 such that

$$\partial f(x_*)\,\lambda + \partial g(x_*)\,r + \partial h(x_*)\,s \ni 0 \tag{1}$$

$$(\forall 1 \leq i \leq p) \quad r_i\, g_i(x_*) = 0 \tag{2}$$

$$(\forall 1 \leq i \leq q) \quad h_i(x_*) = 0\ , \tag{3}$$

where ∂f is the generalized gradient of f[1]. See e.g. [4] for the definition of generalized gradients and related first order conditions. A normalization equation can be added since multipliers are not all 0, e.g.

$$\sum_{1\leq i\leq m} \lambda_i + \sum_{1\leq i\leq p} r_i + \sum_{1\leq i\leq q} s_i^2 = 1. \tag{4}$$

In the case where functions are differentiable, so generalized gradients coincide with gradients and (1) becomes an equality, the first order conditions (1)–(4) give rise to a system of $n+m+p+q$ unknowns and $n+p+q+1$ equations. Therefore, generically its solution set is a manifold of dimension $m-1$. This matches the generic dimension of the Pareto front of such a problem (noteworthily in the case of mono-objective optimization, Pareto optimality is equivalent to the usual order between reals, and the solution set is generically zero dimensional).

In the context of interval branch and bound algorithms, first order conditions allow rejecting parts of the search space that are difficult to reject with other techniques (in particular when exploring the region close to the global optimum where the so-called cluster effect[2] dramatically slows the search, see the discussions about the cluster effect in [6,34], and in Subsection 5.1 below). Their importance in branch and bound algorithm dedicated to mono-objective optimization is well-known, where the well-constrained system of equations (1)–(4) is solved using the interval Newton [18,12,24] or other techniques coming from numerical constraint programming [11]. This presents the drawbacks of solving a system with dimension higher than the original problem, one additional variable for each multiplier, and of requiring sharp initial domains for multipliers. This is balanced by the very sharp enclosure of computed local or global optima.

Subdivision methods (among which interval based branch and bound algorithms) recently started to be developed for solving nonlinear multi-objective problems [14,3,2,31,15,22,38,23,39]. Techniques related to first order conditions are used in [31,22,23] to reject parts of the search space that contain no Pareto optimal solution. As in the context of mono-objective optimization, the system (1)–(4) can be solved using constraints techniques. However, the additional difficulty is that the system (1)–(4) is under-constrained in the multi-objective case, while interval techniques like the interval Newton are efficient only for well-constrained systems: A componentwise Newton operator is used to solve (1)–(4) in [23] but this operator is much less efficient than the traditional Newton operator, which benefits from preconditioning techniques to tackle globally the system of equations.

Rejection tests like the one proposed in [31,22] do not solve explicitly the system (1)–(4) but instead use it to prove that a given region does not contain any

[1] In the case of vector valued functions $f = (f_1, \ldots, f_m)$, ∂f is a matrix whose columns are ∂f_i, so $\partial f(x_*)\,\lambda = \sum_i \lambda_i \partial f_i(x_*)$.

[2] Clusters of small boxes appear around local or global minimizers due to excessive splitting and failure to remove the resulting boxes because too close to these minimizers. This behavior is generic and one of the main issues in deterministic global optimization.

local optimum. The main disadvantage of such test is that they are useless when applied to domains that contain a local optimum, while solving (1)–(4) allows potentially reducing such a domain, hence providing a sharp enclosure of the contained optimum, without any additional subdivision. Nevertheless, rejection tests still allow efficiently tackling the cluster effect. Since they are computationally much cheaper than solving (1)–(4) and easy to implement, they can be used as a preliminary test before solving explicitly the system. They are even more important in the context of multi-objective optimization where traditional techniques for solving (1)–(4) are inefficient.

In this paper, three rejection tests based on first order necessary conditions are proposed. While they are more general and powerful than previously proposed tests [31,22], they are very simple and hence easily included in any branch and bound algorithm dedicated to mono or multi-objective optimization. Section 2 recalls the basics of interval analysis that will be used in Section 3 to develop the proposed rejection test. An analysis of related works is given in Section 4. Finally Section 5 reports experiments that show the usefulness of the proposed rejection tests for decreasing the cluster effect.

2 Interval Analysis

2.1 Interval Analysis

Interval analysis is a branch of numerical analysis that was born in the 1960's. It consists of computing with intervals of reals instead of reals, providing a framework for handling uncertainties and verified computations (see e.g. [27,1,28] and [17] for a survey). Interval analysis is a key ingredient for numerical constraint programming (see e.g. [16]) and global optimization (see e.g. [12,19]).

An interval is a closed connected subset of $\mathbb{R}$, the set of intervals being denoted by $\mathbb{IR}$ (which includes the empty set as well as unbounded intervals). Intervals are denoted by boldface symbols, e.g. $\mathbf{x} \subseteq \mathbb{R}$. There are two equivalent ways of defining interval vectors, the set of such n dimensional interval vectors being denoted by $\mathbb{IR}^n$. On the one hand, given two vectors $\underline{x} \leq \overline{x} \in \mathbb{R}^n$ (where the inequality is defined componentwise), an interval of vectors is obtained by considering $\mathbf{x} := \{x \in \mathbb{R}^n : \underline{x} \leq x \leq \overline{x}\}$. On the other hand, given intervals $\mathbf{x}_i \in \mathbb{IR}$ for $i \in \{1, \ldots, n\}$, a vector of intervals is obtained by considering $\mathbf{x} := \{x \in \mathbb{R}^n : \forall i \in \{1, \ldots n\}, x_i \in \mathbf{x}_i\}$. These two definitions are obviously equivalent following the notational convention $\underline{x} = (\underline{x}_i)$, $\overline{x} = (\overline{x}_i)$ and $[x_i] = [\underline{x}_i, \overline{x}_i]$, and will be used interchangeably. Interval matrices are defined similarly to interval vectors as either intervals of matrices or matrices of intervals, the set of $n \times m$ interval matrices being denoted by $\mathbb{IR}^{n \times m}$. A real number $x \in \mathbb{R}$ (respectively a real vector $x \in \mathbb{R}^n$ or a real matrix $A \in \mathbb{R}^{n \times m}$) will be identified with the degenerate interval $[x, x]$ (respectively a degenerated interval vector $[x, x]$ or a degenerated interval matrix $[A, A]$).

Operations $\circ \in \{+, \times, -, \div\}$ are extended to intervals in the following way: $\mathbf{x} \circ \mathbf{y} := \{x \circ y : x \in \mathbf{x}, y \in \mathbf{y}\}$ (which is an interval since these functions are continuous). The division is defined for intervals $[\underline{y}, \overline{y}]$ that do not contain zero. Unary elementary functions $f(x)$ like $\exp(x)$, $\ln(x)$, $\sin(x)$, etc., are also extended to intervals similarly: $f(\mathbf{x}) = \{f(x) : x \in \mathbf{x}\}$. All these elementary interval extensions form the interval arithmetic (IA). As real numbers are identified to de-

generated intervals, the IA actually generalizes the real arithmetic, and mixed operations like $1 + [1,2] = [2,3]$ are interpreted as interval operations, e.g. in this case $[1,1] + [1,2] = [2,3]$. An interval function $\mathbf{f} : \mathbb{IR}^n \longrightarrow \mathbb{IR}^m$ is an interval extension of the real function $f : \mathbb{R}^n \longrightarrow \mathbb{R}^m$ if for all $\mathbf{x} \in \mathbb{IR}^n$ we have $\mathbf{f}(\mathbf{x}) \supseteq \{f(x) : x \in \mathbf{x}\}$. Thus interval extensions allow computing enclosures of real functions range over boxes. So called natural interval extensions of a function are obtained by evaluating an expression of this function for interval arguments using the IA. In particular when every variable has one unique occurrence in the function's expression the natural interval extension is optimal, i.e. it computes the exact function range. However, when a function expression has several occurrences of some variable, its interval evaluation may be pessimistic. The pessimism of interval evaluation is one of the critical issue to be tackled when applying interval analysis. Finally, interval vectors and matrices arithmetic operations are also extended to intervals using IA, e.g. $\mathbf{A}\,\mathbf{x} = \mathbf{y}$ with $\mathbf{y}_i = \sum_k \mathbf{A}_{ik}\mathbf{x}_k$ with the enclosure property $\mathbf{y} \supseteq \{Ax : A \in \mathbf{A}, x \in \mathbf{x}\}$ (note that since the expression of y_i contains only one occurrence of each involved interval, $\mathbf{y}$ is actually the smallest interval vector satisfying this enclosure). Finally, given $\mathbf{A} \in \mathbb{IR}^{n\times m}$ and $\mathbf{b} \in \mathbb{IR}^n$, the united solution set $\{x \in \mathbb{R}^m : \exists A \in \mathbf{A}, \exists b \in \mathbf{b}, Ax = b\}$ is denoted by $\Sigma(\mathbf{A}, \mathbf{b})$ and is often called an interval linear system of equations.

Interval evaluations of the function derivatives or gradients are of particular interest in the present paper. Such interval evaluations can be performed whenever an explicit expression of the derivatives is available, or using automatic differentiation with IA [28,33,25]. Subgradients [4] are convex hulls of sets of representative gradients, and can also be enclosed inside interval evaluations. For example, in case the function expression contains some absolute value, the following rule can be used for enclosing its generalized gradient: $\partial\mathrm{abs}(\mathbf{x}) = -1$ if $\underline{x} \leq 0$, $\partial\mathrm{abs}(\mathbf{x}) = 1$ if $\overline{x} \geq 0$ and $\partial\mathrm{abs}(\mathbf{x}) = [-1,1]$ otherwise (see [28] for details).

Rounded Computations As real numbers are approximately represented by floating point numbers [9], the IA cannot match the real definitions of interval extensions exactly. In order to preserve the inclusion property, the IA has to be implemented using an outward rounding. For example, the exact definition of IA is $[1,3]/[10,10] = [0.1, 0.3]$, but both 0.1 and 0.3 cannot be exactly represented with standard floating point numbers. Therefore, the computed result will be $[0.1^-, 0.3^+]$ where 0.1^- (respectively 0.3^+) is a floating point number smaller than 0.1 (respectively greater than 0.3). We expect actually the greatest floating point number smaller than 0.1 and the smallest floating point number greater than 0.3 which is often achieved by IA implementations. Among numerous implementations of IA, we can cite the C/C++ libraries PROFIL/BIAS [20] and Gaol [10], the Matlab toolbox INTLAB [33] and Mathematica [41]. The developments presented in the rest of the paper use the ideal real IA. The algorithms are finally implemented using outwardly rounded floating point IA.

3 Rejection Tests Based On First Order Conditions

Given a box $\mathbf{x} \in \mathbb{IR}^n$, three rejection tests are proposed that allow proving that this box does not contain any local minimum, and hence needs not to be explored anymore by the branch and bound algorithm. Inactive inequality constraints don't

play any role in optimality check. It is therefore critical to identify them before applying any test (see Example 1 in Subsection 3.1). An inequality constraint $g_i(x) \leq 0$ is said active inside $\mathbf{x}$ if and only if there exists $x \in \mathbf{x}$ such that $g_i(x) = 0$. Some interval extensions $\mathbf{g}_i$ for each g_i allows identifying rigorously inequality constraints that are inactive inside $\mathbf{x}$, their indices being denoted by

$$I(\mathbf{x}) := \{i \in \mathbb{N} : 1 \leq i \leq p \wedge 0 \notin \mathbf{g}_i(\mathbf{x})\}. \tag{5}$$

Its complement

$$A(\mathbf{x}) := \{i \in \mathbb{N} : 1 \leq i \leq p \wedge 0 \in \mathbf{g}_i(\mathbf{x})\} \tag{6}$$

contains indices of inequality constraints that are potentially[3] active inside $\mathbf{x}$, the number of such constraint being denoted by $p_\bullet$. In the following, $\partial\mathbf{f}(\mathbf{x}) \in \mathbb{IR}^{n\times m}$, $\partial\mathbf{g}_\bullet(\mathbf{x}) \in \mathbb{IR}^{n\times p_\bullet}$ and $\partial\mathbf{h}(\mathbf{x}) \in \mathbb{IR}^{n\times q}$ are some interval enclosure over the box $\mathbf{x}$ of the generalized gradients objectives, potentially active inequality constraints and equality constraints respectively. Also, $\mathbf{G}(\mathbf{x}) \in \mathbb{IR}^{n\times(m+p_\bullet+q)}$ is the interval matrix made of all these interval vectors:

$$\mathbf{G}(\mathbf{x}) := \Big(\partial\mathbf{f}(\mathbf{x}) \mid \partial\mathbf{g}_\bullet(\mathbf{x}) \mid \partial\mathbf{h}(\mathbf{x})\Big). \tag{7}$$

Remark 1 When g includes bound constraints, some specific treatment may turn out to be more efficient than simply include them in $\mathbf{G}(\mathbf{x})$ (e.g. a pivoting strategy can start using this bound constraint gradient, see also Remark 3 in Subsection 4.1).

3.1 Full Column Rank Test

Theorem 1 *If* $\mathbf{G}(\mathbf{x})$ *is full column rank then* $\mathbf{x}$ *contains no local optimum.*

Proof Consider an arbitrary $x \in \mathbf{x}$ and some $0 \leq \lambda \in \mathbb{R}^m$, $0 \leq r \in \mathbb{R}^p$ and $s \in \mathbb{R}^q$ satisfying (1)–(3). For $i \in I(\mathbf{x})$, $0 \notin \mathbf{g}_i(\mathbf{x})$ so $g_i(x) \neq 0$ and hence (2) entails $r_i = 0$. Therefore (1) becomes

$$\partial f(x)\,\lambda + \partial g_\bullet(x)\,r_\bullet + \partial h(x)\,s \ni 0, \tag{8}$$

where $\partial g_\bullet(x) \in \mathbb{R}^{n\times p_\bullet}$ and $r_\bullet \in \mathbb{R}^{p_\bullet}$ are restrictions to potentially active inequality constraints. Now, (8) implies that there exists $(u|v|w) \in \big(\partial f(x)|\partial g_\bullet(x)|\partial h(x)\big) \subseteq \mathbf{G}(\mathbf{x})$ such that $u\lambda + vr_\bullet + ws = 0$. Since $\mathbf{G}(\mathbf{x})$ is fullrank, so is $(u|v|w)$, and therefore $u\lambda + vr_\bullet + ws = 0$ implies that all multipliers are zero, thus contradicting the Karush-John conditions. Therefore x is eventually not a local optimum. □

Remark 2 Note that $\mathbf{G}(\mathbf{x})$ is full column rank implies that $\big(\partial\mathbf{g}_\bullet(\mathbf{x}) \mid \partial\mathbf{h}(\mathbf{x})\big)$ is also full column rank. Therefore Theorem 1 can reject a box $\mathbf{x}$ only in the case where the constraints satisfy the linear independence constraint qualification in this box.

[3] These inequality constraints are only potentially active because interval extensions are generally pessimistic. All rejection tests proposed remain correct when a potentially active constraint is actually inactive, although they are more efficient as inactive constraints are more accurately detected.

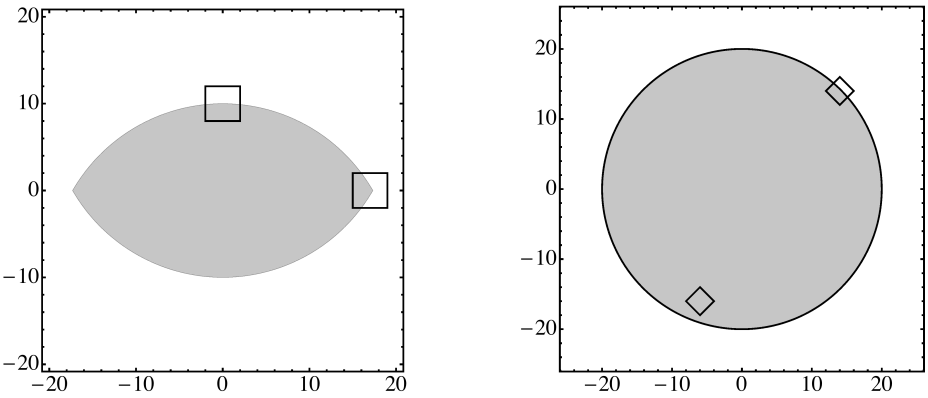

Fig. 1 Left: Feasible set and boxes involved in Example 1 and Example 2. Right: The image of the feasible set in the two objective space of Example 3 is the gray disk (among which the black circle satisfy the Fritz John conditions).

Checking if an interval matrix is full column rank is NP-hard (since checking the regularity of square interval matrices is NP-hard, see e.g. [30] and references therein). However, when the interval entries of the matrix are thin enough (which is generally the case when the cluster effect appears), two sufficient conditions can be used to check if $\mathbf{G}(\mathbf{x})$ is full column rank: The interval Gauss elimination (see e.g. [28,8]) and checking strict diagonal dominance of $C\mathbf{G}(\mathbf{x})$, where $C \in \mathbb{R}^{(m+p_\bullet+q)\times n}$ is e.g. some approximate midpoint pseudo-inverse preconditioner. Both methods complexity is cubic, while our experiments seem to show that the interval Gauss elimination is generally stronger. In Section 5, Theorem 1 is implemented using the interval Gauss elimination technique.

The following three examples illustrate the application of Theorem 1 to a differentiable mono-objective problem, a non differentiable mono-objective problem, and to a differentiable multi-objective problem respectively.

Example 1 Consider the problem of minimizing x_1 subject to $g_1(x) \leq 0$ and $g_2(x) \leq 0$ where $g_1(x) := x_1^2 + (x_2 - 10)^2 - 20^2$ and $g_2(x) := x_1^2 + (x_2 + 10)^2 - 20^2$, whose global minimum is attained at $(-10\sqrt{3}, 0)^T$. The feasible set of this problem is depicted on the left graphic of Figure 1.

Consider the box $\mathbf{x} = ([-2, 2], [8, 12])$, also depicted on Figure 1. In order to check constraints activity, we first consider the trivial interval extensions $\mathbf{g}_1(\mathbf{x}) = [-\infty, +\infty]$ and $\mathbf{g}_2(\mathbf{x}) = [-\infty, +\infty]$, that is we consider all constraints are potentially active. In this case, $\mathbf{G}(\mathbf{x})$ has two lines and three columns, and hence cannot be full column rank. Theorem 1 fails rejecting the box. Consider now the natural interval extensions of the constraints, that gives rise to $\mathbf{g}_1(\mathbf{x}) = [-400, -392]$ and $\mathbf{g}_2(\mathbf{x}) = [-76, 88]$. This proves that $g_1(x) \leq 0$ is not active (while g_2 is potentially

active, and even active since the interval evaluation is optimal), therefore

$$\mathbf{G}(\mathbf{x}) = \Big(\nabla\mathbf{f}(\mathbf{x}) \mid \nabla\mathbf{g}_2(\mathbf{x})\Big) = \begin{pmatrix} 1 & [-4,4] \\ 0 & [36,44] \end{pmatrix}. \tag{9}$$

This interval matrix being obviously full column rank, Theorem 1 now allows rejecting the box, hence the critical importance of preliminary checking the constraint activity.

Consider now the box $\mathbf{x} = ([15,19],[-2,2])$, also depicted on Figure 1. The natural interval extensions of the constraints give rise to the following enclosures: $\mathbf{g}_1(\mathbf{x}) = [-111,105]$ and $\mathbf{g}_2(\mathbf{x}) = [-111,105]$. Hence the two constraints are potentially active, while we can see on Figure 1 that they are actually both active inside $\mathbf{x}$. Thus Theorem 1 fails rejecting the box (a 2×3 matrix cannot be full column rank), which is normal since this box contains a local maximum that also satisfies the Karush-John conditions.

Example 2 Consider the problem of minimizing x_2 subject to $g(x) \leq 0$ where $g(x) = x_1^2 + (|x_2|+10)^2 - 20^2$. The feasible set of this problem is the same as in Example 1, and its optimum is attained at $(0,-10)^T$. As in Example 1, consider the box $\mathbf{x} = ([15,19],[-2,2])$. Then $\mathbf{g}(\mathbf{x}) = [-75,105]$ so the constraint is potentially active inside $\mathbf{x}$ (even active since the interval evaluation is optimal), so

$$\mathbf{G}(\mathbf{x}) = \Big(\nabla\mathbf{f}(\mathbf{x}) \mid \partial\mathbf{g}(\mathbf{x})\Big) = \begin{pmatrix} 0 & [30,38] \\ 1 & [-24,24] \end{pmatrix}. \tag{10}$$

The enclosure of the generalized gradient of g is computed as explained in Section 2. This interval matrix being obviously full column rank, Theorem 1 allows rejecting the box.

Example 3 Consider the bi-objective problem with three variables that consists in minimizing $f(x) = (x_1 + x_2, x_1 - x_2)$ subject to the constraint $x_1^2 + x_2^2 + x_3^2 \leq 200$. The image of the feasible set by the vectorial objective function is depicted on the right graphic of Figure 1, together with the image of the solutions that satisfy the Karush-John conditions in black, among which the Pareto optimal solutions.

Consider the box $\mathbf{x} = ([-12,-10],[4,6],[5,7])$. It's image in the objective space is depicted as the lower most parallelogram, which does not contain any Pareto optimal solution although close to the Pareto frontier (hence the box could be rejected by some rejection test). The interval evaluation of the constraint is $\mathbf{g}(\mathbf{x}) = [-59,29]$, hence it is potentially active (actually it is active since this interval evaluation is optimal). Hence

$$\mathbf{G}(\mathbf{x}) = \Big(\nabla\mathbf{f}_1(\mathbf{x}) \mid \nabla\mathbf{f}_2(\mathbf{x}) \mid \nabla\mathbf{g}(\mathbf{x})\Big) = \begin{pmatrix} 1 & 1 & [-24,-20] \\ 1 & -1 & [8,12] \\ 0 & 0 & [10,14] \end{pmatrix}. \tag{11}$$

This interval matrix is obviously full column rank (note that the inverse midpoint preconditioning is unable to prove $\mathbf{G}(\mathbf{x})$ is full rank because interval entries are too wide, while interval Gauss elimination is able to provided that the right pivoting strategy is used), hence the box can be rejected.

Consider the box $\mathbf{x} = ([13,15],[-1,1],[-1,1])$. It's image in the objective space is depicted as the upper most parallelogram, which contains some solutions of the

Karush-John conditions (hence the box should not be rejected by any rejection test). The interval evaluation of the constraint is $\mathbf{g}(\mathbf{x}) = [-31, 27]$, hence it is potentially active (actually it is active since this interval evaluation is optimal). Hence

$$\mathbf{G}(\mathbf{x}) = \Big(\nabla \mathbf{f}_1(\mathbf{x}) \mid \nabla \mathbf{f}_2(\mathbf{x}) \mid \nabla \mathbf{g}(\mathbf{x})\Big) = \begin{pmatrix} 1 & 1 & [26,30] \\ 1 & -1 & [-2,2] \\ 0 & 0 & [-2,2] \end{pmatrix}. \tag{12}$$

Since the last line of $\mathbf{G}(\mathbf{x})$ contains the null vector, $\mathbf{G}(\mathbf{x})$ is not full column rank and the rejection test does not reject it as expected.

3.2 Multipliers Sign Test

The second test uses the multiplier non-negativeness and is therefore restricted to problems with only inequality constraints.

Theorem 2 *If the problem has only inequality constraints (i.e. $q = 0$) and $\mathbf{G}(\mathbf{x})$ has a line where all interval entries don't contain zero and have the same sign then $\mathbf{x}$ contains no local optimum.*

Proof Consider an arbitrary $x \in \mathbf{x}$. Hence, $(\partial f(x) | \partial g_\bullet(x)) \subseteq \mathbf{G}(\mathbf{x})$ so the i^{th} line of $(\partial f(x) | \partial g_\bullet(x))$ has only strictly positive (respectively negative) entries. Since the vector (λ, r) is nonnegative, with at least a strictly positive component, the i^{th} component of $\partial f(x)\,\lambda + \partial g_\bullet(x)\,r_\bullet$ is strictly positive (respectively negative). Hence Equation (8) cannot hold, so Equation (1) does not hold neither, and x is not a local optimum. □

Example 4 Consider the last case of Example 1, i.e. the box $\mathbf{x} = ([15, 19], [-2, 2])$. Using the natural interval extension, we have seen that the two constraints are potentially active inside $\mathbf{x}$ (they are actually active inside $\mathbf{x}$) and thus using the natural interval extension of the gradients we obtain

$$\mathbf{G}(\mathbf{x}) = \begin{pmatrix} 1 & [30,38] & [30,38] \\ 0 & [-24,-16] & [16,24] \end{pmatrix}. \tag{13}$$

As seen in Example 1 it is not full column rank and thus Theorem 1 does not apply. However, its first line contains only strictly positive entries, and thus Theorem 2 allows rejecting the box.

Example 5 Consider the last case of Example 3. As seen in Example 3 it is not full column rank and thus Theorem 1 does not apply. However, its first line contains only strictly positive entries, and thus Theorem 2 allows rejecting the box.

Theorem 2 has a different aim from Theorem 1: While the latter aims at rejecting boxes close to local minima (hence helping fighting the cluster effect), Theorem 2 will allow rejecting boxes that contain e.g. maxima. In the context of mono-objective optimization, the usual strategy that consists in exploring regions where the objective function is potentially low does lower the impact of Theorem 2. However, its usefulness in the context of multiple objective optimization could turn out to be greater (as pointed out e.g. in [22] where a weaker test is used, see Subsection 4.1).

3.3 Multipliers Domain Test

The preconditioning method for checking the rank of $\mathbf{G}(\mathbf{x})$ can be adapted to enclose the multipliers domains in view of applying an interval Newton operator to the first order conditions system when the matrix $\mathbf{G}(\mathbf{x})$ turns out to be non full column rank. We introduce the interval matrix $\mathbf{G}_*(\mathbf{x}) \in \mathbb{IR}^{(n+1)\times(m+p_\bullet+q)}$ where one last line made of 1 and 0 is added to $\mathbf{G}(\mathbf{x})$:

$$\mathbf{G}_*(\mathbf{x}) := \left(\begin{array}{c|c|c} \partial\mathbf{f}(\mathbf{x}) & \partial\mathbf{g}_\bullet(\mathbf{x}) & \partial\mathbf{h}(\mathbf{x}) \\ \hline 1^T & 0^T & 0^T \end{array}\right), \tag{14}$$

where 1^T and 0^T are horizontal vectors made of 1 and 0 respectively. This last line intends normalizing the multipliers λ by $\sum \lambda_i = 1$, which is a valid normalization provided that $\big(\partial\mathbf{g}_\bullet(\mathbf{x})|\partial\mathbf{h}(\mathbf{x})\big)$ is full rank. This is formalized in the following theorem.

Theorem 3 *Let $k = p_\bullet + q$, $C \in \mathbb{R}^{(m+k)\times(n+1)}$ be a matrix[4], and define $\mathbf{A} := C\,\mathbf{G}_*(\mathbf{x})$. Consider the following block representations of these matrices:*

$$C = \begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix} \quad \textit{and} \quad \mathbf{A} = \begin{pmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{pmatrix} \tag{15}$$

with $C_{11} \in \mathbb{R}^{m\times n}$, $C_{12} \in \mathbb{R}^{m\times 1}$, $C_{21} \in \mathbb{R}^{k\times n}$, $C_{22} \in \mathbb{R}^{k\times 1}$, $\mathbf{A}_{11} \in \mathbb{IR}^{m\times m}$, $\mathbf{A}_{12} \in \mathbb{IR}^{m\times k}$, $\mathbf{A}_{21} \in \mathbb{IR}^{k\times m}$ and $\mathbf{A}_{22} \in \mathbb{IR}^{k\times k}$, so that

$$\mathbf{A}_{11} = C_{11}\,\partial\mathbf{f}(\mathbf{x}) + C_{12}\,1^T \tag{16}$$
$$\mathbf{A}_{12} = C_{11}\,(\partial\mathbf{g}(\mathbf{x}) \mid \partial\mathbf{h}(\mathbf{x})) \tag{17}$$
$$\mathbf{A}_{21} = C_{21}\,\partial\mathbf{f}(\mathbf{x}) + C_{22}\,1^T \tag{18}$$
$$\mathbf{A}_{22} = C_{21}\,(\partial\mathbf{g}(\mathbf{x}) \mid \partial\mathbf{h}(\mathbf{x})). \tag{19}$$

Note that $C_{12}\,1^T$ and $C_{22}\,1^T$ are actually vector outer products. Suppose that $\mathbf{A}_{22}$ is strictly diagonally dominant. Then:

(i) *$\big(\partial\mathbf{g}_\bullet(\mathbf{x}) \mid \partial\mathbf{h}(\mathbf{x})\big)$ is full column rank, and any local minimizer inside $\mathbf{x}$ satisfy the Karush-John conditions with multipliers $(\lambda, r_\bullet, s)$ satisfying $\sum_{1\le i\le m} \lambda_i = 1$, called normalized multipliers below (multipliers of inactive inequality constraints being 0).*

(ii) *The normalized multipliers $(\lambda, r_\bullet, s)$ belong to both united solution sets $\Sigma(\mathbf{G}_*, e) \subseteq \Sigma(\mathbf{A}, Ce)$, with $e = (0, \ldots, 0, 1)^T \in \mathbb{R}^{n+1}$.*

(iii) *The normalized multipliers λ of the objectives have the initial domains $\boldsymbol{\lambda}_i = [0, 1]$. When $m = 1$, the domain of the single objective multiplier is reduced to $\boldsymbol{\lambda}_1 = [1, 1]$. The normalized multipliers $(r_\bullet, s)$ of the constraints have the initial domains $(\mathbf{r}_\bullet, \mathbf{s}) = C_{22} + \|C_{22} - \mathbf{A}_{21}\boldsymbol{\lambda} - \mathbf{A}_{22}C_{22}\|_\infty(\pm 1)$, where $(\pm 1) \in \mathbb{IR}^k$ is the interval vector whose components are $[-1, 1]$. Furthermore, the initial domains $\mathbf{r}_i$ of inequality constraints multiplier are intersected with $[0, +\infty[$.*

[4] Typically, an approximate generalized inverse of the midpoint of $\mathbf{G}_*(\mathbf{x})$.

(iv) The normalized multipliers domains can be improved contracting them by applying the interval Gauss-Seidel iteration to the united solution sets $\Sigma(\mathbf{G}_*, e)$ and $\Sigma(\mathbf{A}, Ce)$[5]. When the Gauss-Seidel iteration proves that one of the the united solution sets is empty, the box $\mathbf{x}$ does not contain any local minimizer, and hence can be rejected.

Proof From the block representation of the product $C\,\mathbf{G}(\mathbf{x})$, we see that $C\,\mathbf{G}(\mathbf{x})$ strictly diagonally dominant entails that $\mathbf{A}_{22} = C_{21}\,(\partial\mathbf{g}(\mathbf{x})\ \partial\mathbf{h}(\mathbf{x}))$ is also strictly diagonally dominant. Therefore, $(\partial\mathbf{g}(\mathbf{x})\ \partial\mathbf{h}(\mathbf{x}))$ is full column rank. Now, any local minimizer inside $\mathbf{x}$ satisfies the Karush-John conditions, i.e. $G\,(\lambda, r_\bullet, s)^T = 0$ for some $G \in \mathbf{G}(\mathbf{x})$, the multipliers being not all 0 and $\lambda, r_\bullet \geq 0$. Since $(\partial\mathbf{g}(\mathbf{x})\ \partial\mathbf{h}(\mathbf{x}))$ is full column rank, all λ_i cannot be zero (otherwise all multipliers would be 0 contradicting the Karush-John conditions), and since they are nonnegative we have $\sum_{1\leq i\leq m} \lambda_i > 0$. Therefore, all multipliers can be scaled by $(\sum_{1\leq i\leq m} \lambda_i)^{-1}$, and any local minimizer inside $\mathbf{x}$ has some multipliers that satisfy $\sum_{1\leq i\leq m} \lambda_i = 1$, hence (i).

Normalized multipliers satisfy $G_*\,(\lambda, r_\bullet, s)^T = e$ for some $G_* \in \mathbf{G}_*(\mathbf{x})$, that is $(\lambda, r_\bullet, s)^T \in \Sigma(\mathbf{G}_*, e)$. Preconditioning this system with C gives rise to $\Sigma(\mathbf{G}_*, e) \subseteq \Sigma(\mathbf{A}, c)$, hence (ii).

Since normalized multipliers satisfy $\sum_{1\leq i\leq m} \lambda_i = 1$ and $\lambda_i \geq 0$, we have $\lambda_i \in [0, 1]$. In the case of mono-objective problems, the normalization equation becomes $\lambda_1 = 1$ and its domains is obviously reduced to $[1, 1]$. Now, from the block decomposition of the linear system $\Sigma(\mathbf{A}, Ce)$, the multipliers (r, s) also belong to $\Sigma(\mathbf{A}_{22}, C_{22} - \mathbf{A}_{21}\boldsymbol{\lambda})$. By Lemma 4.2.1 of [28] (with $\tilde{x} = C_{22}$), we can translate the solution set as follows: $\Sigma(\mathbf{A}_{22}, C_{22} - \mathbf{A}_{21}\boldsymbol{\lambda}) \subseteq C_{22} + \Sigma(\mathbf{A}_{22}, C_{22} - \mathbf{A}_{21}\boldsymbol{\lambda} - \mathbf{A}_{22}C_{22})$. Finally, Proposition 4.1.9 of [28] (with $C = C' = I$ and $u = v = 1$) provide the enclosure of (r, s) given in (iii). Since the multipliers of the inequality constraints are non-negative, their initial domains can be intersected with $[0, +\infty[$.

Finally, (iv) is obvious since by (ii) normalized multipliers belong to the given linear interval equations, while the interval Gauss-Seidel iteration reduces domains keeping all solutions of these systems. □

When C is an approximation of some midpoint pseudo-inverse of $\mathbf{G}(\mathbf{x})_*$, the interval matrix $C\,\mathbf{G}_*(\mathbf{x})$ is approximately centered on the identity matrix, therefore so is $\mathbf{A}_{22}$ and $\mathbf{A}_{21}$ is centered on 0. As a consequence, the error bound $\|C_{22} - \mathbf{A}_{21}\boldsymbol{\lambda} - \mathbf{A}_{22}C_{22}\|_\infty$ is expected to be small provided that the interval entries of $\mathbf{G}$ are thin enough, hence a good enclosure of the multipliers domains for small enough boxes $\mathbf{x}$. Note that the block representation (16)–(19) needs not to be explicitly formed in practice: The strict diagonal dominance of $\mathbf{A}_{22}$ and the error bound $\|C_{22} - \mathbf{A}_{21}\boldsymbol{\lambda} - \mathbf{A}_{22}C_{22}\|_\infty$ can be both computed directly using $\mathbf{A}$. The following three examples show the application of Theorem 3 to the problems introduced in Example 1, Example 2 and Example 3 respectively.

Example 6 Consider the problem of Example 1 with the first box, so the interval matrix $\mathbf{G}_*(\mathbf{x})$ is the interval matrix $\mathbf{G}(\mathbf{x})$ given in Equation (9) with one additional row (1 0). The midpoint Moore-Penrose pseudo-inverse C, and the corresponding

[5] Note that $\Sigma(\mathbf{A}, Ce)$ is preconditioned so the Gauss-Seidel iteration needs solving only diagonal entries of $\mathbf{A}$. On the other hand, $\Sigma(\mathbf{G}_*, e)$ is not preconditioned so all entries of $\mathbf{G}_*$ need to be solved. This can be efficiently performed using inner subtraction.

preconditioned interval matrix $\mathbf{A}$ are

$$C = \left(\begin{array}{c|cc} 0.5 & 0 & 0.5 \\ \hline 0 & 0.025 & 0 \end{array}\right) \quad \text{and} \quad \mathbf{A} = \left(\begin{array}{c|c} 1 & [-2,2] \\ \hline 0 & [0.9,1.1] \end{array}\right), \tag{20}$$

where the block decompositions used in Theorem 3 are displayed. The submatrix $\mathbf{A}_{22}$, which is here a 1×1 interval matrix, is strictly diagonally dominant. Hence, the initial domain of λ_1 is $\boldsymbol{\lambda}_1 = [1, 1]$, and we proceed by computing $\|C_{22} - \mathbf{A}_{21}\boldsymbol{\lambda} - \mathbf{A}_{22}C_{22}\|_\infty$ which is here equal to 0. Hence the initial domain of r_1 is $\mathbf{r}_1 = [0, 0]$. Finally, applying one iteration of the interval Gauss-Seidel to the preconditioned system $\Sigma(\mathbf{A}, Ce)$ proves that there is no normalized multipliers, and thus no local minimum in $\mathbf{x}$.

Now consider the second box of Example 1, so $\mathbf{G}_*(\mathbf{x})$, C and $\mathbf{A}$ are respectively approximately

$$\left(\begin{array}{ccc} 1 & [30,38] & [30,38] \\ 0 & [-24,-16] & [16,24] \\ 1 & 0 & 0 \end{array}\right), \quad \left(\begin{array}{c|cc} 0 & 0 & 1 \\ \hline 0.0147 & -0.025 & -0.0147 \\ 0.0147 & 0.025 & -0.0147 \end{array}\right) \quad \text{and} \quad \left(\begin{array}{c|cc} 1 & 0 & 0 \\ \hline 0 & [0.841,1.159] & [-0.159,0.159] \\ 0 & [-0.159,0.159] & [0.841,1.159] \end{array}\right). \tag{21}$$

The submatrix $\mathbf{A}_{22}$, which is here a 2×2 interval matrix, is strictly diagonally dominant. Hence, the initial domain of λ_1 is $\boldsymbol{\lambda}_1 = [1, 1]$, and we proceed by computing $\|C_{22} - \mathbf{A}_{21}\boldsymbol{\lambda} - \mathbf{A}_{22}C_{22}\|_\infty$ which is approximately equal to 0.00467. The initial domains of r_1 and r_2 are respectively $\mathbf{r}_1 = \mathbf{r}_2 = [-0.0194, -0.0100]$. Finally, since the normalized multipliers domain are strictly negative, the box can be rejected (it actually contains a local maximum, as seen on Figure 1).

Example 7 Consider the problem of Example 2, so the interval matrix $\mathbf{G}_*(\mathbf{x})$ is the interval matrix $\mathbf{G}(\mathbf{x})$ given in Equation (10) with one additional row (1 0). The midpoint Moore-Penrose pseudo-inverse C, and the corresponding preconditioned interval matrix $\mathbf{A}$ are

$$C = \left(\begin{array}{c|cc} 0 & 0.5 & 0.5 \\ \hline 0.294 & 0 & 0 \end{array}\right) \quad \text{and} \quad \mathbf{A} = \left(\begin{array}{c|c} 1 & [-12,12] \\ \hline 0 & [0.882,0.118] \end{array}\right). \tag{22}$$

The submatrix $\mathbf{A}_{22}$, which is here a 1×1 interval matrix, is strictly diagonally dominant. Hence, the initial domain of λ_1 is $\boldsymbol{\lambda}_1 = [1, 1]$, and we proceed by computing $\|C_{22} - \mathbf{A}_{21}\boldsymbol{\lambda} - \mathbf{A}_{22}C_{22}\|_\infty$ which is here equal to 0. Hence the initial domain of r_1 is $\mathbf{r}_1 = [0, 0]$. Finally, applying one iteration of the interval Gauss-Seidel to the preconditioned system $\Sigma(\mathbf{A}, Ce)$ proves that there is no normalized multipliers, and thus no local minimum in $\mathbf{x}$.

Example 8 Consider the problem of Example 3 with the first box, so the interval matrix $\mathbf{G}_*(\mathbf{x})$ is the interval matrix $\mathbf{G}(\mathbf{x})$ given in Equation (11) with one additional row (1 1 0). The midpoint Moore-Penrose pseudo-inverse C, and the corresponding preconditioned interval matrix $\mathbf{A}$ are

$$C = \left(\begin{array}{ccc|c} 0.236 & 0.5 & 0.0155 & 0.264 \\ -0.0492 & -0.5 & 0.326 & 0.549 \\ \hline -0.0285 & 0 & 0.0311 & 0.0285 \end{array}\right) \quad \text{and} \quad \mathbf{A} = \left(\begin{array}{cc|c} 1 & 0 & [-1.51,1.51] \\ 0 & 1 & [-1.76,1.76] \\ \hline 0 & 0 & [0.880,1.120] \end{array}\right). \tag{23}$$

The submatrix $\mathbf{A}_{22}$, which is here a 1×1 interval matrix, is strictly diagonally dominant. Hence, the initial domain of λ is $\boldsymbol{\lambda} = ([0, 1], [0, 1])$, and we proceed

by computing $\|C_{22} - \mathbf{A}_{21}\boldsymbol{\lambda} - \mathbf{A}_{22}C_{22}\|_\infty$ which is here approximately equal to 0.0034. Hence the initial domain of r_1 is $\mathbf{r}_1 = [0.0251, 0.0319]$. Finally, applying one iteration of the interval Gauss-Seidel to the preconditioned system $\Sigma(\mathbf{A}, Ce)$ reduces the domains to $(\boldsymbol{\lambda}, \mathbf{r}_1) = ([0.216, 0.313], [0.493, 0.606], [0.0254, 0.0319])$, and the next application of the interval Gauss-Seidel iteration to $\Sigma(\mathbf{G}_*, e)$ proves the emptiness of the multipliers domain, allowing rejecting $\mathbf{x}$.

Now consider the second box of Example 3, so the interval matrix $\mathbf{G}_*(\mathbf{x})$ is the interval matrix $\mathbf{G}(\mathbf{x})$ given in Equation (12) with one additional row (1 1 0). The midpoint Moore-Penrose pseudo-inverse C, and the corresponding preconditioned interval matrix $\mathbf{A}$ are

$$C = \left(\begin{array}{ccc|c} 0 & 0.5 & 0 & 0.5 \\ 0 & -0.5 & 0 & 0.5 \\ \hline 0.0357 & 0 & 0 & -0.0357 \end{array}\right) \quad \text{and} \quad \mathbf{A} = \left(\begin{array}{cc|c} 1 & 0 & [-1,1] \\ 0 & 1 & [-1,1] \\ \hline 0 & 0 & [0.928,1.072] \end{array}\right). \tag{24}$$

The submatrix $\mathbf{A}_{22}$, which is again here a 1×1 interval matrix, is strictly diagonally dominant. Hence, the initial domain of λ is $\boldsymbol{\lambda}_1 = ([0, 1], [0, 1])$, and we proceed by computing $\|C_{22} - \mathbf{A}_{21}\boldsymbol{\lambda} - \mathbf{A}_{22}C_{22}\|_\infty$ which is here approximately equal to 0.0025. Hence the initial domain of r_1 is $\mathbf{r}_1 = [-0.0383, -0.0331]$. Finally, since the normalized multipliers domain are strictly negative, the box can be rejected (it actually contains Pareto solutions for the maximization counterpart of the problem, as seen on Figure 1).

4 Related Work

4.1 Monotonicity Test For Multiple-Objective Optimization

A simple extension of the mono-objective monotonicity test to multi-objective optimization was proposed in [22]. Using the notations introduced here, the test proposed in [22] is restricted to inequality constraints (i.e. $q = 0$) and consists in rejecting a box if $p_\bullet = 0$ (i.e. there is no active inequality constraint) and one line of $\mathbf{G}(\mathbf{x})$ has entries that don't contain zero and have the same sign. Therefore, Theorem 2 generalizes the rejection test of [22] to possibly active constraints. This generalization is critical as seen on the typical Example 4 where the test of [22] does not allow rejecting the box. Note finally that the connection between the monotonicity test and the sign of the Lagrange multipliers was not mentioned in [22].

Remark 3 The monotonicity test generally also includes some specific treatment for bound constraints $a \leq x_i \leq b$ that consists in reducing the domain $\mathbf{x}_i$ of the variable x_i to one of its bounds whenever the interval evaluation $\nabla\mathbf{f}_i(\mathbf{x})$ of the i^{th} component of the gradient has a fixed sign.

4.2 Interval Differential Formulation

A rejection test, called interval differential formulation, was proposed in [31]. Its scope is unconstrained multiple-objective optimization with an arbitrary number of variables, although experiments in [31] are restricted to bi-objective problems

with two variables. Informally, it consists in studying interval evaluations of the gradients over a given box, aiming proving that some descent direction for all objectives exists. Then this box can be rejected since all objectives can be improved simultaneously starting from any point inside the box. Although not noted in [31], this is a direct consequence of unconstrained first order conditions. From a computational point of view, in the case of two objectives and two variables, such regions are found intersecting interval angles of the interval evaluations of the gradients. In the case of more than two variables, projections in two variables spaces are proposed to be performed, although not formally justified[6]. The case of more than two objectives is not detailed in [31]. Bound constraints are handled in [31] by a preliminary partition of the domain into an interior part and $2n$ boundary parts. Since the interval Gauss elimination is optimal for checking the linear independence of two interval vectors, Theorem 1 is strictly stronger in this case, while it has a much wider scope since it tackles constrained multi-objective problems. Noteworthily, this rejection test together with the branch and bound algorithm proposed in [31] are patented, see [32].

4.3 Monotonicity Test Improvement for Mono-Objective Optimization

A rejection test, called the modified monotonicity test (MMT), that cannot be derived using first order conditions has been proposed in [21], in the restricted case of inequality constrained mono-objective optimization. The pseudo code given in Section 4 of [21] consists in rejecting a box $\mathbf{x}$ provided that each constraint $g_j(x) \leq 0$ is either, inactive inside the box (i.e. $\mathbf{g}_j(\mathbf{x}) < 0$), or is independent with respect to a variable x_i (i.e. $\frac{\partial \mathbf{g}_j}{\partial x_i}(\mathbf{x}) = 0$) and the objective function is strictly monotonic with respect to this variable (i.e. $0 \notin \frac{\partial \mathbf{f}}{\partial x_i}(\mathbf{x})$). This property is incorrect, as shown by the following counter example: Consider the problem of minimizing $f(x) = (x_1 - 1)^2 + x_2^2$ subject to $g(x) \leq 0$ with $g(x) = 0$ if $x_1 \leq 0$ and $g(x) = x_1^2$ otherwise (note that g is smooth). The global optimum of this problem is $x_* = (0, 0)^T$. Consider the box $\mathbf{x} = ([-1, 0], [-1, 1])^T$. Since $\frac{\partial \mathbf{f}}{\partial x_1}(\mathbf{x}) = [-3, -1] < 0$ and $\frac{\partial \mathbf{g}}{\partial x_1}(\mathbf{x}) = 0$, the MMT rejects this box although it contains the global optimum.

An alternative MMT was proposed by the author of [21] in a personal communication: The property holds if a slightly different assumptions are fulfilled: The derivative $\frac{\partial \mathbf{g}_j}{\partial x_i}(\mathbf{x})$ is equal to zero not only in the box $\mathbf{x}$, but in a superbox, e.g., $[\overline{x} - \varepsilon, \underline{x} + \varepsilon]$. This is, in particular, fulfilled, if the derivatives are equal to zero in the whole domain, i.e., if the formulae for g_j do not contain the variable x_i. Such a test and first order condition rejection tests are complementary.

4.4 Multipliers Domain Enclosures for Mono-Objective Optimization

Two different methods for computing bounds on the multipliers in the mono-objective case are proposed in [13]. The first uses the interval Newton applied to

[6] Although not noted in [31], the angle between two gradients interval evaluations $\mathbf{g}_1$ and $\mathbf{g}_2$ can be proved not to contain π simply by checking that the scalar product $\mathbf{g}_1\,\mathbf{g}_2$ does not intersect $||\mathbf{g}_1||\,||\mathbf{g}_2||$. This is sufficient for rejecting the box, and easily computed for arbitrary dimensions.

the first order system of equations. By using a specific variable ordering and a sharp interval evaluation of the gradients over subdomains, the method of [13] applies the interval Newton without any initial domain for multipliers, and computes some domains using the interval Gauss elimination as an interval linear solver for the interval Newton. The second is similar to Theorem 3 in the sense that it tackles the first order system as an over constrained linear system for the multipliers. However, both methods proposed in [13] use a different normalization equation from the one proposed here: In [13] multipliers are normalized using the interval linear equation

$$\lambda_1 + \sum_{1\leq i\leq p_\bullet} r_i + \sum_{1\leq i\leq q} Es_i = 1, \tag{25}$$

with $E = [1, 1+\epsilon]$ where ϵ is the smallest number such that $1+\epsilon$ is strictly greater than 1 in the machine number representation. The rational is that if $E = 1$ then the normalization equation would be incompatible with some problems, hence leading to loosing some minimizers in these cases. Consider e.g. the problem of minimizing x_1 under the constraint $0.5-0.5x_1^2-0.5x_2^2 = 0$ whose minimum is $x = (-1, 0)$ with multipliers satisfying $\lambda_1 + s_1 = 0$. These multipliers are incompatible with the normalization equation $\lambda_1+s_1 = 1$, which would entail rejecting this minimizer. By using $\lambda_1+Es_1 = 1$, this local minimizer is not rejected. However, this normalization equation does not allow computing any useful multipliers domain in this case, while the normalization equation used in Theorem 3 allows computing sharp multipliers domains in this case too. Note that the normalization equation used here rely on the fact that constraints gradients interval evaluations are linearly independent, which is proved during the test. Although the normalization equation used in [13] does not rely on this constraint qualification, their methods also fails computing multipliers domains when it does not hold (in fact, multipliers domains would be useless when no constraint qualification hold since the interval Newton method would fail in this case).

4.5 Exclusion Regions for Systems of Equations

Exclusion regions (see [34] and references therein) consists in building a region around a solution of a system of equations where no other solution is proved to lie. They therefore allow efficiently fighting the cluster effect, which also appears when solving systems of equations. In the context of global optimization, an exclusion region for the first order conditions is built. The exclusion region is built only one time when the local optimum is found, which is an advantage with respect to the rejection tests proposed here that have to be checked at each node of the search tree. On the other hand, exclusion regions require third order derivatives of the objective function and constraints and have a quadratic complexity with respect to the number of variables, while the rejection test proposed here uses only first order derivatives and have a cubic complexity. Noteworthily, exclusion regions are quite difficult to implement contrarily to rejection tests.

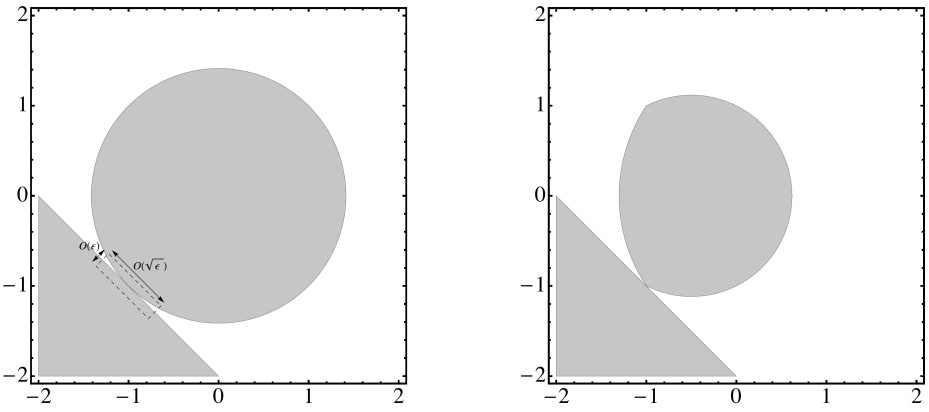

Fig. 2 The solution sets for $n = 2$ of the constraints from Subsection 5.1 and Subsection 5.1 respectively (in the left graphic, the dashed rectangle represent the cylinder where a cluster effect is expected).

5 Experiments

Experiments reported in this section are restricted to mono-objective problems, because of the lack of implementation of branch and bound algorithms dedicated to multi-objective problems. A smooth and a non smooth mono-objective academic problems are investigated in Subsection 5.1 and Subsection 5.2 respectively, as well as a smooth bi-objective academic problem in Subsection 5.3. Finally, the improvement brought by the rejection tests are illustrated on a standard difficult mono-objective problem in Subsection 5.4.

5.1 Smooth Mono-Objective Academic Problem

We consider the problem of minimizing $f(x) := \sum_i x_i$, $x \in \mathbb{R}^n$ subject to the constraint $g(x) := \sum_i x_i^2 - n \leq 0$. The global optimum $x_* = (-1, \ldots, -1)^T$ is easily found using a local solver, hence the branch and bound algorithm consists in solving

$$f(x) \leq f(x_*) \quad \text{and} \quad g(x) \leq 0 \tag{26}$$

(the solution sets of these two constraints are depicted in the left hand side graphic of Figure 2). In the neighborhood of the global optimum of this problem, the minimal distance $d(x, y)$ between a feasible point x and its closest point y that satisfies $f(y) \leq f(x_*)$ behaves like $d(x, y) = O(d(x, x_*)^2) = O(d(y, x_*)^2)$. Interval techniques will be able to reject a box $\mathbf{x}$ provided that at least one of the constraints is false inside it. Hence, we expect that using a criteria for stoping the search when a box is smaller than ϵ should output a paving of such boxes that covers approximately a cylinder with a base formed of a $(n-1)$-sphere of radius $O(\epsilon^{\frac{1}{2}})$ (of measure $O(\epsilon^{\frac{n-1}{2}})$), and a height of $O(\epsilon)$ (see the dashed cylinder depicted in the left hand side graphic of Figure 2). Such a cylinder has a measure of $O(\epsilon^{1+\frac{n-1}{2}})$ and this should result in a cluster of $O(\epsilon^{1+\frac{n-1}{2}}\, \epsilon^{-n}) = O(\epsilon^{-\frac{n-1}{2}})$ boxes of size ϵ

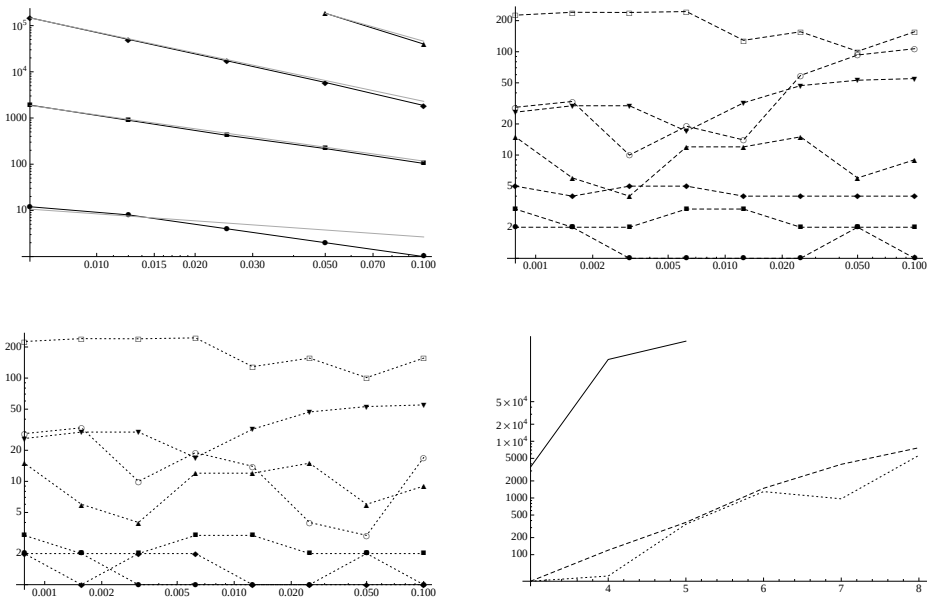

Fig. 3 Smooth mono-objective academic problem. Number of boxes of size ϵ, plotted with respect to ϵ with no rejection test (upper left, for $n \in \{2,\ldots,5\}$), the multipliers sign and full rank test (upper right, for $n \in \{2,\ldots,8\}$), and the multipliers sign and multipliers domain test (lower left, for $n \in \{2,\ldots,8\}$). Different markers represent results for different dimensions. In the upper left graphic, the light gray lines represent the fitted cluster effect model. The lower right graphic shows the number of splits needed to reach boxes of size $\epsilon = 0.006$ with respect to the problem dimension, using no rejection test (full line), the full rank test (dashed line) and the domain enclosure test (dotted line).

(called ϵ-boxes). This asymptotic analysis[7] is pretty well confirmed by experiments carried out using the GloptLab [5][8]: The upper left graphic of Figure 3 shows the number of ϵ-boxes computed for various values of ϵ without any rejection test, the light gray lines corresponding to the asymptotic models. It clearly shows the computed boxes follow very accurately the cluster effect model[9], which makes this academic problem unsolvable using a branch and bound algorithm for even quite small values of n.

The upper right and lower left graphics of Figure 3 show the same graphics in the case where first order rejection tests are used. It clearly shows that for a fixed n, the number of computed ϵ-boxes is drastically decreased, and even does not depend on ϵ anymore: The cluster effect have been canceled allowing tackling problems

[7] The asymptotic analyses of the cluster effect provided in [6], in the context of unconstrained optimization, or in [34], in the context of a system of equations, lead to different models that do not hold here. In particular both [6] and [34] consider some pessimistic interval evaluations, while this academic problem suffers from the cluster effect in spite of exact interval evaluations.

[8] GloptLab implements here a branch and prune algorithm based on constrain propagation on DAGs [35,40].

[9] Experiments non reported here have shown that the constraint propagation can remove the cluster effect when $n = 2$, although the closer to the optimum the slower the convergence of the propagation, converging to infinitely slow convergence (which requires very expensive constraint propagation). This is generic in two variables, but not in higher dimensions where the constraint propagation is not able anymore to remove the cluster effect.

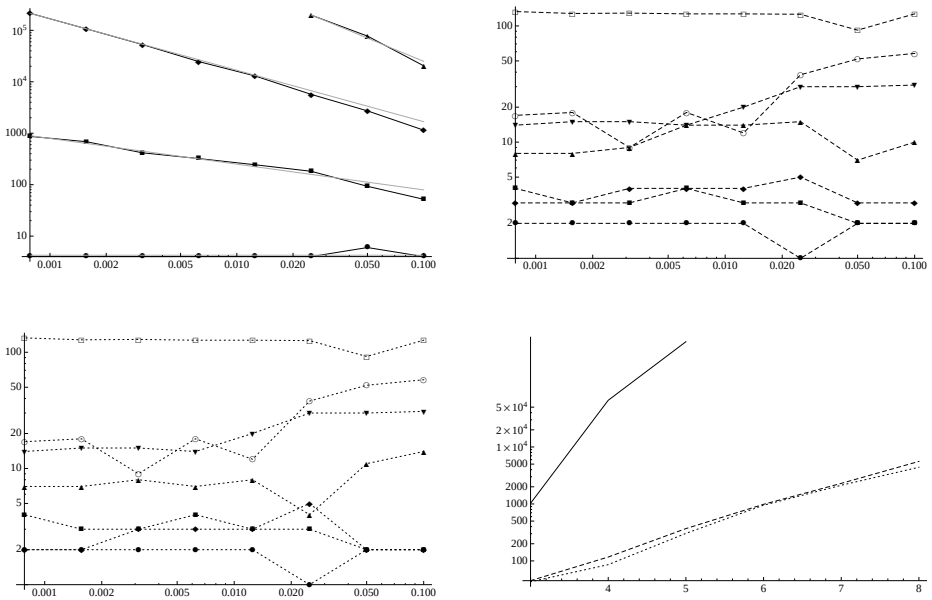

Fig. 4 Nonsmooth mono-objective academic problem. Same graphics as in Figure 3.

with much higher values of n and with much shaper accuracy. These graphics seem to show that for a fixed ϵ the number of ϵ-boxes increases exponentially with respect to n. The lower right graphic of Figure 3 shows the number of splits needed to reach boxes of size $\epsilon = 0.006$ with respect to the problem dimension. It clearly shows an exponential dependence for both the full rank and multipliers domain tests, which would require fully solving the first order optimality conditions to be removed (e.g. with an interval Newton operator).

Remark 4 The asymptotic analysis carried out for this academic problem can be generalized to arbitrarily constrained problems, leading to

$$O(\epsilon^{-\frac{n-m}{2}}), \tag{27}$$

where m is the number of active constraints at the considered strict minimizer (including equality constraints). This analysis holds only under LICQ and if the objective function gradient is not null at the local constrained minimizer (otherwise the exact interval evaluation of the objective function will allow removing the cluster effect).

5.2 Nonsmooth Mono-Objective Academic Problem

The problem of minimizing $f(x) := \sum_i x_i$, $x \in \mathbb{R}^n$ subject to the constraint $g(x) := \sum_i x_i^2 + |x_1 + 1| - n \leq 0$ has a smaller feasible set than the previous one, but has the same global minimizer. Furthermore, the constraint is nonsmooth at the global minimizer. Its feasible set is depicted in the right hand side diagram of

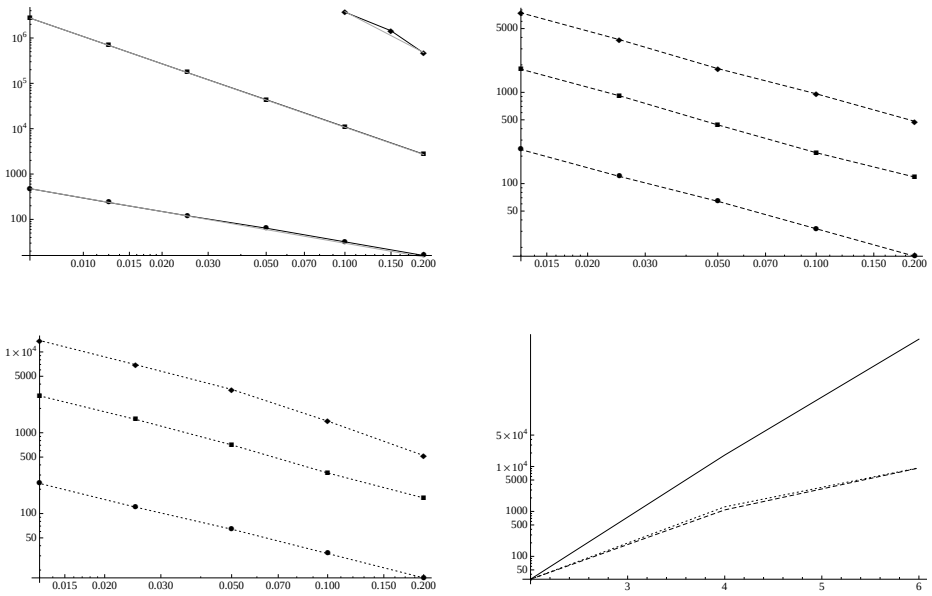

Fig. 5 Smooth bi-objective academic problem. Same graphics as in Figure 3 with $n \in \{2, 4, 6\}$ and $\epsilon = 0.1$ for the lower right graphic.

Figure 2 for $n = 2$. The nonsmooth edge at the global minimum removes the tangency between the feasible set and the objective level set, and therefore should prevent the cluster effect. This is confirmed experimentally: Figure 4 shows the same graphic as Figure 3 for this nonsmooth academic problem. For $n = 2$ with no rejection test (upper left graphic), we see that the number of ϵ-boxes remains constant, hence the absence of the cluster effect. For higher dimensions, the cluster effect reappears but following the model $O(\epsilon^{-\frac{n-2}{2}})$ (these models are shown in gray lines), i.e. the nonsmooth edge removes one dimension of the cluster effect. Still the cluster effect prevents any attempt of solving this academic problem using a branch and bound algorithm for even quite small values of n.

The upper right and lower left graphics of Figure 4 show that again the rejection tests allows drastically reducing the cluster effect, although the exponential dependence of the number of ϵ-boxes with respect to the dimension is confirmed in the lower right graphic of Figure 4.

5.3 Smooth Bi-Objective Academic Problem

We consider the bi-objective problem consisting in minimizing $f(x) = (a^T x, b^T x)^T$, with $a_i = 1$ and $b_i = (-1)^i$ for $i \in \{1, \ldots, n\}$, $x \in \mathbb{R}^n$ subject to $g(x) := x^T x - n \leq 0$. We allow only even n, since we want the vectors a and b to be orthogonal. We again transform the optimization problem into a constraint satisfaction problem by computing its global Pareto frontier and enforcing the dominance constraint. Since the problem is convex, the Pareto frontier can be obtained by minimizing aggregations of the objectives $f_t(x) = (1 - t)\, a^T x + t\, b^T x$ for $t \in [0, 1]$. One

easily computes the global minimizer x_t of this problem, and the corresponding bi-objective evaluation:

$$x_t = -\sqrt{n}\frac{(1-t)a+tb}{\|(1-t)a+tb\|} \quad \text{and} \quad f(x_t) = -n\begin{pmatrix} \frac{1-t}{\sqrt{(1-t)^2+t^2}} \\ \frac{t}{\sqrt{(1-t)^2+t^2}} \end{pmatrix} \tag{28}$$

As t varies inside $[0,1]$, $f(x_t)$ follows the quarter of the radius n circle in the negative quadrant. Therefore, global Pareto dominance can be enforced by the following constraints:

$$(a^T x)^2 + (b^T x)^2 \geq n^2 \ , \ a^T x \leq 0 \ , \ b^T x \leq 0 \ \text{ and } \ g(x) \leq 0. \tag{29}$$

The constraint $(a^T x)^2 + (b^T x)^2 \geq n^2$ can be reformulated as

$$(\sum_{i=1}^{\frac{n}{2}} x_{2i})^2 + (\sum_{i=1}^{\frac{n}{2}} x_{2i-1})^2 \geq \frac{n}{2}^2, \tag{30}$$

which removes the interval evaluation pessimism (since each variable has only one occurrence in (30)) and hence allows us to focus only on the cluster effect. Enforcing (30) is equivalent to enforcing the dominance relation with respect to every point of the Pareto frontier.

The asymptotic analysis conducted in Subsection 5.1 also holds here, although since the solution set is now a 1-dimensional manifold the the problem behaves as it would have one variable less. Furthermore, the contribution of the cluster effect is along the 1-dimensional manifold, leading to a cluster of $O(\epsilon^{-1}\,\epsilon^{-\frac{(n-1)-1}{2}}) = O(\epsilon^{-\frac{n}{2}})$ boxes. This behavior is confirmed by the upper left graphic of Figure 5. The upper right and lower left graphics in the same figure show that the first order rejection tests drastically reduce the cluster effect, however since we have to cover a 1-dimensional manifold of fixed length, we also observe that the number of boxes still increases by $O(\epsilon^{-1})$.

5.4 Benchmark Problem

We finally consider the following nonlinear problem taken from [7]:

$$\begin{array}{ll} \min & -18\log(x_2+1) - 19.2\log(x_1-x_2+1) + 5y_1 + 6y_2 + 8y_3 + 10x_1 - 7x_3 + 10 \\ \text{s.t.} & 0.8\log(x_2+1) + 0.96\log(x_1-x_2+1) - 0.8x_3 \geq 0 \\ & \log(x2+1) + 1.2\log(x_1-x_2+1) - x_3 - 2y_3 + 2 \geq 0 \\ & x_2 - x_1 \leq 0 \\ & x_2 - 2y_1 \leq 0 \\ & -x_2 + x_1 - 2y_2 \leq 0 \\ & y_1 + y_2 - 1 \leq 0 \end{array} \tag{31}$$

with bound constraints $x_1, x_2 \in [0,2]$ and $x_3, y_1, y_2, y_3 \in [0,1]$. Note that the variables y_i are integers in [7] but we solve here the version proposed in the Coconut [37] benchmarks where this integrality constraint is relaxed. Figure 6 shows that on this problem, both the full column rank rejection test and the multipliers domains rejection test allow removing the cluster effect. The left hand side graphic

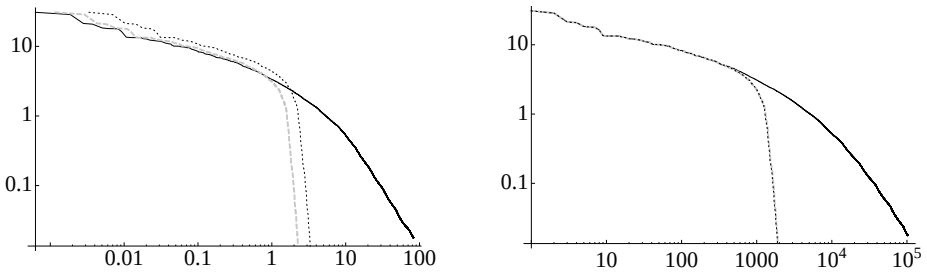

Fig. 6 Benchmark problem: Width of the objective enclosure with respect to computation time (left) and number of splits (right). For each figure, full line, dashed line and dotted line correspond respectively to no rejection test, full column rank rejection test and multipliers domain rejection test.

shows that the full column rank rejection test is approximately 1.5 quicker than the multipliers domain rejection test on this problem, while the right hand side graphic shows that the both achieve the same performance with respect to the number of splits.

6 Conclusion

Three rejection tests based on first order multi-objective optimality conditions and interval arithmetic have been proposed. They allow rejecting boxes not containing any local minimizer. They generalize previously proposed rejection tests and can be easily implemented and included in any bisection or branch and bound algorithm. In the context of mono-objective optimization, although less powerful than fully solving the first order system of equations, they are much simpler to implement and our experiments have shown that they allow drastically deceasing the cluster effect. On the other hand, the proposed rejection tests can be used to preprocess and then solve the first order optimality conditions. In the context of multi-objective optimization, the first order system of equations is under-constrained and hence cannot be solved as efficiently as for mono-objective problems. Therefore, these rejection tests are even more important in the context of multi-objective problems, which also suffer of the cluster effect as illustrated in Subsection 5.3.

Acknowledgements This work was partially funded by the *Région Pays de la Loire* of France, and the National Institute of Informatics of Japan. The authors are much grateful to Christophe Jermann for his valuable help in experimenting and analyzing the cluster effect of the presented academic examples.

References

1. G. Alefeld and J. Herzberger. *Introduction to Interval Computations*. Computer Science and Applied Mathematics, 1974.

2. Vincent Barichard, Herv Deleau, Jin-Kao Hao, and Frdric Saubion. A Hybrid Evolutionary Algorithm for CSP. In *Artificial Evolution*, volume 2936 of *LNCS*, pages 79–90. Springer Berlin / Heidelberg, 2004.
3. Vincent Barichard and Jin-Kao Hao. A Population and Interval Constraint Propagation Algorithm. In *Evolutionary Multi-Criterion Optimization*, volume 2632 of *LNCS*, pages 72–72. Springer Berlin / Heidelberg, 2003.
4. Frank H. Clarke. *Optimization and Nonsmooth Analysis*. Society for Industrial and Applied Mathematics, Philadephia, PA, 1990.
5. Ferenc Domes. Gloptlab: a configurable framework for the rigorous global solution of quadratic constraint satisfaction problems. *Optimization Methods Software*, 24(4-5):727–747, 2009.
6. Kaisheng Du and R. Baker Kearfott. The cluster problem in multivariate global optimization. *Journal of Global Optimization*, 5:253–265, 1994.
7. Marco Duran and Ignacio Grossmann. An outer-approximation algorithm for a class of mixed-integer nonlinear programs. *Mathematical Programming*, 36:307–339, 1986.
8. J. Garloff. Interval Gaussian Elimination with Pivot Tightening. *SIAM Journal on Matrix Analysis and Applications*, 30(4):1761–1772, 2009.
9. D. Goldberg. What every computer scientist should know about floating-point arithmetic. *Computing Surveys*, 23(1):5–48, 1991.
10. F. Goualard. *GAOL 3.1.1: Not Just Another Interval Arithmetic Library*. Laboratoire d'Informatique de Nantes-Atlantique, 4.0 edition, October 2006.
11. Laurent Granvilliers and Alexandre Goldsztejn. A Branch-and-Bound Algorithm for Unconstrained Global Optimization. In *Proceeding of the 14th GAMM-IMACS International Symposium on Scientific Computing, Computer Arithmetic and Validated Numerics (SCAN)*, 2010.
12. E. Hansen. *Global Optimization Using Interval Analysis, Second Edition*. Marcel Dekker, NY, 1992.
13. E.R. Hansen and G.W. Walster. Bounds for Lagrange multipliers and optimal points. *Computers & Mathematics with Applications*, 25(1011):59 – 69, 1993.
14. K. Ichida and Y. Fujii. Multicriterion optimization using interval analysis. *Computing*, 44:47–57, 1990.
15. Johannes Jahn. Multiobjective Search Algorithm with Subdivision Technique. *Comput. Optim. Appl.*, 35(2):161–175, 2006.
16. L. Jaulin, M. Kieffer, O. Didrit, and E. Walter. *Applied Interval Analysis with Examples in Parameter and State Estimation, Robust Control and Robotics*. Springer-Verlag, 2001.
17. R. B. Kearfott. Interval Computations: Introduction, Uses, and Resources. *Euromath*, Bulletin 2(1):95–112, 1996.
18. R. Baker Kearfott. An interval branch and bound algorithm for bound constrained optimization problems. *Journal of Global Optimization*, 2:259–280, 1992.
19. Ralph Baker Kearfott. Interval computations, rigour and non-rigour in deterministic continuous global optimization. *Optimization Methods Software*, 26(2):259–279, 2011.
20. O. Knueppel. PROFIL/BIAS - A Fast Interval Library. *Computing*, 53(3-4):277–287, 1994.
21. Bartlomiej Kubica and Ewa Niewiadomska-Szynkiewicz. An Improved Interval Global Optimization Method and Its Application to Price Management Problem. In *Applied Parallel Computing. State of the Art in Scientific Computing*, volume 4699 of *LNCS*, pages 1055–1064. Springer Berlin / Heidelberg, 2007.
22. Bartlomiej Kubica and Adam Wozniak. Interval Methods for Computing the Pareto-front of a Multicriterial Problem. In *Parallel Processing and Applied Mathematics*, volume 4967 of *LNCS*, pages 1382–1391. Springer Berlin / Heidelberg, 2008.
23. Bartlomiej Kubica and Adam Wozniak. Using the Second-Order Information in Pareto-set Computations of a Multi-criteria Problem. In *Applied Parallel and Scientific Computing*, volume 7134 of *LNCS*, pages 137–147. Springer Berlin / Heidelberg, 2012.
24. Youdong Lin and Mark A. Stadtherr. LP Strategy for Interval-Newton Method in Deterministic Global Optimization. In Christodoulos A. Floudas and Panos M. Pardalos, editors, *Encyclopedia of Optimization*, pages 1937–1943. Springer US, 2009.
25. Kyoko Makino and Martin Berz. New applications of taylor model methods. In George Corliss, Christèle Faure, Andreas Griewank, Lauren Hascoët, and Uwe Naumann, editors, *Automatic differentiation of algorithms*, pages 359–364. Springer-Verlag New York, Inc., New York, NY, USA, 2002.
26. K.M. Miettinen. *Nonlinear Multiobjective Optimization*. Kluwer, 1998.

27. R. Moore. *Interval Analysis*. Prentice-Hall, 1966.
28. A. Neumaier. *Interval Methods for Systems of Equations*. Cambridge Univ. Press, 1990.
29. Jorge Nocedal and Stephen Wright. *Numerical Optimization*. Springer, 2006.
30. Jiri Rohn. Forty Necessary and Sufficient Conditions for Regularity of Interval Matrices: A Survey. *Electronic Journal of Linear Algebra*, 18:500–512, 2009.
31. G.R. Ruetsch. An interval algorithm for multi-objective optimization. *Structural and Multidisciplinary Optimization*, 30:27–37, 2005.
32. G.R. Ruetsch. Using interval techniques of direct comparison and differential formulation to solve a multi-objective optimization problem (patent US–7742902), 2010.
33. S.M. Rump. INTLAB - INTerval LABoratory. In Tibor Csendes, editor, *Developments in Reliable Computing*, pages 77–104. Kluwer Academic Publishers, Dordrecht, 1999.
34. Hermann Schichl and Arnold Neumaier. Exclusion Regions for Systems of Equations. *SIAM Journal on Numerical Analysis*, 42(1):383–408, 2004.
35. Hermann Schichl and Arnold Neumaier. Interval analysis on directed acyclic graphs for global optimization. *J. of Global Optimization*, 33(4):541–562, 2005.
36. Hermann Schichl and Arnold Neumaier. Transposition theorems and qualification-free optimality conditions. *SIAM Journal on Optimization*, 17(4):1035–1055, 2006.
37. Oleg Shcherbina, Arnold Neumaier, Djamila Sam-Haroud, Xuan-Ha Vu, and Tuan-Viet Nguyen. Benchmarking global optimization and constraint satisfaction codes. In Christian Bliek, Christophe Jermann, and Arnold Neumaier, editors, *Global Optimization and Constraint Satisfaction*, volume 2861 of *Lecture Notes in Computer Science*, pages 211–222. Springer Berlin / Heidelberg, 2003.
38. G.L. Soares, R.O. Parreiras, L. Jaulin, J.A. Vasconcelos, and C.A. Maia. Interval robust multi-objective algorithm. *Nonlinear Analysis: Theory, Methods & Applications*, 71(12):1818–1825, 2009.
39. B.G. Tóth and J.F. Hernndez. *Interval Methods for Single and Bi-objective Optimization Problems Applied to Competitive Facility Location Models*. LAP Lambert Academic Publishing, 2010.
40. Xuan-Ha Vu, Hermann Schichl, and Djamila Sam-Haroud. Interval propagation and search on directed acyclic graphs for numerical constraint solving. *J. of Global Optimization*, 45(4):499–531, 2009.
41. *Wolfram Research inc., Mathematica*. Champaign, Illinois, Version 7.0, 2008.