Mathematical Programming manuscript No.
(will be inserted by the editor)

Constraint aggregation for rigorous global optimization

Ferenc Domes · Arnold Neumaier

Received: date / Accepted: date

Abstract In rigorous constrained global optimization, upper bounds on the objective function help to reduce the search space. Their construction requires finding a narrow box around an approximately feasible solution, verified to contain a feasible point. Approximations are easily found by local optimization, but the verification often fails.

In this paper we show that even if the verification of an approximate feasible point fails, the information extracted from the local optimization can still be used in many cases to reduce the search space. This is done by a rigorous filtering technique called constraint aggregation. It forms an aggregated redundant constraint, based on approximate Lagrange multipliers or on a vector valued measure of constraint violation. Using the optimality conditions, two sided linear relaxations, the Gauss-Jordan algorithm and a directed modified Cholesky factorization, the information in the redundant constraint is turned into powerful bounds on the feasible set. Constraint aggregation is especially useful since it also works in a tiny neighborhood of the global optimizer, thereby reducing the cluster effect.

A simple introductory example demonstrates how our new method works. Extensive tests show the performance on a large benchmark.

Keywords constraint aggregation · global optimization · constraint satisfaction · filtering method · verified computing · interval analysis

Ferenc Domes
University of Vienna, Faculty of Mathematics,
Oskar-Morgenstern-Platz 1, A-1090 Vienna,
Tel.: +431 4277 50665, E-mail: Ferenc.Domes@univie.ac.at

Arnold Neumaier
University of Vienna, Faculty of Mathematics,
Oskar-Morgenstern-Platz 1, A-1090 Vienna,
Tel.: +431 4277 50661, E-mail: Arnold.Neumaier@univie.ac.at

Contents

1 Introduction

Global optimization is the task of finding the absolutely best admissible conditions to achieve an objective under given constraints, assuming that both are formulated in mathematical terms. Global optima can be found by a combination of a variety of filtering techniques usually embedded in a branch and bound scheme for complete search (see, e.g., the survey Neumaier [26]). If the results have to be rigorous, the calculations usually involve the use of interval arithmetic (see, e.g., Neumaier [24]).

Filtering (also called pruning) stands for reducing or discarding parts of the search space of an optimization problem. The classical filtering algorithms are based upon local consistencies like 2B-consistency or Box-consistency (see, e.g., Benhamou et al. [1]), 3B-consistency (Lhomme [22]), HC4 (Benhamou et al. [2]), FBPD (Vu et al. [34]), OCTUM (Chabert and Jaulin [3]). For quadratic problems, improved filtering methods are discussed in Domes and Neumaier [5].

Higher order filtering methods usually include linear or convex relaxation. Rigorous linear over- and underestimators for general global nonlinear programming problems involving odd and even powers, reciprocals, exponentials, logarithms, square roots, and uncertain scalar multiples are discussed in Hongthong and Kearfott [14]. The relaxed linear program usually contains more variables and/or constraints than the original problem, but the constraints are much easier to exploit. A classical method by McCormick [23], extended by Sherali and Adams [33], called RLT (reformulation – linearization technique), is used by Lebbah et al. [21] in the QUAD algorithm and by the prize-winning (but nonrigorous) global optimization code BARON [27, 28]. Another interesting approach was given by Kolev [20], and a selection of additional linear relaxation techniques can be found in Domes and Neumaier [6]. Higher degree relaxations and convex relaxations are also discussed in the literature; for example, affine and convex relaxations for non-convex multivariate polynomials in Garloff et al. [11].

Usually, filtering methods are very efficient at the beginning of a branch and bound procedure, but they tend to become inefficient close to the global solution, resulting in excessive branching until the required precision is achieved. This so-called cluster effect was first explained by Kaisheng and Kearfott [15]. Techniques designed to reduce or eliminate the cluster effect are discussed, e.g., by Schichl and Neumaier [30] and Goldsztejn et al. [12].

We introduce a new rigorous filtering technique called constraint aggregation. Based on an approximately most feasible point, an aggregated redundant constraint is formed, using approximate Lagrange multipliers when the approximation is nearly feasible, or a vector valued measure of constraint violation when the approximation is sufficiently infeasible. Using the optimality conditions, two sided linear relaxations, the Gauss-Jordan algorithm and a directed modified Cholesky factorization, the information in the redundant constraint is turned into powerful bounds on the feasible set. Constraint aggregation is especially useful since it also works in a tiny neighborhood of the global optimizer, thereby reducing the cluster effect.

The following motivating example shows that constraint aggregation may drastically improve the enclosure of a feasible set. The theory developed in the present paper then casts the tricks behind this example – in fact variations of techniques used by Neumaier [25] to prove sufficient global optimality conditions for quadratic programs – into a general and powerful technique.

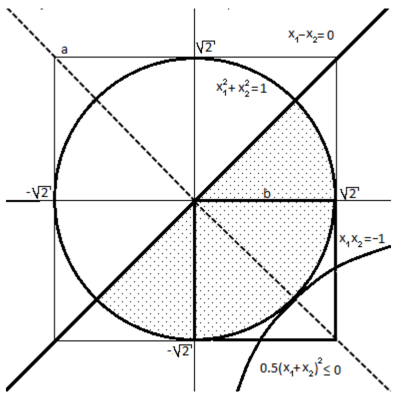

Fig. 1 Motivating Example.

Example 1 Consider the simple two dimensional optimization problem

$$\begin{array}{ll} \min & x_1 x_2 \\ \text{s.t.} & x_1^2 + x_2^2 \le 2, \\ & x_1 - x_2 \ge 0. \end{array} \tag{1}$$

Suppose that a local optimizer found the optimal solution $x_1 = 1$, $x_2 = -1$ of (1), and the associated multipliers $\nu = 1$ (for the objective) and $y = (-0.5, 0)$ (for the constraints). The upper bound of the objective arising from the known point (namely the optimal solution) is $x_1x_2 \le -1$, resulting in the CSP

$$\begin{array}{ll} \text{find} & x \\ \text{s.t.} & x_1x_2 \le -1, \\ & x_1^2 + x_2^2 \le 2, \\ & x_1 - x_2 \ge 0 \end{array} \tag{2}$$

for potentially better points (see Figure 1). In the present case (as always when the local search happened to find the unique global minimizer), the CSP (2) has a feasible set consisting of a single point only – the minimizer.

Constraint propagation on the constraints of (2) only results in $x_1, x_2 \in [-\sqrt{2}, \sqrt{2}]$ (Figure 1, box `a`) but if we aggregate the constraints using the multipliers $\nu = 1$ and $y = (-0.5, 0)$ we obtain

$$x_1x_2 - (-0.5)(x_1^2 + x_2^2) + 0(x_1 - x_2) \le 1(-1) - (-0.5)2 + 0,$$

hence $0.5(x_1+x_2)^2 \le 0$. The introduction of the new variable z for x_1+x_2 transforms this into the constraints

$$z = x_1 + x_2, \quad 0.5z^2 \le 0. \tag{3}$$

Constraint propagation on (3) gives $z \in [0, 0]$. Therefore, the aggregated CSP may be written as

$$x_1x_2 \le -1, \quad x_1^2 + x_2^2 \le 2, \quad x_1 - x_2 \ge 0, \quad z = x_1 + x_2, \quad 0.5z^2 \le 0, \tag{4}$$

with additional bounds $x_1, x_2 \in [-\sqrt{2}, \sqrt{2}]$. The interval hull of the linear subproblem

$$x_1 - x_2 \ge 0, \quad z = x_1 + x_2, \quad x_1, x_2 \in [\,-\sqrt{2}, \sqrt{2}], \quad z \in [0, 0], \tag{5}$$

obtainable constructively through linear bounding for the variables x_1 and x_2 (see Domes and Neumaier [6, p. 17]) is $x_1 \in [0, \sqrt{2}]$ and $x_2 \in [-\sqrt{2}, 0]$ (Figure 1, box `b`). With these improved bounds, constraint propagation on $x_1x_2 \le -1$ and $z = x_1 + x_2$ contracts the bounds to a single point, the optimal solution; cf. Figure 1.

In floating point arithmetic, the same computations should have approximately the same results.

The paper is organized as follows. After providing basic notation and terminology in Section 2.1, 2.2 and 2.3 we discuss a method for computing Lagrange-multipliers in 2.4. Then we introduce uncertainties (Section 2.5) and use them in Section 2.6 to specify of the problem class treated, namely the uncertain optimization problems. Then we conlcude the preliminaries by discussing feasibility (Section 2.7), bounds on the objective and verification of feasible points in Section 2.8. The second part is concerned about the new method, in particular about filtering by constraint aggregation (Section 3.1), filtering a singly-quadratic constraint satisfaction problem (Section 3.2), and finding good aggregators (Section 4). The latter is related to computable certificates of infeasibility (Subsection 4.1). We conclude the paper by giving extensive numerical tests in Section 5.

Acknowledgments This research was supported by the Austrian Science Fund (FWF) under the contract numbers P23554-N13 and P22239-N13.

2 Preliminaries

2.1 Matrix notation

$\mathbb{R}^{m\times n}$ denotes the vector space of all $m \times n$ matrices A with real entries A_{ik} ($i = 1,\dots,m,\ k = 1,\dots,n$), and $\mathbb{R}^n = \mathbb{R}^{n\times 1}$ denotes the vector space of all column vectors of length n. For vectors and matrices, the relations $=$, $\neq$, $<$, $>$, $\leq$, $\geq$ and the absolute value $|A|$ of a matrix A are interpreted component-wise.

The number of nonzero entries of a matrix A is denoted by $\mathrm{nnz}(A)$. The n-dimensional identity matrix is denoted by I and the n-dimensional zero matrix is denoted by $\mathit{0}$. The transpose of a matrix A is denoted by A^T, and A^{-T} is short for $(A^T)^{-1}$. The ith row vector of a matrix A is denoted by $A_{i:}$ and the jth column vector by $A_{:j}$. For an $n\times n$ matrix A, $\mathrm{diag}(A)$ denotes the n-dimensional vector with $\mathrm{diag}(A)_i = A_{ii}$.

The number of elements of an index set N is denoted by $|N|$. The set $\neg N$ denotes the complement of N. Let $I \subseteq \{1,\dots,m\}$ and $J \subseteq \{1,\dots,n\}$ be index sets and let $n_I := |I|$, $n_J := |J|$. For an n-dimensional vector x, x_J denotes the n_J-dimensional vector built from the components of x selected by the index set J. For an $m \times n$ matrix A, the expression $A_{I:}$ denotes the $n_I \times n$ matrix built from the rows of A selected by the index sets I. Similarly, $A_{:J}$ denotes the $m \times n_J$ matrix built from the columns of A selected by the index sets J. Instead of using the index sets I and J we also write $A_{i:k,j:l}$ if $I = \{i, i+1,\dots,k\}$ and $J = \{j, j+1,\dots,l\}$.

2.2 Boxes

A **box** $\mathbf{x} = [\underline{x}, \overline{x}]$, i.e., the Cartesian product of the closed real intervals $\mathbf{x}_i := [\underline{x}_i, \overline{x}_i]$, represents a (bounded or unbounded) axiparallel box in $\mathbb{R}^n$. $\overline{\mathbb{IR}}^n$ denotes the set of all n-dimensional boxes. To take care of one-sided bounds on variables, the values $-\infty$ and ∞ are allowed as lower and upper bounds of a box, respectively. The condition $x \in \mathbf{x}$ is equivalent to the collection of simple bounds

$$\underline{x}_i \leq x_i \leq \overline{x}_i \quad (i = 1,\dots,n),$$

or, with inequalities on vectors and matrices interpreted component-wise, to the two-sided vector inequality $\underline{x} \leq x \leq \overline{x}$. Apart from two-sided constraints, this includes with $\mathbf{x}_i = [a, a]$ variables x_i fixed at a particular value $x_i = a$, with $\mathbf{x}_i = [a,\infty]$ lower bounds $x_i \geq a$, with $\mathbf{x}_i = [-\infty, a]$ upper bounds $x_i \leq a$, and with $\mathbf{x}_i = [-\infty,\infty]$ free variables.

For the notation in interval analysis we mostly follow [19]. The box

$$\square S := [\,\inf(S), \sup(S)]$$

is called the **interval hull** of a set S of points in $\mathbb{R}^n$. We also define the **minimal point**

$$\mu(\mathbf{r}) := \begin{cases} \underline{r} \text{ if } \underline{r} > 0, \\ \overline{r} \text{ if } \overline{r} < 0, \\ 0 \text{ otherwise}, \end{cases} \tag{6}$$

the **mignitude**

$$\langle \mathbf{r} \rangle := |\mu(\mathbf{r})|,$$

and the **magnitude**

$$|\mathbf{r}| := \max(-\underline{r}, \overline{r})$$

of an interval $\mathbf{r}$. The notation extends componentwise to boxes.

2.3 Optimization problems

With the notation introduced, the traditional continuous, single-objective **optimization problem** consisting of smooth equality and inequality constraints may be written in the compact interval form

$$\begin{array}{ll} \min & f(x) \\ \text{s.t.} & F(x) \in \mathbf{F}, \quad x \in \mathbf{x}, \end{array} \tag{7}$$

where $f : \mathbf{x} \to \mathbb{R}$ and $F : \mathbf{x} \to \mathbb{R}^m$ are functions defined on the box $\mathbf{x}$, and $\mathbf{F} \in \overline{\mathbb{IR}}^m$ is a box defining two-sided constraints for the components of $F(x)$; again, equality constraints and one-sided inequality constraints are included. A point $x \in \mathbf{x}$ is called a **feasible point** of (7) if $F(x) \in \mathbf{F}$ is satisfied. If $F(x) \notin \mathbf{F}$ for all $x \in \mathbf{x}$ the constraints are called **inconsistent** and the problem is called **infeasible**.

For reasons of efficiency, we shall consider in place of (7) the slightly more complex formulation

$$\begin{array}{ll} \min & a^T F(x) \\ \text{s.t.} & BF(x) \in \mathbf{b},\ x \in \mathbf{x}, \end{array} \tag{8}$$

$F : \mathbf{x} \to \mathbb{R}^w$, and $a \in \mathbb{R}^w$, $\mathbf{b} \in \mathbb{R}^m$, $B \in \mathbb{R}^{m\times w}$. This is both a special case of (7) and a generalization of it, as the traditional formulation (7) is obtained from (8) if we take $w = m+1$, $\binom{f}{F}$ in place of F, $u = 1$, $a = \binom{1}{0}$, $B = (0\ I)$ and $\mathbf{b} = \mathbf{F}$. From the point of view of solvability, (7) and (8) are therefore equivalent, as one can redefine $\widetilde{f}(x) := a^T F(x)$ and $\widetilde{F}(x) := BF(x)$. However, from a computational point of view, the form (8) has advantages that typically lead to improved linear relaxations once $\binom{a}{B}$ has more than one entry in some column. As we shall see in Section 2.6, this form also allows a natural formulation of problems with uncertain coefficients.

If the objective function is missing or it is constant then (7) and (8) take the form

$$\begin{array}{ll} \text{find} & x \in \mathbf{x} \\ \text{s.t.} & F(x) \in \mathbf{F}, \end{array} \tag{9}$$

$$\begin{array}{ll} \text{find} & x \in \mathbf{x} \\ \text{s.t.} & BF(x) \in \mathbf{b}, \end{array} \tag{10}$$

respectively, of a **constraint satisfaction problem** (**CSP**).

2.4 Lagrange multipliers

We now consider the first order optimality conditions for a minimizer $\widehat{x}$ of an optimization problem of the form (8), where $f(x)$ and $F(x)$ are continuously differentiable. We put

$$\begin{array}{ll} L^b & := \{i \mid \underline{x}_i = \widehat{x}_i < \overline{x}_i\}, \\ U^b & := \{i \mid \underline{x}_i < \widehat{x}_i = \overline{x}_i\}, \\ N^b & := \{i \mid \underline{x}_i < \widehat{x}_i < \overline{x}_i\}, \end{array} \tag{11}$$

$$
\begin{array}{l}
E^c := \{j \mid \underline{b}_j = \overline{b}_i\}, \\
L^c := \{j \mid \underline{b}_j = BF_j(\widehat{x}) < \overline{b}_j\}, \\
U^c := \{j \mid \underline{b}_j < BF_j(\widehat{x}) = \overline{b}_j\},
\end{array} \tag{12}
$$

and write $\mathbf{y}$ and $\mathbf{z}$ for the interval vectors with components

$$
\mathbf{z}_i := \begin{cases} [0,\infty] & \text{if } i \in L^b, \\ [-\infty,0] & \text{if } i \in U^b, \\ 0 & \text{if } i \in N^b, \end{cases} \qquad \mathbf{y}_j := \begin{cases} [-\infty,\infty] & \text{if } j \in E^c, \\ [0,\infty] & \text{if } j \in L^c, \\ [-\infty,0] & \text{if } j \in U^c, \\ 0 & \text{otherwise.} \end{cases} \tag{13}
$$

The necessary optimality conditions say that there are multipliers $\nu \in \mathbb{R}$ and $y \in \mathbf{y}$, not both zero, such that

$$
Z(\nu, x, y) := F'(x)^T(\nu a - B^T y) \in \mathbf{z}, \tag{14}
$$

and the complementarity conditions

$$
\max((BF_j(x) - \underline{b}_j)y_j, (BF_j(x) - \overline{b}_j)y_j) = 0 \tag{15}
$$

hold for $j = 1, \dots, m$.

These conditions comprise the **Karush-John optimality conditions** for the problem (8); cf. the derivation and discussion of the history in Schichl and Neumaier [31].

If $\nu \neq 0$ we may rescale the multipliers to have $\nu = 1$, leading to the **Kuhn-Tucker optimality conditions**. We normalize instead by rescaling so that

$$
\max(\nu, \|y\|_\infty) = 1,
$$

which is possible even when $\nu = 0$ and leads to bounded multipliers. This is achieved by

$$
\nu \leftarrow \frac{1}{\max(1, \|y\|_\infty)}, \qquad y \leftarrow \nu y. \tag{16}
$$

The following result suggests a way to define Lagrange multipliers $y \in \mathbb{R}^m$ (for the constraints) and $\nu \in \mathbb{R}$ (for the objective) at an arbitrary point x (intended to be an approximate local minimizer).

Theorem 1 *If, for some $\widehat{x} \in \mathbb{R}^n$, the constrained optimization problem*

$$
\begin{array}{l}
\min\ g(y) := \|\mu(Z(1,\widehat{x},y) - \mathbf{z})\|_2^2 \\
\text{s.t.}\ \ y \in \mathbf{y}
\end{array} \tag{17}
$$

(with Z from (14) *and μ from* (6)*) has a solution $\widehat{y}$ with $g(\widehat{y}) = 0$ then $(\widehat{x}, \widehat{y})$ satisfies the Kuhn-Tucker conditions for* (8)*, and* (16) *defines associated normalized multipliers satisfying the Karush-John conditions.*

Proof From $g(\widehat{y}) = 0$ follows that $\mu(Z(1,\widehat{x},y) - \mathbf{z}) = 0$ implying $Z(1,\widehat{x},y) \in \mathbf{z}$ therefore (14) is satisfied. Since $\widehat{y} \in \mathbf{y}$, the definition (13) of $\mathbf{y}$ implies that the complementary conditions (15)

$$
\max(c_1, c_2) = 0, \quad c_1 := \ (BF_j(\widehat{x}) - \underline{b}_j)\widehat{y}_j, \quad c_2 := (BF_j(\widehat{x}) - \overline{b}_j)\widehat{y}_j),
$$

are satisfied:
If $BF_j(\widehat{x}) = \underline{b}_j < \overline{b}_j$ then $\widehat{y}_j \geq 0$ therefore $c_1 = 0$ and $c_2 \leq 0$.
If $BF_j(\widehat{x}) = \overline{b}_j > \underline{b}_j$ then $\widehat{y}_j \leq 0$ therefore $c_1 \leq 0$ and $c_2 = 0$.
If $BF_j(\widehat{x}) = \underline{b}_j = \overline{b}_j$ then $\widehat{y}_j$ is arbitrary and $c_1 = 0$, $c_2 = 0$.
If $\underline{b}_j < BF_j(\widehat{x}) < \overline{b}_j$ then $\widehat{y}_j = 0$ therefore $c_1 = 0$ and $c_2 = 0$.
In each case, $\max(c_1, c_2) = 0$. Therefore the Kuhn-Tucker conditions are satisfied and $\widehat{x}$ must be a critical point of (8). □

Note that if $\widehat{x}$ is a Kuhn–Tucker point then the (possibly underdetermined) system of equations

$$F'(\widehat{x})^T_{N^b:} B^T_{:V} y_V = F'_{N^b}(\widehat{x})^T a, \quad y_{\neg V} = 0, \quad V = E^c \cup L^c \cup U^c, \tag{18}$$

must have a solution $\widehat{y}$. If, in addition to this, $\widehat{y} \in \mathbf{y}$, and the inequalities

$$Z(1, \widehat{x}, \widehat{y})_i \geq 0 \text{ for } i \in L^b, \quad Z(1, \widehat{x}, \widehat{y})_i \leq 0 \text{ for } i \in U^b, \tag{19}$$

are satisfied then $\widehat{y}$ is a Lagrange multiplier corresponding to $\hat{x}$, and (16) defines the associated normalized multipliers. If it works, this method gives a cheaper alternative for computing $\widehat{y}$; otherwise the more expensive constrained non-linear optimization problem (17) must be solved.

Algorithm 1: Computing the Lagrange multipliers

Input: A point $\widehat{x} \in \mathbb{R}^n$ approximately satisfying the bound constraint $x \in \mathbf{x}$ of (8) and a small tolerance $\delta \ll 1$ (e.g., $\delta := 10^{-9}$).
Output: The Lagrange multipliers: $\widehat{\nu}$ for the objective and $\widehat{y}$ for the constraints.

Compute $\delta^b_i = \min(\delta, \mathrm{wid}(\mathbf{x}_i)/10)$ for $i = 1, \dots, m$;
if $\widehat{x}_i < \min(\underline{x}_i + \delta^b_i, \overline{x}_i)$ **then** $\widehat{x}_i \leftarrow \underline{x}_i$;
if $\max(\underline{x}_i, \overline{x}_i - \delta^b_i) < \widehat{x}_i$ **then** $\widehat{x}_i \leftarrow \overline{x}_i$;
Form the index sets L^b, U^b and N^b as defined in (11);
Compute $\delta^c_j = \min(\delta, \mathrm{wid}(\mathbf{b}_j)/10)$ for $j = 1, \dots, m$;
Form the index sets from (12) by

$$\begin{aligned} &E^c := \{j \mid \overline{b}_j - \underline{b}_j \leq \delta\}, \quad L^c := \{j \notin E^c \mid BF_j(\widehat{x}) \leq \min(\underline{b}_j + \delta^c_j, \overline{b}_j)\}, \\ &U^c := \{j \notin E^c \mid BF_j(\widehat{x}) \geq \max(\underline{b}_j, \overline{b}_j - \delta^c_j)\}. \end{aligned} \tag{20}$$

Construct the boxes $\mathbf{z}$ and $\mathbf{y}$ as given by (13);
Solve the linear system of equations (18) in order to obtain $\widehat{y}$;
if $\widehat{y} \notin \mathbf{y}$ *or one of the conditions* (19) *is not satisfied* **then**
    Solve the problem (17) by using a bound constrained solver;
    if *the solver found the solution* $\widetilde{y}$ *with* $g(\widetilde{y}) = 0$ **then** set $\widehat{y} \leftarrow \widetilde{y}$;
    else $\widehat{x}$ cannot satisfy the Kuhn-Tucker conditions, therefore signal failure;
end
Compute and return $\widehat{\nu}$ and $\widehat{y}$ according to (16);

In floating point arithmetic, the equalities and inequalities from above are often not satisfied exactly but only by a small tolerance δ. Algorithm 1 describes the solution process suitable for numerical computations.

2.5 Uncertain vectors and matrices

To rigorously account for inaccuracies in computed entries of a matrix, we use interval matrices, standing for uncertain real matrices whose coefficients are between given lower and upper bounds. Note that all boxes may be considered as interval vectors, i.,e., column vectors ($n \times 1$ matrices) with uncertain components, whose values are known only to lie withing given intervals. The midpoint, width and the radius of an interval matrix $\mathbf{A}$ are the noninterval matrices defined by

$$\mathrm{mid}(\mathbf{A}) := (\overline{A} + \underline{A})/2, \quad \mathrm{wid}(\mathbf{A}) := \overline{A} - \underline{A}, \quad \mathrm{rad}(\mathbf{A}) := \mathrm{wid}(\mathbf{A})/2,$$

respectively. An interval, interval vector, or interval matrix is called **thin** or **degenerate** if its width is zero, and **thick** if its width is positive. A real matrix A is identified with the thin interval matrix with $\underline{A} = \overline{A} = A$.

The expression $\mathbf{A} := [\underline{A}, \overline{A}] \in \overline{\mathbb{IR}}^{m\times n}$ denotes an $m \times n$ interval matrix with lower bound $\underline{A}$ and upper bound $\overline{A}$. $\mathbf{A} \in \overline{\mathbb{IR}}^{n\times n}$ is symmetric if $\mathbf{A}_{ik} = \mathbf{A}_{ki}$ for all $i, k \in \{1, \ldots, n\}$. The comparison matrix $\langle \mathbf{A} \rangle$ of a square interval matrix $\mathbf{A}$ is defined by

$$\langle \mathbf{A} \rangle_{ij} := \begin{cases} -|\,\mathbf{A}_{ij}| & \text{for } i \neq j, \\ \langle \mathbf{A}_{ij} \rangle & \text{for } i = j. \end{cases}$$

Given an expression $p(x)$ in $x = (x_1, \ldots, x_n)^T$ such that the evaluation at any $x \in \mathbf{x}$ is a real number, there are a number of methods for defining an **interval enclosure** of $p(x)$, i.e., a box $p(\mathbf{x})$ such that $p(x) \in p(\mathbf{x})$ holds for all $x \in \mathbf{x}$. The simplest is the interval evaluation, where one substitutes $\mathbf{x}_i$ for each occurrence of x_i in $p(x)$. More sophisticated (and often, but not always, better) possibilities include centered forms (for details, see, e.g., [24]).

2.6 Uncertain optimization problems

Traditionally, the coefficients of $f(x)$ and $F(x)$ are taken to be exactly known. To be able to rigorously account for uncertainties due to one of the following sources:

- measurements of limited accuracy,
- conversion errors from an original representation to our normal form,
- rounding errors when creating new constraints by relaxation techniques,

we allow the coefficients to vary in narrow intervals. All uncertainties can be conveniently expressed if we formulate an arbitrary optimization problem with uncertain coefficients as an instance of the following **uncertain optimization problem** (**UOP**)

$$\begin{array}{ll} \min & a^T F(x) \\ \text{s.t.} & BF(x) \in \mathbf{b},\ x \in \mathbf{x}, \\ \text{for some } a \in \mathbf{a},\ B \in \mathbf{B}. & \end{array} \tag{21}$$

Here $F : \mathbf{x} \to \mathbb{R}^w$ is defined on the box $\mathbf{x} \in \overline{\mathbb{IR}}^n$, and $a \in \mathbf{a} \in \mathbb{IR}^u$, $\mathbf{b} \in \overline{\mathbb{IR}}^m$, $B \in \mathbf{B} \in \mathbb{IR}^{m\times w}$. The entries of a and B are *not* variables but uncertain constants, whose precise values within the bounds $a \in \mathbf{a}$ and $B \in \mathbf{B}$ are not known. Thus whether a particular vector x is a solution of the UOP may depend on which $a \in \mathbf{a}$ and $B \in \mathbf{B}$ is the true value. This ambiguity makes working with uncertain constraints nontrivial. It requires great care in the derivation of methods to ensure the validity of an enclosure no matter which value $a \in \mathbf{a}$ and $B \in \mathbf{B}$ is the true value.

If $\mathbf{a}$ and $\mathbf{B}$ contain only a single matrix, (21) reduces to the **exact optimization problem** ((**UOP**) (8).

If $\mathbf{a} = 0$ then (21) becomes the **uncertain constraint satisfaction problem** (**UCSP**)

$$\begin{array}{ll} \text{find} & x \in \mathbf{x} \\ \text{s.t.} & BF(x) \in \mathbf{b} \\ \multicolumn{2}{l}{\text{for some } a \in \mathbf{a},\ B \in \mathbf{B}.} \end{array} \tag{22}$$

If, in addition, $\mathbf{a}$ and $\mathbf{B}$ contain only a single matrix, (21) reduces to the **exact constraint satisfaction problem** (ECSP) (10).

Any optimization problem with uncertain coefficients can be brought into the UOP form (21) by introducing new variables for every subexpression composed of a product with an uncertain coefficient or a linear combination in which a coefficient is uncertain. The transformation to this form is done automatically in the upcoming version of GloptLab (Domes [4]).

As an example we consider the nonlinear, exact optimization problem

$$\begin{array}{l} \min\ x_1 + x_2 \\ \text{s.t.}\ \ x_1 + e^{0.1x_1+0.2x_2^2} \le 1,\ x_1 \in [-1,1],\ x_2 \in [-2,0]. \end{array} \tag{23}$$

Since the decimal numbers occuring in the problem are not exactly representable as floating-point numbers, (23) must be represented internally as an UOP by introducing the intermediate variable $x_3 = 0.1x_1 + 0.2x_2^2$. Thus we have

$$\begin{array}{l} \min\ x_1 + x_2 \\ \text{s.t.}\ \ 0.1x_1 - x_3 + 0.2x_2^2 = 0, \quad x_1 + e^{x_3} \le 1, \\ \qquad x_1 \in [-1,1], \quad x_2 \in [-2,0], \quad x_3 \in [-\infty,\infty], \end{array}$$

ending up in

$$\begin{array}{l} \min\ a^T F(x) \\ \text{s.t.}\ \ BF(x) \in \mathbf{b}, \quad x \in \mathbf{x}, \end{array} \tag{24}$$

where

$$F(x) := (x_1,\ x_2,\ x_3,\ x_2^2,\ e^{x_3})^T,$$

$$a^T := (1\ 1\ 0\ 0\ 0), \quad B := \begin{pmatrix} 0.1 & 0 & -1 & 0.2 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{pmatrix},$$

$$\mathbf{x} := ([-1,1]\ [-2,0]\ [-\infty,\infty])^T, \quad \mathbf{b} := \begin{pmatrix} [0,0] \\ [-\infty,1] \end{pmatrix}.$$

In binary floating point arithmetic, the coefficient 0.1 cannot be represented. To ensure that the problem can be solved rigorously even in floating point arithmetic, we rewrite the unrepresentable exact problem (24) as the representable uncertain problem

$$\begin{array}{ll} \min & a^T F(x) \\ \text{s.t.} & BF(x) \in \mathbf{b}, \quad x \in \mathbf{x}, \\ \multicolumn{2}{l}{\text{for some } B \in \mathbf{B},} \end{array} \tag{25}$$

where a and $F(x)$ are as before, and

$$\mathbf{x} := \big(\,[-1,1]\ [-2,0]\ [-\infty,\infty]\,\big)^T,$$

$$\mathbf{B} := \begin{pmatrix} [\nabla 0.1, \Delta 0.1] & [0,0] & [-1,-1] & [\nabla 0.2, \Delta 0.2] & [0,0] \\ [1,1] & [0,0] & [0,0] & [0,0] & [1,1] \end{pmatrix},$$

where ∇x denotes the largest vector of floating point numbers with $\nabla x \leq x$, and Δx denotes the smallest vector of floating point number with $\Delta x \geq x$.

2.7 **Feasibility**

The traditional definition of feasibility for an optimization problem does not make sense for the uncertain constraint satisfaction problem (21). For example, in the UOP

$$\begin{array}{ll} \min & x_1 + x_2 \\ \text{s.t.} & x_1 + a x_2 = 1, \quad x_1 \in [0,2], \quad x_2 \in [0,2] \\ \multicolumn{2}{l}{\text{for some } a \in [0.79, 0.81],} \end{array} \tag{26}$$

no single point can be feasible since it cannot satisfy (26) for all $B \in \mathbf{B}$. But the problem should not be classified as infeasible since, e.g., $x_1 = 1 - a$, $x_2 = 1$ should be considered as a coefficient-dependent solution. Since a is uncertain, this "solution" comprises the set $\{x \in \mathbb{R}^2 \mid x_1 \in [0.19, 0.21],\ x_2 = 1\}$. Therefore we must generalize the definition:

A set $Z \subseteq \mathbf{x}$ is called **feasible** for the uncertain optimization problem (21) if for all $B \in \mathbf{B}$ there is an $x \in Z$ with $BF(x) \in \mathbf{F}$, **infeasible** if $BF(x) \notin \mathbf{F}$ for all $B \in B$ and $x \in Z$, and **partially feasible** otherwise. The problem (7) is called feasible (infeasible) if $\mathbf{x}$ is feasible (infeasible). The **feasible set** of (21) is the set

$$\widehat{Z} := \{x \in \mathbf{x} \mid BF(x) \in \mathbf{b} \text{ for some } B \in \mathbf{B}\}$$

of all feasible or partially feasible points of (21).

The definition implies that if the set Z is feasible then all sets $Z' \subseteq \mathbf{x}$ containing Z are also feasible. In particular, the definition applies to boxes $Z = \mathbf{z}$, and a feasible set exists iff the box $\mathbf{x}$ is feasible, i.e., iff the problem itself is feasible. The solution set is nonempty iff $\mathbf{x}$ is feasible or partially feasible.

For example (26), the box $\mathbf{z}_1 := ([1.2, 1.27]\ [0,0])^T$ is feasible, the box $\mathbf{z}_2 := ([0.9, 0.95]\ [0,0])^T$ is infeasible and the box $\mathbf{z}_3 := ([1.25, 1.25]\ [0,0])^T$ is partially feasible. The problem (26) is feasible since $\mathbf{z}_1$ is feasible, $\mathbf{z}_1 \subset \mathbf{x}$ and thus the box $\mathbf{x}$ is feasible.

Given the uncertain constraint satisfaction or optimization problem (21), we use the minimal point (6) to define the vector-valued **feasibility measure**

$$d(x) := \mu(\mathbf{B}F(x) - \mathbf{b}) \tag{27}$$

of a point $x \in \mathbb{R}^n$. For a given positive definite, diagonal scaling matrix D, the number $\|d(x)\|_D^2$ is called the **feasibility distance** of x for the problem (21).

A point x is called **δ-feasible** if $x \in \mathbf{x}$ and $\|d(x)\|_D^2 \leq \delta$, where $\delta > 0$ is a **feasibility tolerance**. In particular, feasible points are δ-feasible for every $\delta > 0$, and in an UCSP, a point is 0-feasible iff it is feasible for some choice of the uncertainties.

2.8 **Bounds on the objective and verification of feasible points**

To take account of the best known upper and lower bound in the objective function of an UOP we define

$$\begin{array}{ll} \min & a^TF(x) \in \mathbf{r} \\ \text{s.t.} & BF(x) \in \mathbf{b},\ x \in \mathbf{x}, \\ \multicolumn{2}{l}{\text{for some } a \in \mathbf{a},\ B \in \mathbf{B},} \end{array} \tag{28}$$

as shorthand for

$$\begin{array}{ll} \min & a^TF(x) \\ \text{s.t.} & a^TF(x) \in \mathbf{r},\ BF(x) \in \mathbf{b},\ x \in \mathbf{x}, \\ \multicolumn{2}{l}{\text{for some } a \in \mathbf{a},\ B \in \mathbf{B},} \end{array}$$

and similarly

$$\begin{array}{l} \min\ a^TF(x) \in \mathbf{r} \\ \text{s.t.}\ \ BF(x) \in \mathbf{b},\ x \in \mathbf{x}, \end{array} \tag{29}$$

as shorthand for

$$\begin{array}{l} \min\ a^TF(x) \\ \text{s.t.}\ \ a^TF(x) \in \mathbf{r},\ BF(x) \in \mathbf{b},\ x \in \mathbf{x}. \end{array}$$

Finding a valid upper bound on the optimal objective function value is essential for efficiently solving global optimization problems. Using an upper bound from a feasible (and ideally nearly optimal) point eliminates most of the search space – leaving a CSP with a tiny feasible region only – and therefore usually saves a large amount of time by speeding up the branch and bound process.

To find a rigorously valid upper bound requires the verification of feasible points. Verification techniques usually consist of finding a narrow box centered at a given approximately feasible point, for which it was verified that it contains a feasible point. An upper bound on the function value over this box, computed by interval evaluation, then gives arigorous upper bound on the objective function value. Of course, there may be no close feasible point, in which case a verification attempt will return without a result.

Various verification techniques are discussed by Hansen [13, Section 12] and Kearfott [16–18]; they where summarized and improved by Domes and Neumaier [7]. These verification techniques do not require to have an approximate local solution of the UOP (21) or the EOP (8); in principle, an arbitrary approximately feasible point suffices. But although finding a local optimizer takes more time, it is usually preferable over just finding an arbitrary approximately feasible point.

3 Constraint aggregation

In this section we present a novel and efficient filtering method called constraint aggregation. It consists of taking a suitable linear combination of constraints (including constraints on the objective) and applying a strong filtering method to the resulting constraint and a box defining bound constraints. The aggregator, i.e., the vector of coefficients of the linear combination, can in favorable cases be determined such that the resulting constraint intersects the box in a point or even not at all. The filtering method used on the aggregated constraint should therefore be designed so that it reduces the box in these cases to a very narrow box or to the empty set. We achieve this in the case of a quadratic constraint, so that the aggregation technique is directly applicable to quadratically constrained quadratic programs. Using quadratic underestimation techniques such as those discussed in Schichl and Markót [29, page 13], it could be extended to work for general nonlinear programs.

As demonstrated by our later numerical results, the resulting aggregation filtering method often leads to far better improvements than traditional filtering based on a quadratic constraint and bound constraints only.

3.1 **Filtering by constraint aggregation**

An **aggregator** of the constraint satisfaction problem (22) is a nonzero vector $y \in \mathbb{R}^m$. The corresponding **constraint aggregation** of (22) is the uncertain constraint

$$u^T F(x) \in \mathbf{v} := y^T \mathbf{b} \text{ for some } u \in \mathbf{u} := \mathbf{B}^T y. \tag{30}$$

Let $Ex \leq d$ a linear relaxation of the constraints of (22) over the box $\mathbf{x}$ and let $\mathcal{P}$ be a pruning method. The vector y is a **certificate of infeasibility** if $\mathcal{P}$ applied to the constraint (30), the polyhedron defined by $Ex \in \mathbf{d}$, and the bound constraints $x \in \mathbf{x}$ results in the elimination of $\mathbf{x}$.

Using centered form the two sided inequality (30) with interval coefficients can be transformed into a single scalar inequality

$$s(x) \leq \gamma, \quad s(x) := \widehat{u}^T F(x), \quad \gamma := \sup\{\overline{v} - (u - \widehat{u})^T F(x) \mid u \in \mathbf{u}, x \in \mathbf{x}\}. \tag{31}$$

(A lower bound on $s(x)$ could also be considered, but by construction of the aggregator, usually only the upper bound will have a significant effect.) Note that if $\mathbf{B}$ is the identity matrix then $\gamma = \overline{v}$. Now (30) together with $Ex \in \mathbf{d}$ and $x \in \mathbf{x}$ defines a singly-nonlinear constraint satisfaction problem as in (34).

If a valid bound on the objective of an uncertain optimization problem is known, the problem can be represented as in (28). An **aggregator** of the uncertain optimization problem (28) is a pair (ν, y) with $\nu \in \mathbb{R}$, $y \in \mathbb{R}^m$. The corresponding **constraint aggregation** of (28) is the uncertain constraint

$$wf(x) + u^T F(x) \in \mathbf{v} := \nu\mathbf{r} + y^T \mathbf{b} \text{ for some } w \in \mathbf{w} := \mathbf{a}^T \nu, \quad u \in \mathbf{u} := \mathbf{B}^T y, \tag{32}$$

and (31) changes to

$$\begin{aligned} &s(x) \leq \gamma, \quad s(x) := \widehat{w} f(x) + \widehat{u}^T F(x), \\ &\gamma := \sup\{\overline{v} - (w - \widehat{w}) f(x) - (u - \widehat{u})^T F(x) \mid w \in \mathbf{w}, u \in \mathbf{u}, x \in \mathbf{x}\}. \end{aligned} \tag{33}$$

Using rigorous filtering methods (e.g., constraint propagation [5] or linear relaxations [6]) on (33) may yield tighter bounds $x \in \widehat{\mathbf{x}} \subseteq \mathbf{x}$. Since each solution of the original problem is also a solution of (33) and $\widehat{\mathbf{x}}$ was obtained by rigorous methods, $x \in \widehat{\mathbf{x}}$ can be used as improved bound constraints without losing feasible points. This gives a cheap filtering method that reuses the information obtained from the local search procedure.

If the above method resulted in a 'significant' bound improvement the point $\widetilde{x}$ may now lie outside the new box $\widehat{\mathbf{x}}$. In this case it may be worth to start a new local search in order to find a feasible point or to further improve the bounds. In this case the $\widetilde{x}$ could be projected into the new box and taken as the starting point of the local search.

3.2 **Filtering a singly-quadratic constraint satisfaction problem**

We now introduce a new method for enclosing the feasible set of constraint satisfaction problems with linear constraints and a single quadratic constraint. In particular, it can be applied to an aggregated constraint if it is quadratic (or, if not, to a quadratic relaxation of it). In this case, the linear constraints will be those of a linear relaxation of all constraints.

We consider the uncertain **singly-quadratic constraint satisfaction problem (SQCSP)**

$$\begin{array}{ll} \text{find} & x \in \mathbf{x} \\ \text{s.t.} & c^T x + \frac{1}{2} x^T G x \leq \gamma, \\ & Ex \in \mathbf{d} \\ \text{for some} & G \in \mathbf{G},\ E \in \mathbf{E},\ \ c \in \mathbf{c}, \end{array} \tag{34}$$

where $\mathbf{x} \in \mathbb{IR}^n$ is a bounded box, $\mathbf{G} \in \mathbb{IR}^{n\times n}$ is symmetric, $\mathbf{E} \in \mathbb{IR}^{m\times n}$, $\mathbf{c} \in \mathbb{IR}^n$, and $\mathbf{d} \in \mathbb{IR}^m$. Intersecting $\mathbf{d}$ with $E\mathbf{x}$ if necessary, we may assume without loss of generality that $\mathbf{d} \subseteq E\mathbf{x}$ is bounded, too.

We compute an approximate local minimizer $\widetilde{x}$ of the quadratic program

$$\begin{array}{l} \min\ c^T x + \frac{1}{2} x^T G x \\ \text{s.t.}\ \ Ex = d,\ \ \ x \in \mathbf{x}, \end{array} \tag{35}$$

where G, E, c and d are approximate midpoints of $\mathbf{G}$, $\mathbf{E}$, $\mathbf{c}$ and $\mathbf{d}$. If γ is close to or less than the associated objective function value, one expects that the feasible domain of (34) is tiny or empty. Our goal is to transform the problem into an uncertain CSP where the single quadratic constraint becomes separable, in such a way that it becomes obvious that under the stated conditions constraint propagation will reduce the box to a narrow or empty domain.

To handle the equality constraint, we construct a rigorous null space representation as follows. We use $\widetilde{x}$ to find an estimate N of the set of indices of variables that are free at the exact minimizer of (35). We fix the variables estimated active (with indices in the complement $\neg N$) at the values determined by $\widetilde{x}$, and consider the remaining uncertain linear system

$$E_{:N} x_N = e \quad \text{for some } E \in \mathbf{E},\ e \in \mathbf{e} := \mathbf{d} - \mathbf{E}_{:\neg N} \widetilde{x}_{\neg N}.$$

Later we require $|N| \geq m$ which is only possible if the index set $\neg N$ of estimated active variables has size at most $n-m$. Since the minimizer $\widetilde{x}$ and thus the associated bound activities are only approximative, we can build the index set of the active variables

$$N^0 := \{i \mid \widetilde{x}_i \leq \underline{x}_i \text{ or } \widetilde{x}_i \geq \overline{x}_i\},$$

and the index set $N^1 \subseteq \neg N^0$ of the most approximately active variables such that $|N^1| \leq n - m - |N^0|$. Then we define $\neg N := N^0 \cup N^1$ which satisfies $|\neg N| \leq n - m$ and therefore we have $|N| \geq m$. We then redefine $\widetilde{x}$ to respect these activities exactly.

The Gauss-Jordan algorithm from Domes and Neumaier [6, pp. 13-16] is applied to an approximate midpoint of $\mathbf{E}_{:N}$, with scaling factors

$$\delta = \sqrt{\varepsilon}, \quad U = \operatorname{diag}(u), \quad V = \operatorname{diag}(v) \tag{36}$$

determined such that the equations matching the constraints with tighter bounds are preferred as pivot rows, and columns of E matching the variables with tighter bounds are preferred as pivot columns. Here ε denotes the machine precision and

$$u = (\overline{e} - \underline{e}) + \delta|\mathbf{e} - \mathbf{E}_{:N}\mathbf{x}_N|, \quad w := \max\{-\underline{x}_i,\ \overline{x}_j \mid i, j \in N\},$$
$$\mathbf{w} = \mathbf{x}_N \cap [-w, w]^{|N|}, \quad v = (\overline{w} - \underline{w})/\max\{\overline{w}_k - \underline{w}_k \mid k = 1, \dots, |N|\}.$$

This produces an index list $P \subseteq N$ of size $p := |P|$ and a matrix $C \in \mathbb{R}^{p\times m}$ such that

$$CE_{:P} \approx I \in \mathbb{R}^{p\times p}.$$

By construction, $E_{:P}$ is likely to be invertible, and we compute enclosures $\mathbf{X}$, $\mathbf{S}$ of

$$X := E_{:P}^{-1} \in \mathbb{R}^{p\times p}, \quad S := XE_{:Q} \in \mathbb{R}^{p\times(n-p)},$$

where $Q := \{1, \dots, n\} \setminus P$ is the complementary index list of size $|Q| = n - p$. This may be done, e.g., by computing

$$\mathbf{X} := (C\mathbf{E}_{:P})^G C, \quad \mathbf{S} := \mathbf{X}\mathbf{E}_{:Q}, \tag{37}$$

where $\mathbf{A}^G\mathbf{B}$ is the interval matrix obtained by applying interval Gauss elimination (see e.g., Neumaier [24, pp. 152-166]) to the uncertain linear equation $AX = B$ for some $A \in \mathbf{A}$ and some $B \in \mathbf{B}$. Note that in the present case, the coefficient matrix is nearly the identity, so that interval Gauss elimination should not suffer from excessive overestimation. If interval Gauss elimination fails, the problem is considered degenerate, and no relaxation is computed.

The columns of the matrix $Z \in \mathbb{R}^{n\times(n-p)}$ defined by $Z_{P:} = -S$ and $Z_{Q:} = I$ form a basis of the null space of the matrix E; indeed, we have

$$XEZ = XE_{:P}(-S) + XE_{:Q}I = -S + S = 0,$$

hence $EZ = 0$. Assuming for the moment that the P indices are sorted before the Q indices, we have $Z = \binom{-S}{I}$, and the reduced Hessian $G_{\mathrm{red}} \in \mathbb{R}^{(n-p)\times(n-p)}$ takes the form

$$G_{\mathrm{red}} := Z^TGZ = (-S^T\ \ I)\begin{pmatrix} G_{PP} & G_{PQ} \\ G_{QP} & G_{QQ} \end{pmatrix}\begin{pmatrix} -S \\ I \end{pmatrix}.$$

It is easily seen that the resulting equation

$$G_{\mathrm{red}} = S^TG_{PP}S - S^TG_{PQ} - G_{QP}S + G_{QQ}$$

remains valid even when the indices are not sorted. From a directed modified Cholesky factorization (cf. Domes and Neumaier [9]) of the enclosure

$$\mathbf{G}_{\mathrm{red}} = \mathbf{S}^T(\mathbf{G}_{PP}\mathbf{S} - \mathbf{G}_{PQ}) - \mathbf{G}_{QP}\mathbf{S} + \mathbf{G}_{QQ} \tag{38}$$

of G_{red} with $M := \{i \mid Q_i \in N\}$ and $\zeta = 10^{-6}$, we may obtain a nonsingular matrix $R \in \mathbb{R}^{(n-p)\times(n-p)}$ and a diagonal matrix $D \in \mathbb{R}^{(n-p)\times(n-p)}$ such that the residual matrix

$$\varDelta := \mathbf{G}_{\mathrm{red}} + D - R^TR \tag{39}$$

is positive semidefinite and tiny. Note that the bracketing in (38) improves the enclosure, which can be improved further by intersecting $\mathbf{G}_{\mathrm{red}}$ with its transpose, which is valid since G_{red} is symmetric.

With the approximate solution $\widetilde{x}$ found by a local solver, we form

$$\widetilde{d} := Ex - E\widetilde{x} \in \widetilde{\mathbf{d}} := \mathbf{d} - \mathbf{E}\widetilde{x} \approx 0. \tag{40}$$

The for any feasible x, the correction vector

$$s := x - \widetilde{x}$$

satisfies

$$Es = \widetilde{d} \in \widetilde{\mathbf{d}}, \quad s \in \mathbf{s} := \mathbf{x} - \widetilde{x}. \tag{41}$$

Moreover, since $s_P + Ss_Q = X(E_{:P}s_P + E_{:Q}s_Q) = XEs = X\widetilde{d}$, we have

$$(Zs_Q)_P = Z_{P:}s_Q = -Ss_Q = s_P - X\widetilde{d}, \quad (Zs_Q)_Q = Z_{Q:}s_Q = s_Q,$$

hence

$$r := s - Zs_Q \in \mathbf{r} \approx 0,$$

where

$$\mathbf{r}_P := \mathbf{X}\widetilde{\mathbf{d}}, \quad \mathbf{r}_Q := [0, 0]. \tag{42}$$

Thus $x = \widetilde{x} + s = x' + Zs_Q$, where

$$x' = \widetilde{x} + r \in \mathbf{x}' := \widetilde{x} + \mathbf{r}, \tag{43}$$

and

$$\begin{aligned}\widetilde{\sigma} &:= 2c^Tx + x^TGx - (2c^Tx' + x'^TGx')\\ &= 2g^TZs_Q + (Zs_Q)^TGZs_Q\\ &= h^Ts_Q + s_Q^TG_{\mathrm{red}}s_Q,\end{aligned}$$

where

$$g := c + Gx' \in \mathbf{g} := \mathbf{c} + \mathbf{G}\mathbf{x}', \tag{44}$$

$$h := 2Z^Tg = 2(g_Q - S^Tg'_P) \in \mathbf{h} := 2(\mathbf{g}_Q - \mathbf{S}^T\mathbf{g}_P). \tag{45}$$

Since

$$s_Q^TG_{\mathrm{red}}s_Q + s_Q^T(R^TR - D)s_Q + s_Q^T\Delta s_Q \geq (Rs_Q)^T(Rs_Q) - s_Q^TDs_Q,$$

we may introduce

$$z := Rs_Q \in \mathbf{z} := R\mathbf{s}_Q, \tag{46}$$

and find the separable quadratic relaxation

$$z^Tz - s_Q^TDs_Q + h^Ts_Q \leq \sigma := \sup\Big(2\gamma - (\mathbf{c} + \mathbf{g})^T\mathbf{x}'\Big), \tag{47}$$

where σ bounds $\widetilde{\sigma}$, and the linear constraints

$$Es \in \widetilde{d}, \quad s_P + Ss_Q \in \mathbf{r}_P, \quad Rs_Q - z = 0, \quad s \in \mathbf{s}, \quad z \in \mathbf{z}. \tag{48}$$

With an approximate constraint multiplier y for (34) at $\widetilde{x}$ (computed by Algorithm 1, or simply $y = 0$), we may introduce

$$h' := 2(g - E^Ty) \in \mathbf{h}' := 2(\mathbf{g} - \mathbf{E}^Ty), \tag{49}$$

and rewrite the linear term $h^T s_Q$ in (47) as

$$h^T s_Q = 2g^T Z s_Q = (h'^T + y^T E) Z s_Q = h'^T Z s_Q = h'^T (s - r)$$

since $EZ = 0$. This gives the alternative separable quadratic relaxation

$$z^T z - s_Q^T D s_Q + h'^T s \le \sigma' := \sigma + \sup(\mathbf{h}'^T \mathbf{r}). \tag{50}$$

Therefore s and z are solutions of the uncertain quadratic constraint satisfaction problem

$$\begin{array}{ll} \text{find} & s \in \mathbf{s},\ z \in \mathbf{z}, \\ \text{s.t.} & z^T z - s_Q^T D s_Q + h^T s_Q \le \sigma, \\ & z^T z - s_Q^T D s_Q + h'^T s \le \sigma', \\ & Es \in \widetilde{\mathbf{d}}, \quad (CE)s \in C\widetilde{\mathbf{d}}, \\ & s_P + S s_Q \in \mathbf{r}_P, \quad R s_Q - z = 0, \\ \multicolumn{2}{l}{\text{for some } E \in \mathbf{E}, \quad h \in \mathbf{h}, \quad h' \in \mathbf{h}', \quad S \in \mathbf{S}.} \end{array} \tag{51}$$

depending on the $n + (n - p)$ variables x and z and consisting of two separable quadratic inequality constraints and $m+p+(n-p) = m+n$ linear equality constraints.

Algorithm 2: Quadratic Filtering (QuadFil)

Input: An SQCSP given by (34) , the local solver precision $\delta \ll 1$ and an arbitrary starting point $x^0 \in \mathbf{x}$.
Output: A reduced box $\mathbf{x}^{\text{new}} \subseteq \mathbf{x}$ (or the empty set), containing all solutions of (34).

```
Solve the quadratic program (35) by an approximate local solver starting from the
  point x^0, to precision δ, and obtain the optimizer x̃ ∈ ℝ^n;
if x̃_i < min(x̲_i + δ, x̄_i) then x̃_i ← x̲_i;
if max(x̲_i, x̄_i − δ) < x̃_i then x̃_i ← x̄_i;
Form the index set N := {i | x̲_i < x̃_i < x̄_i} and compute e := d − E_{:¬N} x̃_{¬N};
Use the Gauss-Jordan algorithm for mid(E_{:N}), with scaling factors U, V and δ
  determined as in (36). This results in the index list P ⊆ N and the matrix C;
Use the interval Gauss elimination on the interval system of equations (CE_{:P})X = C
  to find the solution set X ∈ IR^{p×p};
Compute Q := {1, ..., n} \ P and S := XE_{:Q};
Partition G and compute the enclosure G_red for the reduced Hessian as given by (38);
Improve the enclosure G_red by computing G_red ← G_red ∩ G_red^T;
Use the directed modified Cholesky factorization (Domes and Neumaier [9, Algorithm
  ModDirChol]) with M := {i | Q_i ∈ N} and ζ = 10^{−6} to find a matrices R and D such
  that the residual matrix defined by (39) is positive semidefinite for all G_red ∈ G_red;
if the directed modified Cholesky factorization failed then return signaling failure;
else
    Compute d̃ by (40), s by (41), r by (42), x′ by (43), g by (44), h by (45), z by (46)
      and δ by (47);
    Find the approximate constraint multiplier y for (34) at x̃ by using Algorithm 1
      and compute h′ by (49) and δ′ by (50);
    Create the uncertain quadratic constraint satisfaction problem (51);
    Use a rigorous filtering method (e.g., quadratic separable constraint propagation)
      on (51) in order to obtain tighter bounds s^new and z^new for the variables s and z;
    if s^new or z^new is empty then return that x contains no solution of (34);
    else return the box x^new := x̃ + s^new;
end
```

Note that (as as discussed in Domes and Neumaier [6, Section 3]) adding the redundant constraint $(CE)s \in C\widetilde{\mathbf{d}}$ in (51) sometimes significantly improves the quality of the enclosure, though the main effect of the filtering by aggregation is due to the other constraints.

A constraint propagator that optimally handles single linear and separable quadratic constraints (see Domes and Neumaier [5]) produces an improved enclosure $\mathbf{s}^{\text{new}}$ for s or proves that (51) and therefore (34) is infeasible. In the first case one recovers an improved enclosure $\mathbf{x}^{\text{new}} := \widetilde{x} + \mathbf{s}^{\text{new}}$ of solution set of (34).

Algorithm 2 turns the above considerations into a precise prescription.

4 Choosing the aggregator

It remains to discuss the choice of aggregators. Clearly, an arbitrary chice is unlikely to be beneficial; for example if we take as aggregator a $(0, 1)$ unit vector, we just recover the original constraints, without any advantage.

In this section, we discuss two sensible choices. The first choice is based on the solution of an auxiliary least squares problem and gives under suitable conditions an aggregator that provides a certificate of infeasibility, reducing the box defining the bound constraints to the empty set. The second choice utilizes the Lagrange multipliers of an approximately optimal point.

4.1 **Certificates of infeasibility**

If for the uncertain optimization or constraint satisfaction problem (21) a local search yields no (weakly) feasible point but an infeasible $\widetilde{x} \in \mathbf{x}$, the nonzero, signed feasibility measure vector (27) can be used as an aggregator.

The following theorem gives sufficient conditions under which an appropriate aggregator may serve as a certificate of infeasibility, proving that the aggregated constraint is trivially infeasible. If these conditions are not satisfied by a limited margin only, one may expect that the aggregated constraint, while usually not sufficient to prove infeasibility, will still be strong enough to reduce the box when the reduction technique described above is applied.

Theorem 2 *Let F be continuously differentiable, and let D be a scaling matrix. Let $\widehat{x}$ be a stationary point of*

$$\begin{array}{ll} \min & f(x) := \frac{1}{2}||\mu(BF(x) - \mathbf{b})||_D^2 \\ \text{s.t.} & x \in \mathbf{x} \end{array} \tag{52}$$

and let

$$y := D\mu(BF(\widehat{x}) - \mathbf{b}), \quad u := B^T y. \tag{53}$$

(i) $f(x)$ is continuously differentiable with

$$\frac{\partial f(x)}{\partial x_k} = F'(x)^T B^T D d(x) \tag{54}$$

and $g := F'(\widehat{x})^T u$ satisfies

$$\begin{cases} g_i \geq 0 & \textit{if } \underline{x}_i = \widehat{x}_i < \overline{x}_i, \\ g_i \leq 0 & \textit{if } \underline{x}_i < \widehat{x}_i = \overline{x}_i, \\ g_i = 0 & \textit{if } \underline{x}_i < \widehat{x}_i < \overline{x}_i. \end{cases} \tag{55}$$

(ii) The inequality

$$u^T F(x) \in y^T \mathbf{b} < u^T F(\widehat{x}) \tag{56}$$

holds for all feasible x, with equality only if $y = 0$.

(iii) If $y = 0$ then $\widehat{x}$ is feasible.

(iv) If $y \neq 0$ and $u = 0$ then there is no feasible point.

(v) If $y \neq 0$ and F is linear then y is a certificate of infeasibility.

Proof (iii) D is nonsingular and $y = 0$; therefore $\mu(BF(\widehat{x}) - \mathbf{b}) = 0$, giving

$$0 \in BF(\widehat{x}) - \mathbf{b} \implies BF(\widehat{x}) \in \mathbf{b}.$$

Since $\widehat{x} \in \mathbf{x}$ by construction $\widehat{x}$ must be feasible.

(i) Write $d(x) := \mu(BF(x) - \mathbf{b})$. Then

$$\frac{\partial d_i(x)}{\partial x_k} = \begin{cases} (BF'(x))_{ik} & \text{if } d_i(x) \neq 0, \\ 0 & \text{otherwise,} \end{cases}$$

is a continuous partial derivate except when $(BF(x))_i \in \{\underline{b}_i, \overline{b}_i\}$. Since

$$f(x) = \frac{1}{2}||d(x)||_D^2 = \frac{1}{2} d(x)^T D d(x)$$

vanishes at each point of discontinuity, the derivative

$$\frac{\partial f(x)}{\partial x_k} = \frac{\partial d(x)^T}{\partial x_k} D d(x) = \sum_i D_{ii} \frac{\partial d_i(x)}{\partial x_k} d_i$$

is continuous. (54) follows from

$$\begin{aligned} \frac{\partial f(x)}{\partial x_k} &= \sum_i \frac{\partial d_i(x)}{\partial x_k} D_{ii} d_i(x) = \sum_i [BF'(x)]_{ik} D_{ii} d_i(x) \\ &= [BF'(x)]_{:k}^T D d(x) = [F'(x)^T B^T D d(x)]_{:k}. \end{aligned}$$

Since $y = Dd(\widehat{x})$ and $u = B^T y$, we have

$$\frac{\partial f(\widehat{x})}{\partial x} = F'(x)^T B^T D d(x) = F'(\widehat{x})^T u = g. \tag{57}$$

The first order necessary conditions for optimality now give

$$\frac{\partial f(\widehat{x})}{\partial x_k} \geq 0 \;\text{ if } \underline{x}_i = \widehat{x}_i < \overline{x}_i$$

$$\frac{\partial f(\widehat{x})}{\partial x_k} \leq 0 \;\text{ if } \underline{x}_i < \widehat{x}_i = \overline{x}_i$$

$$\frac{\partial f(\widehat{x})}{\partial x_k} = 0 \;\text{ if } \underline{x}_i < \widehat{x}_i < \overline{x}_i,$$

and using (57), we find (55).

(ii) Let $y \neq 0$. Then the constraints can be aggregated, resulting in $u^T F(x) \in y^T \mathbf{b}$. By construction of y we have

$$\begin{array}{ll} (BF(\widehat{x}))_i > \overline{b}_i & \text{if } y_i > 0, \\ (BF(\widehat{x}))_i < \underline{b}_i & \text{if } y_i < 0, \\ (BF(\widehat{x}))_i \in \mathbf{b}_i & \text{if } y_i = 0. \end{array}$$

From this it follows that $y_i(BF(\widehat{x}))_i \geq \sup(y_i \mathbf{b}_i)$, with equality only if $y_i = 0$. We conclude that

$$u^T F(\widehat{x}) = y^T BF(\widehat{x}) = \sum y_i(BF(\widehat{x}))_i \geq \sup \sum y_i \mathbf{b}_i = \sup(y^T \mathbf{b}), \tag{58}$$

with equality only if all $y_i = 0$. However $y = 0$ leads to a contradiction. Therefore (58) holds with strict inequality, proving (ii).

(iv) is an immediate consequence of (ii).

(v) If F is linear then $F(x) = F_0 + F_0' x$, $F'(x) = F_0'$ and $g = F'(\widehat{x})^T u = (F_0')^T u$, leading to

$$\begin{aligned} u^T(F(\widehat{x}) - F(x)) &= u^T(F_0 + F_0'\widehat{x} - (F_0 + F_0' x)) = u^T F_0'(\widehat{x} - x) \\ &= g^T(\widehat{x} - x) = \textstyle\sum_i g_i(\widehat{x}_i - x_i). \end{aligned} \tag{59}$$

By (55),

$$\begin{array}{lll} \widehat{x}_i = \underline{x}_i & \Longrightarrow & \widehat{x}_i - x_i = \underline{x}_i - x_i \leq 0, \quad g_i \geq 0, \\ \widehat{x}_i = \overline{x}_i & \Longrightarrow & \widehat{x}_i - x_i = x_i - \overline{x}_i \geq 0, \quad g_i \leq 0, \\ \widehat{x}_i \in \iint \mathbf{x}_i & \Longrightarrow & g_i = 0, \\ \widehat{x}_i = \underline{x}_i = \overline{x}_i & \Longrightarrow & \widehat{x}_i - x_i = 0. \end{array}$$

In all cases, $g_i(\widehat{x}_i - x_i) \leq 0$, and by (59) we conclude that $u^T(F(\widehat{x}) - F(x)) \leq 0$. On the other hand, (56) gives the inequality $0 \leq u^T(F(\widehat{x}) - F(x))$ for all feasible x, with equality only if $y = 0$. Therefore $y = 0$ for all feasible x. □

4.2 Aggregation heuristics

Theorem 2 suggests that we use the feasibility violation vector as an aggregator. Indeed, we found it suitable in the case when a box is unlike to contain a nearly feasible point. On the other hand, if we know a nearly feasible point, the feasibility violation vector typically consists of noise only, and we need a different aggregator. The introductory example suggests that we use in this case a Lagrange multiplier. Uncertainties can be ignored in the computation of the multipliers, thus we use approximate midpoints to define the problem passed to Algorithm 1.

Algorithm 3: Aggregator chooser (AGGRCH)

Input: An uncertain optimization problem given by (28), the point $x \in \mathbf{x}$ and the feasibility tolerance $\delta \geq 0$.

Output: An objective and constraints aggregator (ν, y) as well as the aggregator type `aggrt` $\in \{$ `feas`, `mult`$\}$.

1 Compute the feasibility violation vector $y := \mu(\mathbf{B}F(x) - \mathbf{b})$ and the objective violation $\nu := \mu(\mathbf{a}^T f(x) - \mathbf{r})$;

2 **if** $\|(\nu, y)\|_D^2 \leq \delta$ **then**

3 Recompute (ν, y) by applying Algorithm 1 to x and the midpoint approximation of (28). Also put `aggrt=mult`;

4 **else** put `aggrt=feas`;

5 **return** (ν, y) and `aggrt`;

5 Numerical results

In this section we present numerical results for the new constraint aggregation method on a large set of test problems. Our tests simulate the situation in a branch and bound scheme when a narrow box containing the global solution is processed. This case is especially interesting since this is the point where the traditional filtering methods usually lose their efficiency, resulting in excessive splitting due to the cluster effect.

5.1 Testing procedure

We selected from the COCONUT Environment Testset ([32]) all constrained quadratic optimization problems with no more than 300 variables and no more than 300 constraints, resulting in 135 problems. The TEST ENVIRONMENT [10] provides for each problem an approximate (global) solution x^*, the best approximation among those found by a number of solvers applied to the problem.

We performed two kinds of tests, one to test the contraction properties when a box contains the global minimizer, and one to test the elimination properties when a box is close to the global minimizer but does not contain it. Both cases are important to assess the degree to which the cluster effect can be reduced by our new method.

The first test therefore tests the case of aggregation by multiplier, given an approximate minimizer x^* (which we take as the best point provided by the TEST ENVIRONMENT). For each problem, we construct a box with tiny `start_box_radius` r around x^* and intersect it with the original bound constraints to get the test box $\mathbf{x}^*$. We impose on the objective function the upper bound

$$f(x) \leq \overline{f} := f(x^*) + f_\epsilon, \quad f_\epsilon := \max(e_a, e_r|f(x^*)|),$$

where $0 < e_a \ll 1$ is an absolute and $0 < e_r \ll 1$ a relative error factor. These factors have to be chosen carefully since choosing them to small may result in an infeasible problem (since x^* is usually only approximately feasible), and choosing them to large may prevent the contraction of $\mathbf{x}^*$. In this test we first choose them as $e_a = e_r = 0$ then increase them (to $e_a = e_r = 10^{-14}$ and then by successively multiplication with 10) until the problem becomes feasible for the constructed box $\mathbf{x}^*$. The f_ϵ used for each problem can be found in the `objsh` column of the detailed results table.

Each test problem is represented in the form (28) with $\mathbf{r} := [-\infty, \overline{f}]$, $\mathbf{x} := \mathbf{x}^*$. We first apply constraint propagation to the problem in order to eliminate the dominant effects of a contraction obtained by traditional zero order filtering methods. Then a local solver is used to find an approximately feasible point $\widehat{x} \in x^*$. If $\widehat{x}$ is not δ-feasible, the nonzero, signed feasibility distance vector (27) is used to aggregate as in (31). If $\widehat{x}$ is δ-feasible, a local search is started from $\widehat{x}$ to obtain a local minimum of (28) and $\widehat{x}$ is replaced by the local minimum found. In this case, the multipliers (ν, y) at the local minimum were used to aggregate as in (33). This step is referred as the constraint aggregation (`AG`). Then the constraints are linearly relaxed around $\widehat{x}$ inside the box $\mathbf{x}^*$, and the new system is solved by linear constraint propagation (`LR`). Finally the quadratic filtering Algorithm 2 (`QF`) is applied.

In the second test we assess the quality of aggregation by feasibility violation, applied to boxes that are very close to the solution but contain no feasible point. For each problem we consider the small box $\mathbf{x}'$ around the solution obtained in the first test and construct $2n$ ($n =$ problem dimension) boxes

$$\mathbf{x}^{(k)} \text{ with } \mathbf{x}_i^{(k)} := \begin{cases} [\underline{x}'_i - \kappa - 2r, \underline{x}'_i - \kappa] & \text{if } k \leq n \text{ and } k = i \\ [\overline{x}'_i + \kappa, \overline{x}'_i + \kappa + 2r] & \text{if } k > n \text{ and } k - n = i \\ \mathbf{x}'_i & \text{otherwise,} \end{cases}$$

where κ is a small `infeasible_box_shift` and r is again the `start_box_radius`. Then we apply `AG`, `LR` and `QF` to each box and count how often they prove that the box contains no feasible point. We define the **elimination factor** `elim`$:= e/n$, where e denotes the number of eliminated boxes.

5.2 **Test results**

We applied the above test procedures to the 135 test problems selected, with the following test settings:

`start_box_radius`	$= 1{\cdot}10^{-2}$
`feasibility_tolerance`	$= 1{\cdot}10^{-6}$
`infeasible_box_shift`	$= 1{\cdot}10^{-6}$
`minimal_gain`	$= 1{\cdot}10^{-1}$
`maximum_iteration`	$= 5$
`aggregator_selection_type`	= Multipliers
`solver_accuracy`	$= 1{\cdot}10^{-10}$
`solver_maximum_iteration`	$= 200$
`use_ehull`	= on
`adjust_reference_solutions`	= on
`forced_use_of_reference_solutions`	= on
`use_the_first_separable_constraint`	= on
`use_the_multipliers_at_`$\widetilde{x}$`_instead_of_`$y = 0$	= on

The results are given in a large table, using the following abbreviations:

`n`	– number of variables (n),
`m`	– number of constraints (m),
`objsh`	– objective upper bound shift,
`start box`	– start box parameters,
`final box`	– final box parameters,
`s`	– number of steps,
`ag`	– aggregator type: multipliers, feasibility or none,
`AG`	– constraint aggregation,
`LR`	– linear relaxation,
`QF`	– quadratic filtering,
`ratio`	– reduction ratio, values between 0 (full reduction) and 1 (-)(no gain),
`wid`	– maximal-width measure ($\max_{i=1}^{n} \mathrm{width}(\mathbf{x}_i)$),
`cub`	– cube measure ($\sqrt[n]{\prod_{i=1}^{n} \mathrm{width}(\mathbf{x}_i)}$),
`elim`	– the infeasible box elimination factor (e/n).

Due to the poor reference solutions for 6 problems we could not create a box around them which contained any feasible points. For these problems, no infeasible boxes were constructed.

5.3 **Discussion**

The following table summarizes the test results.

Test Result Summary (135 problems)	`AG`	`LR`	`QF`
First step; arithmetic mean cube-ratio	0.42	0.83	0.33
First step; arithmetic mean max-width-ratio	0.67	0.98	0.57
First step; geometric mean cube-ratio	0.19	-	0.31
First step; geometric mean max-width-ratio	0.11	0.55	0.14
First step; one component width reduced below $1 \cdot 10^{-8}$	21	2	16
First step; all component widths reduced below $1 \cdot 10^{-8}$	9	1	8
Additional steps; arithmetic mean cube-ratio	0.99	0.99	0.98
Additional steps; arithmetic mean max-width-ratio	1	1	1
Infeasible problems	6		
Elimination test success ratio; arithmetic mean	0.82		
Elimination test success ratio; geometric mean	0.78		

The summary table shows that the method – and especially the aggregation (`AG`) and the quadratic filter (`QF`) part – strongly contracts most boxes containing the solution. Performing more than one step of the method does not improve the quality anymore. Since the boxes to be contracted and the used objective upper bound depend on the approximate reference solution used (which sometimes was not very accurate), we did not expect contraction to a single point; however for 18 problems, all component widths were reduced below $1 \cdot 10^{-8}$).

The elimination test shows that for each problem, most of the $2n$ boxes created near to the solution were eliminated, hence proved to contain no feasible point. Since the tests focus on the most difficult problem of filtering boxes very close to the global solution, this shows that our method is indeed a powerful tool for eliminating the cluster effect.

The performance of the new techniques within a framework for global optimization depends on how these techniques are combined with other, more traditional methods. We intend to report on this in a separate paper [8] describing the Java implementation JGloptLab of our earlier GloptLab constraint satisfaction package [4].

A Appendix: Detailed test results

problem	n	m	objsh	s	ag	start box		AG ratio		LR ratio		QF ratio		final box		elim
						cub	wid	cub	wid	cub	wid	cub	wid	cub	wid	
abel	28	14	0	1	mul	$1.95 \cdot 10^{-2}$	$2 \cdot 10^{-2}$	0	0.03	1	-	0.3	-	$2.31 \cdot 10^{-4}$	$5.05 \cdot 10^{-4}$	0.68
aircrftb	15	10	0	1	mul	$1.22 \cdot 10^{-2}$	$2 \cdot 10^{-2}$	1	-	0.31	-	0	$5 \cdot 10^{-3}$	$8.45 \cdot 10^{-6}$	$9.73 \cdot 10^{-5}$	0.87
				1	mul	$8.45 \cdot 10^{-6}$	$9.73 \cdot 10^{-5}$	-	-	1	-	0.99	1	$8.44 \cdot 10^{-6}$	$9.73 \cdot 10^{-5}$	
airport	84	42	$4.8 \cdot 10^{-4}$	1	mul	$1.64 \cdot 10^{-2}$	$2 \cdot 10^{-2}$	1	-	-	-	0	0.15	$6.19 \cdot 10^{-4}$	$2.93 \cdot 10^{-3}$	0.98
				1	mul	$6.19 \cdot 10^{-4}$	$2.93 \cdot 10^{-3}$	-	-	-	-	0.98	1	$6.19 \cdot 10^{-4}$	$2.93 \cdot 10^{-3}$	
aljazzaf	3	1	0	1	mul	$3.05 \cdot 10^{-3}$	$1 \cdot 10^{-2}$	0	$6 \cdot 10^{-5}$	-	-	0.13	0.77	$3.79 \cdot 10^{-10}$	$9.77 \cdot 10^{-8}$	1
biggsc4	4	7	$2.5 \cdot 10^{-6}$	1	mul	$2 \cdot 10^{-2}$	$2 \cdot 10^{-2}$	-	-	-	-	$7 \cdot 10^{-6}$	0.05	$1.04 \cdot 10^{-3}$	$1.05 \cdot 10^{-3}$	1
bt1	2	1	0	1	mul	$1 \cdot 10^{-3}$	$2 \cdot 10^{-2}$	$3 \cdot 10^{-13}$	$6 \cdot 10^{-5}$	-	-	-	-	$5.01 \cdot 10^{-10}$	$1.26 \cdot 10^{-6}$	1
bt12	5	3	0	1	mul	$1.27 \cdot 10^{-2}$	$2 \cdot 10^{-2}$	0	$1 \cdot 10^{-5}$	0.99	-	0.98	1	$1.33 \cdot 10^{-7}$	$2.77 \cdot 10^{-7}$	1
				1	mul	$1.33 \cdot 10^{-7}$	$2.77 \cdot 10^{-7}$	1	-	1	-	1	-	$1.33 \cdot 10^{-7}$	$2.77 \cdot 10^{-7}$	
bt13	5	1	0	1	mul	$4.37 \cdot 10^{-2}$	$2 \cdot 10^{-2}$	-	-	-	-	0	-	0	0	0.8
bt3	5	3	0	1	mul	$1.29 \cdot 10^{-2}$	$2 \cdot 10^{-2}$	-	-	-	-	0	$1 \cdot 10^{-5}$	$1.24 \cdot 10^{-7}$	$2.2 \cdot 10^{-7}$	1
bt8	5	2	0	1	mul	$8.39 \cdot 10^{-3}$	$2 \cdot 10^{-2}$	$4 \cdot 10^{-12}$	-	-	-	1	-	$4.51 \cdot 10^{-5}$	$2 \cdot 10^{-2}$	0.6
congigmz	3	5	$2.8 \cdot 10^{-6}$	1	mul	$1.45 \cdot 10^{-2}$	$1.75 \cdot 10^{-2}$	0.02	-	1	-	$1 \cdot 10^{-4}$	0.41	$1.89 \cdot 10^{-4}$	$9.59 \cdot 10^{-4}$	0.67
				1	mul	$1.89 \cdot 10^{-4}$	$9.59 \cdot 10^{-4}$	1	-	1	-	1	-	$1.89 \cdot 10^{-4}$	$9.59 \cdot 10^{-4}$	
degenlpa	20	14	0	No feasible box found!												
demymalo	3	3	$3 \cdot 10^{-6}$	1	mul	$9.65 \cdot 10^{-3}$	$1.5 \cdot 10^{-2}$	$5 \cdot 10^{-11}$	$6 \cdot 10^{-4}$	-	-	0.17	-	$1.93 \cdot 10^{-6}$	$3 \cdot 10^{-6}$	0.67
discs	33	66	$1.2 \cdot 10^{-7}$	1	mul	$1.5 \cdot 10^{-2}$	$2 \cdot 10^{-2}$	1	-	0.41	-	0	0.03	$1.39 \cdot 10^{-5}$	$5.92 \cdot 10^{-4}$	0.8
				1	mul	$1.39 \cdot 10^{-5}$	$5.92 \cdot 10^{-4}$	-	-	1	-	1	-	$1.39 \cdot 10^{-5}$	$5.92 \cdot 10^{-4}$	
dispatch	4	2	$3.2 \cdot 10^{-5}$	1	mul	$1.14 \cdot 10^{-2}$	$2 \cdot 10^{-2}$	-	-	0.95	-	$2 \cdot 10^{-3}$	-	$2.47 \cdot 10^{-3}$	$2 \cdot 10^{-2}$	0.62
dual1	85	1	0	1	mul	$1.46 \cdot 10^{-2}$	$2 \cdot 10^{-2}$	-	-	-	-	0	0.14	$2.87 \cdot 10^{-4}$	$1.98 \cdot 10^{-3}$	0.86
				1	mul	$2.87 \cdot 10^{-4}$	$1.98 \cdot 10^{-3}$	-	-	-	-	1	-	$2.87 \cdot 10^{-4}$	$1.98 \cdot 10^{-3}$	
dual2	96	1	$1 \cdot 10^{-9}$	1	mul	$1.64 \cdot 10^{-2}$	$2 \cdot 10^{-2}$	-	-	-	-	0	0.01	0	$1.94 \cdot 10^{-4}$	0.92
dual3	111	1	0	1	mul	$1.55 \cdot 10^{-2}$	$2 \cdot 10^{-2}$	-	-	-	-	0	0.04	0	$7.07 \cdot 10^{-4}$	0.91
				1	mul	0	$7.07 \cdot 10^{-4}$	-	-	-	-	-	1	0	$7.04 \cdot 10^{-4}$	
dual4	75	1	0	1	mul	$1.68 \cdot 10^{-2}$	$2 \cdot 10^{-2}$	-	-	-	-	0	0.03	$7.47 \cdot 10^{-5}$	$4.69 \cdot 10^{-4}$	0.91
				1	mul	$7.47 \cdot 10^{-5}$	$4.69 \cdot 10^{-4}$	-	-	-	-	1	1	$7.47 \cdot 10^{-5}$	$4.69 \cdot 10^{-4}$	

problem	n	m	objsh	s	ag	start box		AG ratio		LR ratio		QF ratio		final box		elim
						cub	wid	cub	wid	cub	wid	cub	wid	cub	wid	
dualc1	9	13	0	1	mul	$1.26\cdot10^{-2}$	$2\cdot10^{-2}$	$7\cdot10^{-5}$	-	0.98	-	0	0.03	$4.32\cdot10^{-7}$	$6.26\cdot10^{-4}$	0.67
dualc2	7	9	0	1	mul	$1.35\cdot10^{-2}$	$2\cdot10^{-2}$	0.2	-	-	-	0	0.02	$7.77\cdot10^{-7}$	$4.64\cdot10^{-4}$	0.71
				1	mul	$7.77\cdot10^{-7}$	$4.64\cdot10^{-4}$	-	-	-	-	1	-	$7.77\cdot10^{-7}$	$4.64\cdot10^{-4}$	
eigmaxa	101	101	0	1	mul	$1.96\cdot10^{-2}$	$2\cdot10^{-2}$	0.46	-	0	-	-	$1\cdot10^{-9}$	0	$1.14\cdot10^{-13}$	1
eigmaxb	101	101	0	1	mul	$2\cdot10^{-2}$	$2\cdot10^{-2}$	-	-	-	-	0	$3\cdot10^{-12}$	0	$5.35\cdot10^{-14}$	1
eigmaxc	22	22	0	1	mul	$3.38\cdot10^{-3}$	$2\cdot10^{-2}$	0.03	-	0	-	0	0.08	$3.13\cdot10^{-6}$	$1.56\cdot10^{-3}$	1
				1	mul	$3.13\cdot10^{-6}$	$1.56\cdot10^{-3}$	1	-	0.11	-	0.36	-	$2.7\cdot10^{-6}$	$1.56\cdot10^{-3}$	
eigminb	101	101	0	1	mul	$1.89\cdot10^{-2}$	$2\cdot10^{-2}$	0.56	-	0.39	-	0	$2\cdot10^{-11}$	0	$4.68\cdot10^{-13}$	0.99
eigminc	22	22	0	No feasible box found!												
ex14-1-6	9	15	$1\cdot10^{-8}$	1	mul	$1.14\cdot10^{-2}$	$2\cdot10^{-2}$	0	-	$7\cdot10^{-3}$	-	0	1	$2.98\cdot10^{-10}$	$1.22\cdot10^{-9}$	0.89
ex2-1-1	5	1	0	1	mul	$1\cdot10^{-2}$	$1\cdot10^{-2}$	0	$2\cdot10^{-13}$	0.88	-	0.58	0.94	$1.43\cdot10^{-15}$	$1.78\cdot10^{-15}$	0.5
ex2-1-10	20	10	0	1	mul	$1.04\cdot10^{-2}$	$2\cdot10^{-2}$	0	$7\cdot10^{-6}$	0.64	-	0.54	-	$2.71\cdot10^{-9}$	$7.64\cdot10^{-8}$	0.8
				1	mul	$2.71\cdot10^{-9}$	$7.64\cdot10^{-8}$	1	-	-	-	0.32	-	$2.56\cdot10^{-9}$	$4.39\cdot10^{-8}$	
ex2-1-2	6	2	$2.1\cdot10^{-5}$	1	mul	$4.64\cdot10^{-3}$	$1\cdot10^{-2}$	0	$7\cdot10^{-4}$	-	-	0.37	-	$1.3\cdot10^{-6}$	$6.92\cdot10^{-6}$	0.58
ex2-1-3	13	6	$1.5\cdot10^{-7}$	1	mul	$8.52\cdot10^{-3}$	$1\cdot10^{-2}$	0	$7\cdot10^{-6}$	-	-	0.12	-	$1.28\cdot10^{-8}$	$3.01\cdot10^{-8}$	0.62
ex2-1-4	6	4	$1.1\cdot10^{-6}$	1	mul	$1.07\cdot10^{-2}$	$2\cdot10^{-2}$	0	$4\cdot10^{-4}$	-	-	0.71	-	$7.51\cdot10^{-7}$	$5.5\cdot10^{-6}$	0.58
ex2-1-6	10	5	0	1	mul	$1\cdot10^{-2}$	$1\cdot10^{-2}$	0	$4\cdot10^{-13}$	0.6	-	0.68	-	$3.14\cdot10^{-15}$	$3.33\cdot10^{-15}$	0.5
ex2-1-7	20	10	$4.2\cdot10^{-4}$	1	mul	$1.23\cdot10^{-2}$	$2\cdot10^{-2}$	0	-	-	-	0	0.07	$3.17\cdot10^{-6}$	$4.15\cdot10^{-5}$	0.45
				1	mul	$3.17\cdot10^{-6}$	$4.15\cdot10^{-5}$	-	-	-	-	1	1	$3.17\cdot10^{-6}$	$4.15\cdot10^{-5}$	
ex2-1-8	24	10	$1.6\cdot10^{-4}$	1	mul	$1.12\cdot10^{-2}$	$2\cdot10^{-2}$	0	$2\cdot10^{-4}$	1	-	0.75	1	$4.05\cdot10^{-7}$	$3.04\cdot10^{-6}$	0.94
				1	mul	$4.05\cdot10^{-7}$	$3.04\cdot10^{-6}$	-	-	-	-	1	-	$4.05\cdot10^{-7}$	$3.04\cdot10^{-6}$	
ex3-1-1	8	6	0	1	mul	$2\cdot10^{-2}$	$2\cdot10^{-2}$	-	-	$7\cdot10^{-4}$	-	0.09	-	$5.99\cdot10^{-3}$	$2\cdot10^{-2}$	0.69
ex3-1-2	5	6	$3.1\cdot10^{-4}$	1	mul	$1.32\cdot10^{-2}$	$2\cdot10^{-2}$	-	-	0.75	-	0	$6\cdot10^{-4}$	$3.59\cdot10^{-6}$	$8.3\cdot10^{-6}$	0.7
				1	mul	$3.59\cdot10^{-6}$	$8.3\cdot10^{-6}$	-	-	0.95	-	0.95	1	$3.52\cdot10^{-6}$	$7.89\cdot10^{-6}$	
ex3-1-4	3	3	$4\cdot10^{-7}$	1	mul	$1.26\cdot10^{-2}$	$2\cdot10^{-2}$	-	-	0.65	-	$2\cdot10^{-14}$	$5\cdot10^{-5}$	$2.97\cdot10^{-7}$	$5.4\cdot10^{-7}$	0.67
				1	mul	$2.97\cdot10^{-7}$	$5.4\cdot10^{-7}$	-	-	1	-	1	-	$2.97\cdot10^{-7}$	$5.4\cdot10^{-7}$	
ex5-2-2-c2	9	6	0	1	mul	$1.32\cdot10^{-2}$	$2\cdot10^{-2}$	-	-	0.01	-	$4\cdot10^{-12}$	-	$4.34\cdot10^{-4}$	$2\cdot10^{-2}$	0.72

problem	n	m	objsh	s	ag	start box		AG ratio		LR ratio		QF ratio		final box		elim
						cub	wid	cub	wid	cub	wid	cub	wid	cub	wid	
ex5-2-4	7	6	$4.5\cdot10^{-6}$	1	mul	$1.49\cdot10^{-2}$	$2\cdot10^{-2}$	-	-	0.2	-	0	0.07	$9.01\cdot10^{-6}$	$1.43\cdot10^{-3}$	0.71
				1	mul	$9.01\cdot10^{-6}$	$1.43\cdot10^{-3}$	-	-	1	-	-	-	$9.01\cdot10^{-6}$	$1.43\cdot10^{-3}$	
ex5-2-5	32	19	0	1	mul	$1.42\cdot10^{-2}$	$2\cdot10^{-2}$	-	-	0.65	-	-	-	$1.4\cdot10^{-2}$	$2\cdot10^{-2}$	0.22
ex5-3-3	62	53	$3.2\cdot10^{-8}$	1	mul	$1.67\cdot10^{-2}$	$2\cdot10^{-2}$	-	-	0	-	0	$3\cdot10^{-3}$	0	$6.35\cdot10^{-5}$	0.85
ex7-3-3	5	8	$1\cdot10^{-7}$	1	mul	$1.38\cdot10^{-2}$	$2\cdot10^{-2}$	0.27	-	$5\cdot10^{-5}$	0.98	0	$2\cdot10^{-4}$	$1.32\cdot10^{-7}$	$6.16\cdot10^{-7}$	0.8
				1	mul	$1.32\cdot10^{-7}$	$6.16\cdot10^{-7}$	-	-	1	-	1	-	$1.32\cdot10^{-7}$	$6.16\cdot10^{-7}$	
ex8-3-5	110	76	0	No feasible box found!												
ex9-1-1	13	12	0	1	mul	$3.18\cdot10^{-5}$	$2\cdot10^{-2}$	0	-	0.86	-	0.46	-	$1.24\cdot10^{-6}$	$2\cdot10^{-2}$	0.92
ex9-1-10	14	12	$3.3\cdot10^{-8}$	1	mul	$4.01\cdot10^{-3}$	$1.12\cdot10^{-2}$	$1\cdot10^{-7}$	-	1	-	0	-	$5.41\cdot10^{-5}$	$1.12\cdot10^{-2}$	0.86
ex9-1-2	10	9	0	1	mul	$5.3\cdot10^{-5}$	$2\cdot10^{-2}$	0	-	-	-	0.77	-	$9.23\cdot10^{-7}$	$2\cdot10^{-2}$	1
ex9-1-4	10	9	0	1	mul	$1.47\cdot10^{-2}$	$2\cdot10^{-2}$	$1\cdot10^{-9}$	-	0	-	$5\cdot10^{-12}$	-	$2.17\cdot10^{-6}$	$2\cdot10^{-2}$	1
ex9-1-5	13	12	0	1	mul	$5.14\cdot10^{-6}$	$2\cdot10^{-2}$	-	-	-	-	-	-	$5.14\cdot10^{-6}$	$2\cdot10^{-2}$	1
ex9-1-8	14	12	$3.3\cdot10^{-8}$	1	mul	$4.01\cdot10^{-3}$	$1.12\cdot10^{-2}$	$1\cdot10^{-7}$	-	1	-	0	-	$5.41\cdot10^{-5}$	$1.12\cdot10^{-2}$	0.86
ex9-2-1	10	9	0	1	mul	$3.75\cdot10^{-2}$	$2\cdot10^{-2}$	0	-	1	-	$4\cdot10^{-11}$	-	$9.66\cdot10^{-8}$	$1\cdot10^{-2}$	1
ex9-2-2	10	9	$1\cdot10^{-3}$	1	mul	$4.07\cdot10^{-2}$	$2\cdot10^{-2}$	0.81	-	0.95	-	$8\cdot10^{-4}$	-	$1.95\cdot10^{-2}$	$2\cdot10^{-2}$	0.85
ex9-2-4	8	7	0	1	mul	$5.08\cdot10^{-7}$	$2\cdot10^{-2}$	$2\cdot10^{-15}$	-	-	-	0.56	-	$6.96\cdot10^{-9}$	$2.15\cdot10^{-7}$	1
ex9-2-5	8	7	$5\cdot10^{-14}$	1	mul	$6.69\cdot10^{-2}$	$2\cdot10^{-2}$	0	-	0.83	-	1	-	$1.04\cdot10^{-4}$	$7.37\cdot10^{-7}$	1
ex9-2-6	16	12	0	1	mul	$1.87\cdot10^{-2}$	$1.5\cdot10^{-2}$	$8\cdot10^{-4}$	-	1	-	1	-	$1.2\cdot10^{-2}$	$1.02\cdot10^{-2}$	0.69
ex9-2-7	10	9	0	1	mul	$3.75\cdot10^{-2}$	$2\cdot10^{-2}$	0	-	1	-	$4\cdot10^{-11}$	-	$9.66\cdot10^{-8}$	$1\cdot10^{-2}$	1
ex9-2-8	3	2	0	1	mul	$1\cdot10^{-2}$	$1\cdot10^{-2}$	0	$6\cdot10^{-14}$	-	-	0	0.8	0	$4.44\cdot10^{-16}$	1
extrasim	2	1	0	1	mul	$7.07\cdot10^{-3}$	$1\cdot10^{-2}$	0	$1\cdot10^{-13}$	-	-	0	0.8	0	$4.44\cdot10^{-16}$	1
genhs28	10	8	0	1	mul	$2\cdot10^{-2}$	$2\cdot10^{-2}$	-	-	-	-	0	$2\cdot10^{-5}$	$3.04\cdot10^{-8}$	$3\cdot10^{-7}$	1
gigomez1	3	3	$3\cdot10^{-6}$	1	mul	$9.65\cdot10^{-3}$	$1.5\cdot10^{-2}$	$5\cdot10^{-11}$	$6\cdot10^{-4}$	-	-	0.17	-	$1.93\cdot10^{-6}$	$3\cdot10^{-6}$	0.67
goffin	51	50	0	1	mul	$1.97\cdot10^{-2}$	$2\cdot10^{-2}$	-	-	-	-	-	-	$1.97\cdot10^{-2}$	$2\cdot10^{-2}$	0.36
grouping	100	125	0	1	mul	$6.44\cdot10^{-4}$	$1.11\cdot10^{-16}$	-	-	-	-	-	-	$6.44\cdot10^{-4}$	$1.11\cdot10^{-16}$	1
hanging	288	180	0	No feasible box found!												
hatfldh	4	7	$2.5\cdot10^{-6}$	1	mul	$2\cdot10^{-2}$	$2\cdot10^{-2}$	-	-	-	-	$7\cdot10^{-6}$	0.05	$1.04\cdot10^{-3}$	$1.05\cdot10^{-3}$	1

problem	n	m	objsh	s	ag	start box		AG ratio		LR ratio		QF ratio		final box		elim
						cub	wid	cub	wid	cub	wid	cub	wid	cub	wid	
himmel11	9	3	$3.1\cdot10^{-3}$	1	mul	$8.56\cdot10^{-3}$	$2\cdot10^{-2}$	0.93	-	0.43	-	0	$9\cdot10^{-3}$	$1.52\cdot10^{-5}$	$1.02\cdot10^{-4}$	0.67
				1	mul	$1.52\cdot10^{-5}$	$1.02\cdot10^{-4}$	-	-	1	-	1	-	$1.52\cdot10^{-5}$	$1.02\cdot10^{-4}$	
himmelbk	24	14	0	1	mul	$1.09\cdot10^{-2}$	$2\cdot10^{-2}$	-	-	$1\cdot10^{-3}$	-	$1\cdot10^{-3}$	-	$6.19\cdot10^{-3}$	$2\cdot10^{-2}$	0.85
hs006	2	1	0	1	mul	$1.41\cdot10^{-2}$	$2\cdot10^{-2}$	$6\cdot10^{-11}$	$8\cdot10^{-6}$	-	-	-	-	$1.07\cdot10^{-7}$	$1.52\cdot10^{-7}$	1
hs011	2	1	$8.5\cdot10^{-8}$	1	mul	$1.68\cdot10^{-2}$	$2\cdot10^{-2}$	$3\cdot10^{-4}$	0.02	-	-	0.6	0.78	$2.3\cdot10^{-4}$	$3.62\cdot10^{-4}$	1
				1	mul	$2.3\cdot10^{-4}$	$3.62\cdot10^{-4}$	-	-	-	-	1	1	$2.3\cdot10^{-4}$	$3.62\cdot10^{-4}$	
hs012	2	1	$3\cdot10^{-7}$	1	mul	$1.66\cdot10^{-2}$	$2\cdot10^{-2}$	0.02	0.21	-	-	0.02	0.13	$3.43\cdot10^{-4}$	$5.61\cdot10^{-4}$	1
				1	mul	$3.43\cdot10^{-4}$	$5.61\cdot10^{-4}$	-	-	-	-	1	1	$3.43\cdot10^{-4}$	$5.61\cdot10^{-4}$	
hs021	2	1	0	1	mul	$1.41\cdot10^{-2}$	$2\cdot10^{-2}$	$2\cdot10^{-15}$	$2\cdot10^{-5}$	0.45	0.77	0.17	0.55	$1.59\cdot10^{-10}$	$1.58\cdot10^{-7}$	0.75
hs022	2	2	$1\cdot10^{-7}$	1	mul	$1.73\cdot10^{-2}$	$2\cdot10^{-2}$	$4\cdot10^{-9}$	$7\cdot10^{-5}$	0.08	0.32	0.05	0.32	$6.93\cdot10^{-8}$	$1.2\cdot10^{-7}$	1
hs028	3	1	0	1	mul	$2\cdot10^{-2}$	$2\cdot10^{-2}$	-	-	-	-	$1\cdot10^{-16}$	$7\cdot10^{-6}$	$1.09\cdot10^{-7}$	$1.42\cdot10^{-7}$	1
hs030	3	1	0	1	mul	$7.35\cdot10^{-9}$	$2\cdot10^{-2}$	$4\cdot10^{-6}$	-	-	-	0.37	-	$8.39\cdot10^{-11}$	$2.98\cdot10^{-8}$	0.83
hs035	3	1	$1\cdot10^{-8}$	1	mul	$2\cdot10^{-2}$	$2\cdot10^{-2}$	$2\cdot10^{-4}$	0.08	-	-	$5\cdot10^{-4}$	0.1	$1.01\cdot10^{-4}$	$1.16\cdot10^{-4}$	1
hs042	3	1	0	1	mul	$1.82\cdot10^{-2}$	$2\cdot10^{-2}$	$3\cdot10^{-15}$	$2\cdot10^{-5}$	-	-	1	-	$2.6\cdot10^{-7}$	$4.37\cdot10^{-7}$	1
				1	mul	$2.6\cdot10^{-7}$	$4.37\cdot10^{-7}$	-	-	-	-	1	-	$2.6\cdot10^{-7}$	$4.37\cdot10^{-7}$	
hs043	4	3	$4.4\cdot10^{-7}$	1	mul	$2\cdot10^{-2}$	$2\cdot10^{-2}$	$4\cdot10^{-7}$	0.04	1	-	0.45	0.97	$4.22\cdot10^{-4}$	$4.93\cdot10^{-4}$	1
				1	mul	$4.22\cdot10^{-4}$	$4.93\cdot10^{-4}$	-	-	-	-	1	1	$4.22\cdot10^{-4}$	$4.93\cdot10^{-4}$	
hs044	4	6	0	1	mul	$1\cdot10^{-2}$	$1\cdot10^{-2}$	-	-	-	-	0	$2\cdot10^{-12}$	$1.03\cdot10^{-14}$	$2\cdot10^{-14}$	0.75
hs048	5	2	0	1	mul	$2\cdot10^{-2}$	$2\cdot10^{-2}$	-	-	-	-	0	$1\cdot10^{-5}$	$2.1\cdot10^{-7}$	$2.53\cdot10^{-7}$	1
hs051	5	3	0	1	mul	$1.29\cdot10^{-2}$	$2\cdot10^{-2}$	-	-	-	-	0	$2\cdot10^{-5}$	$1.69\cdot10^{-7}$	$3.57\cdot10^{-7}$	1
hs052	5	3	0	1	mul	$1.29\cdot10^{-2}$	$2\cdot10^{-2}$	-	-	-	-	0	$8\cdot10^{-6}$	$4.74\cdot10^{-8}$	$1.63\cdot10^{-7}$	1
				1	mul	$4.74\cdot10^{-8}$	$1.63\cdot10^{-7}$	-	-	-	-	1	1	$4.74\cdot10^{-8}$	$1.63\cdot10^{-7}$	
hs053	5	3	0	1	mul	$1.29\cdot10^{-2}$	$2\cdot10^{-2}$	-	-	-	-	0	$1\cdot10^{-5}$	$1.28\cdot10^{-7}$	$2.28\cdot10^{-7}$	1
hs054	6	1	0	1	mul	$1.78\cdot10^{-2}$	$2\cdot10^{-2}$	$2\cdot10^{-4}$	0.52	-	-	$4\cdot10^{-10}$	0.12	$1.15\cdot10^{-4}$	$1.31\cdot10^{-4}$	1
hs061	3	2	0	1	mul	$1.21\cdot10^{-2}$	$2\cdot10^{-2}$	$3\cdot10^{-14}$	$3\cdot10^{-5}$	-	-	1	1	$3.75\cdot10^{-7}$	$6.21\cdot10^{-7}$	1
				1	mul	$3.75\cdot10^{-7}$	$6.21\cdot10^{-7}$	-	-	-	-	1	1	$3.75\cdot10^{-7}$	$6.21\cdot10^{-7}$	

problem	n	m	objsh	s	ag	start box		AG ratio		LR ratio		QF ratio		final box		elim
						cub	wid	cub	wid	cub	wid	cub	wid	cub	wid	
hs065	3	1	$1\cdot10^{-7}$	1	mul	$2\cdot10^{-2}$	$2\cdot10^{-2}$	0.3	-	-	-	$4\cdot10^{-5}$	0.04	$4.45\cdot10^{-4}$	$6.04\cdot10^{-4}$	1
				1	mul	$4.45\cdot10^{-4}$	$6.04\cdot10^{-4}$	-	-	-	-	1	1	$4.45\cdot10^{-4}$	$6.04\cdot10^{-4}$	
hs076	4	3	$4.7\cdot10^{-8}$	1	mul	$1.68\cdot10^{-2}$	$2\cdot10^{-2}$	$9\cdot10^{-12}$	0.02	-	-	0.49	-	$2.44\cdot10^{-5}$	$3.58\cdot10^{-4}$	0.87
hs083	5	3	$3.1\cdot10^{-4}$	1	mul	$1.32\cdot10^{-2}$	$2\cdot10^{-2}$	-	-	0.75	-	0	$6\cdot10^{-4}$	$3.59\cdot10^{-6}$	$8.3\cdot10^{-6}$	0.7
				1	mul	$3.59\cdot10^{-6}$	$8.3\cdot10^{-6}$	-	-	0.95	-	0.95	-	$3.52\cdot10^{-6}$	$7.89\cdot10^{-6}$	
hs084	5	3	0	1	mul	$1.15\cdot10^{-2}$	$2\cdot10^{-2}$	-	-	-	-	-	-	$1.15\cdot10^{-2}$	$2\cdot10^{-2}$	0.3
hs097	6	4	0	1	mul	$1.04\cdot10^{-2}$	$2\cdot10^{-2}$	0.71	-	0.53	-	0.97	-	$8.84\cdot10^{-3}$	$2\cdot10^{-2}$	0.42
hs098	6	4	0	1	mul	$1.04\cdot10^{-2}$	$2\cdot10^{-2}$	0.71	-	0.53	-	0.97	-	$8.84\cdot10^{-3}$	$2\cdot10^{-2}$	0.42
hs113	10	8	$2.4\cdot10^{-6}$	1	mul	$1.9\cdot10^{-2}$	$2\cdot10^{-2}$	$7\cdot10^{-3}$	-	0.72	-	$2\cdot10^{-14}$	0.06	$4.67\cdot10^{-4}$	$9.9\cdot10^{-4}$	1
				1	mul	$4.67\cdot10^{-4}$	$9.9\cdot10^{-4}$	-	-	-	-	1	1	$4.67\cdot10^{-4}$	$9.9\cdot10^{-4}$	
hs118	15	17	0	1	mul	$1.66\cdot10^{-2}$	$2\cdot10^{-2}$	0	$8\cdot10^{-9}$	-	-	$2\cdot10^{-8}$	0.32	$7.65\cdot10^{-12}$	$4.91\cdot10^{-11}$	1
				1	mul	$7.65\cdot10^{-12}$	$4.91\cdot10^{-11}$	0.16	-	0.84	-	1	-	$6.68\cdot10^{-12}$	$4.8\cdot10^{-11}$	
hs21mod	7	1	0	1	mul	$1.35\cdot10^{-2}$	$2\cdot10^{-2}$	0	$3\cdot10^{-3}$	0.99	1	0.99	1	$3.97\cdot10^{-7}$	$6.31\cdot10^{-5}$	0.71
hs268	5	5	0	1	mul	$2\cdot10^{-2}$	$2\cdot10^{-2}$	0.13	-	-	-	$4\cdot10^{-13}$	$6\cdot10^{-3}$	$4.5\cdot10^{-5}$	$1.29\cdot10^{-4}$	0.5
hs35mod	2	1	0	1	mul	$1.73\cdot10^{-2}$	$2\cdot10^{-2}$	$5\cdot10^{-9}$	$8\cdot10^{-5}$	-	-	0.03	0.26	$2.21\cdot10^{-7}$	$2.21\cdot10^{-7}$	1
hs44new	4	5	0	1	mul	$1\cdot10^{-2}$	$1\cdot10^{-2}$	-	-	-	-	0	$2\cdot10^{-12}$	$1.1\cdot10^{-14}$	$2.13\cdot10^{-14}$	0.75
immun	19	6	0	1	mul	$1.9\cdot10^{-2}$	$2\cdot10^{-2}$	0.86	-	-	-	0.01	-	$1.49\cdot10^{-2}$	$2\cdot10^{-2}$	0.42
linspanh	72	32	$7.7\cdot10^{-11}$	1	mul	$1.28\cdot10^{-3}$	$2\cdot10^{-2}$	0.69	-	-	-	1	-	$1.28\cdot10^{-3}$	$2\cdot10^{-2}$	0.29
lotschd	12	7	$2.4\cdot10^{-6}$	1	mul	$1.14\cdot10^{-2}$	$2\cdot10^{-2}$	0	0.45	1	-	0	1	$1.04\cdot10^{-7}$	$5.76\cdot10^{-6}$	0.92
lsqfit	2	1	$1\cdot10^{-8}$	1	mul	$2\cdot10^{-2}$	$2\cdot10^{-2}$	$3\cdot10^{-3}$	0.06	-	-	0.03	0.17	$1.92\cdot10^{-4}$	$1.92\cdot10^{-4}$	1
makela3	21	20	$1\cdot10^{-14}$	1	mul	$5.19\cdot10^{-3}$	$2\cdot10^{-2}$	0	-	-	-	-	-	$8.98\cdot10^{-8}$	$2\cdot10^{-7}$	0.95
makela4	21	40	$1\cdot10^{-8}$	1	mul	$1.94\cdot10^{-2}$	$2\cdot10^{-2}$	0	$8\cdot10^{-8}$	-	-	0.5	-	$7.74\cdot10^{-10}$	$8\cdot10^{-10}$	0.95
maratos	2	1	0	1	mul	$1\cdot10^{-3}$	$2\cdot10^{-2}$	$1\cdot10^{-16}$	$4\cdot10^{-6}$	-	-	0.82	-	$1.16\cdot10^{-11}$	$8.69\cdot10^{-8}$	1
matrix2	6	2	0	1	mul	$1.26\cdot10^{-2}$	$2\cdot10^{-2}$	-	-	-	-	0	0	0	0	1
meanvar	7	2	0	1	mul	$1.49\cdot10^{-2}$	$2\cdot10^{-2}$	-	-	-	-	0	0.02	$1.22\cdot10^{-5}$	$3.26\cdot10^{-4}$	0.79
				1	mul	$1.22\cdot10^{-5}$	$3.26\cdot10^{-4}$	-	-	-	-	1	-	$1.22\cdot10^{-5}$	$3.26\cdot10^{-4}$	

problem	n	m	objsh	s	ag	start box		AG ratio		LR ratio		QF ratio		final box		elim
						cub	wid	cub	wid	cub	wid	cub	wid	cub	wid	
mifflin2	3	2	$1{\cdot}10^{-7}$	1	mul	$5.62{\cdot}10^{-3}$	$2{\cdot}10^{-2}$	$9{\cdot}10^{-11}$	0.04	-	-	-	-	$2.47{\cdot}10^{-6}$	$8.39{\cdot}10^{-4}$	0.67
minmaxrb	3	4	$1{\cdot}10^{-8}$	1	mul	$1.06{\cdot}10^{-2}$	$2{\cdot}10^{-2}$	0	$1{\cdot}10^{-8}$	-	-	0.5	1	$8.14{\cdot}10^{-11}$	$1.68{\cdot}10^{-10}$	0.67
				1	mul	$8.14{\cdot}10^{-11}$	$1.68{\cdot}10^{-10}$	-	-	-	-	1	-	$8.14{\cdot}10^{-11}$	$1.68{\cdot}10^{-10}$	
model	60	32	0	1	mul	$1.16{\cdot}10^{-2}$	$2{\cdot}10^{-2}$	$9{\cdot}10^{-3}$	-	0.97	-	0	-	$3.55{\cdot}10^{-3}$	$2{\cdot}10^{-2}$	0.92
optcntrl	28	19	0	No feasible box found!												
optctrl3	118	80	0	1	mul	$1.76{\cdot}10^{-2}$	$2{\cdot}10^{-2}$	0	0.02	-	1	-	-	0	$2.04{\cdot}10^{-5}$	0.97
optctrl6	118	80	0	1	mul	$1.76{\cdot}10^{-2}$	$2{\cdot}10^{-2}$	0	0.02	-	1	-	-	0	$2.04{\cdot}10^{-5}$	0.97
polak4	3	3	$1{\cdot}10^{-7}$	1	mul	$1.82{\cdot}10^{-4}$	$2{\cdot}10^{-2}$	$2{\cdot}10^{-7}$	0.99	-	-	1	1	$1.05{\cdot}10^{-6}$	$4.24{\cdot}10^{-4}$	0.67
portfl1	12	1	0	1	mul	$1.41{\cdot}10^{-2}$	$2{\cdot}10^{-2}$	-	-	-	-	0	$1{\cdot}10^{-4}$	$4{\cdot}10^{-10}$	$2.52{\cdot}10^{-6}$	0.75
				1	mul	$4{\cdot}10^{-10}$	$2.52{\cdot}10^{-6}$	-	-	-	-	1	1	$4{\cdot}10^{-10}$	$2.52{\cdot}10^{-6}$	
portfl2	12	1	0	1	mul	$1.33{\cdot}10^{-2}$	$2{\cdot}10^{-2}$	-	-	-	-	0	0.04	$3.34{\cdot}10^{-6}$	$7.48{\cdot}10^{-4}$	0.62
				1	mul	$3.34{\cdot}10^{-6}$	$7.48{\cdot}10^{-4}$	-	-	-	-	1	1	$3.34{\cdot}10^{-6}$	$7.48{\cdot}10^{-4}$	
portfl3	12	1	0	1	mul	$1.4{\cdot}10^{-2}$	$2{\cdot}10^{-2}$	-	-	-	-	0	0.06	$6.93{\cdot}10^{-6}$	$1.11{\cdot}10^{-3}$	0.75
				1	mul	$6.93{\cdot}10^{-6}$	$1.11{\cdot}10^{-3}$	-	-	-	-	1	1	$6.93{\cdot}10^{-6}$	$1.11{\cdot}10^{-3}$	
portfl4	12	1	$1{\cdot}10^{-9}$	1	mul	$1.41{\cdot}10^{-2}$	$2{\cdot}10^{-2}$	-	-	-	-	0	0.06	$1.05{\cdot}10^{-5}$	$1.23{\cdot}10^{-3}$	0.75
				1	mul	$1.05{\cdot}10^{-5}$	$1.23{\cdot}10^{-3}$	-	-	-	-	1	-	$1.05{\cdot}10^{-5}$	$1.23{\cdot}10^{-3}$	
portfl6	12	1	0	1	mul	$1.41{\cdot}10^{-2}$	$2{\cdot}10^{-2}$	-	-	-	-	0	0.04	$5.76{\cdot}10^{-6}$	$8{\cdot}10^{-4}$	0.58
				1	mul	$5.76{\cdot}10^{-6}$	$8{\cdot}10^{-4}$	-	-	-	-	1	-	$5.76{\cdot}10^{-6}$	$8{\cdot}10^{-4}$	
prodpl0	60	29	$6.1{\cdot}10^{-5}$	1	mul	$1.35{\cdot}10^{-2}$	$2{\cdot}10^{-2}$	-	-	-	-	0	0.01	$2.73{\cdot}10^{-5}$	$2.42{\cdot}10^{-4}$	0.67
				1	mul	$2.73{\cdot}10^{-5}$	$2.42{\cdot}10^{-4}$	-	-	0.8	-	1	1	$2.72{\cdot}10^{-5}$	$2.21{\cdot}10^{-4}$	
prolog	20	22	0	1	mul	$1.66{\cdot}10^{-2}$	$2{\cdot}10^{-2}$	$6{\cdot}10^{-7}$	-	$2{\cdot}10^{-6}$	-	1	-	$4.25{\cdot}10^{-3}$	$2{\cdot}10^{-2}$	0.2
qp1	50	2	0	1	mul	$1.22{\cdot}10^{-2}$	$2{\cdot}10^{-2}$	-	-	-	-	0	0.51	$2.16{\cdot}10^{-6}$	$1.03{\cdot}10^{-2}$	0.65
				1	mul	$2.16{\cdot}10^{-6}$	$1.03{\cdot}10^{-2}$	-	-	-	-	1	1	$2.16{\cdot}10^{-6}$	$1.03{\cdot}10^{-2}$	
qp2	50	2	0	1	mul	$1.22{\cdot}10^{-2}$	$2{\cdot}10^{-2}$	-	-	-	-	0	0.51	$2.15{\cdot}10^{-6}$	$1.03{\cdot}10^{-2}$	0.65
				1	mul	$2.15{\cdot}10^{-6}$	$1.03{\cdot}10^{-2}$	-	-	-	-	1	1	$2.15{\cdot}10^{-6}$	$1.03{\cdot}10^{-2}$	
qp3	100	52	$1{\cdot}10^{-9}$	1	mul	$1.4{\cdot}10^{-2}$	$2{\cdot}10^{-2}$	-	-	-	-	0	-	0	$2{\cdot}10^{-2}$	0.83

probl	n	m	objsh	s	ag	start box		AG ratio		LR ratio		QF ratio		final box		eli
						cub	wid	cub	wid	cub	wid	cub	wid	cub	wid	
qp4	79	31	0	1	mul	$1.46{\cdot}10^{-2}$	$2{\cdot}10^{-2}$	0	-	-	-	-	0.88	0	$9.07{\cdot}10^{-4}$	0.66
				1	mul	0	$9.07{\cdot}10^{-4}$	-	-	-	-	-	1	0	$9.07{\cdot}10^{-4}$	
qp5	108	31	0	1	mul	$1.17{\cdot}10^{-2}$	$2{\cdot}10^{-2}$	0	-	-	-	-	$5{\cdot}10^{-5}$	0	$9.66{\cdot}10^{-7}$	1
rosenmx	5	4	$4.4{\cdot}10^{-7}$	1	mul	$1.74{\cdot}10^{-2}$	$2{\cdot}10^{-2}$	$3{\cdot}10^{-7}$	0.67	1	-	$1{\cdot}10^{-4}$	1	$1.5{\cdot}10^{-4}$	$4.88{\cdot}10^{-4}$	0.8
				1	mul	$1.5{\cdot}10^{-4}$	$4.88{\cdot}10^{-4}$	-	-	-	-	1	1	$1.5{\cdot}10^{-4}$	$4.88{\cdot}10^{-4}$	
simpllpa	2	2	$1{\cdot}10^{-12}$	1	mul	$1.41{\cdot}10^{-2}$	$2{\cdot}10^{-2}$	0	$5{\cdot}10^{-11}$	-	-	1	1	$6.97{\cdot}10^{-13}$	$9.86{\cdot}10^{-13}$	1
				1	mul	$6.97{\cdot}10^{-13}$	$9.86{\cdot}10^{-13}$	-	-	1	1	1	-	$6.97{\cdot}10^{-13}$	$9.86{\cdot}10^{-13}$	
simpllpb	2	3	$1.1{\cdot}10^{-8}$	1	mul	$1.73{\cdot}10^{-2}$	$2{\cdot}10^{-2}$	$3{\cdot}10^{-6}$	$2{\cdot}10^{-3}$	$6{\cdot}10^{-10}$	$3{\cdot}10^{-5}$	0.89	0.95	$6.76{\cdot}10^{-10}$	$8.28{\cdot}10^{-10}$	1
				1	mul	$6.76{\cdot}10^{-10}$	$8.28{\cdot}10^{-10}$	1	1	1	-	1	1	$6.76{\cdot}10^{-10}$	$8.28{\cdot}10^{-10}$	
sseblin	192	72	0	No feasible box found!												
ssebnln	192	96	1.6	1	mul	$2.25{\cdot}10^{-2}$	$2{\cdot}10^{-2}$	0	-	-	-	-	-	0	$2{\cdot}10^{-2}$	0.14
supersim	2	2	0	1	mul	$1.91{\cdot}10^{-8}$	$1.91{\cdot}10^{-8}$	$5{\cdot}10^{-13}$	$1{\cdot}10^{-6}$	0.29	-	0.02	0.2	$1.11{\cdot}10^{-15}$	$1.11{\cdot}10^{-15}$	0.5
swopf	82	91	0	1	mul	$1.53{\cdot}10^{-2}$	$2{\cdot}10^{-2}$	1	-	0.81	-	0	0.16	$1.85{\cdot}10^{-4}$	$3.27{\cdot}10^{-3}$	0.97
				1	mul	$1.85{\cdot}10^{-4}$	$3.27{\cdot}10^{-3}$	1	-	1	-	0.95	-	$1.85{\cdot}10^{-4}$	$3.27{\cdot}10^{-3}$	
tame	2	1	0	1	mul	$2{\cdot}10^{-2}$	$2{\cdot}10^{-2}$	-	-	-	-	$2{\cdot}10^{-12}$	$2{\cdot}10^{-6}$	$3.16{\cdot}10^{-8}$	$3.16{\cdot}10^{-8}$	1
try-b	2	1	0	1	mul	$1{\cdot}10^{-3}$	$2{\cdot}10^{-2}$	$2{\cdot}10^{-14}$	$4{\cdot}10^{-6}$	-	-	-	-	$1.56{\cdot}10^{-10}$	$7.6{\cdot}10^{-8}$	1
zecevic2	2	2	$4.1{\cdot}10^{-8}$	1	mul	$2{\cdot}10^{-2}$	$2{\cdot}10^{-2}$	$6{\cdot}10^{-6}$	$3{\cdot}10^{-3}$	-	-	1	1	$5.02{\cdot}10^{-5}$	$5.02{\cdot}10^{-5}$	1
				1	mul	$5.02{\cdot}10^{-5}$	$5.02{\cdot}10^{-5}$	-	-	-	-	1	1	$5.02{\cdot}10^{-5}$	$5.02{\cdot}10^{-5}$	
zecevic3	2	2	$9.7{\cdot}10^{-7}$	1	mul	$1.67{\cdot}10^{-2}$	$2{\cdot}10^{-2}$	$1{\cdot}10^{-4}$	0.02	-	-	0.83	0.91	$1.71{\cdot}10^{-4}$	$2.75{\cdot}10^{-4}$	1
				1	mul	$1.71{\cdot}10^{-4}$	$2.75{\cdot}10^{-4}$	-	-	-	-	1	1	$1.71{\cdot}10^{-4}$	$2.75{\cdot}10^{-4}$	
zecevic4	2	2	$7.6{\cdot}10^{-8}$	1	mul	$2{\cdot}10^{-2}$	$2{\cdot}10^{-2}$	0.31	0.61	0.03	0.6	$8{\cdot}10^{-4}$	0.03	$5.15{\cdot}10^{-5}$	$2.04{\cdot}10^{-4}$	1
				1	mul	$5.15{\cdot}10^{-5}$	$2.04{\cdot}10^{-4}$	-	-	1	-	-	-	$5.15{\cdot}10^{-5}$	$2.04{\cdot}10^{-4}$	

References

1. F. Benhamou, D. McAllister, and P. Van Hentenryck. CLP(intervals) revisited. In *Proc. International Symposium on Logic Programming*, pages 124–138. MIT Press, 1994.
2. F. Benhamou, F. Goualard, L. Granvilliers, and J. F. Puget. Revising hull and box consistency. In *International Conference on Logic Programming*, pages 230–244, 1999. URL `citeseer.ist.psu.edu/benhamou99revising.html`.
3. G. Chabert and L. Jaulin. Hull consistency under monotonicity. In *Principle and practices of constraint programming - CP2009*, pages 188–195, 2009.
4. F. Domes. GloptLab – a configurable framework for the rigorous global solution of quadratic constraint satisfaction problems. *Optimization Methods and Software*, 24:727–747, 2009. URL `http://www.mat.univie.ac.at/~dferi/publ/Gloptlab.pdf`.
5. F. Domes and A. Neumaier. Constraint propagation on quadratic constraints. *Constraints*, 15:404–429, 2010. URL `http://www.mat.univie.ac.at/~dferi/publ/Propag.pdf`.
6. F. Domes and A. Neumaier. Rigorous filtering using linear relaxations. *Journal Global Optimization*, 53:441–473, 2012. URL `http://www.mat.univie.ac.at/~dferi/publ/Linear.pdf`.
7. F. Domes and A. Neumaier. Rigorous verification of feasibility. *Journal of Global Optimization*, pages 1–24, 2014. URL `http://www.mat.univie.ac.at/~dferi/publ/Feas_csp.pdf`. online first.
8. F. Domes and A. Neumaier. JGloptLab – a rigorous global optimization software. in preparation, 2014. URL `http://www.mat.univie.ac.at/~dferi/publ/`.
9. F. Domes and A. Neumaier. Directed modified Cholesky factorization and ellipsoid relaxations. *SIAM Journal on Matrix Analysis and Applications*, 2014. URL `http://www.mat.univie.ac.at/~dferi/publ/`. submitted.
10. F. Domes, M. Fuchs, H. Schichl, and A. Neumaier. The Optimization Test Environment. *Optimization and Engineering*, pages 443–468, 2014. URL `http://www.mat.univie.ac.at/~dferi/testenv.html`.
11. J. Garloff, C. Jansson, and A. Smith. Lower bound functions for polynomials. *Journal of Computational and Applied Mathematics*, 157:207–225, 2003. URL `citeseer.ist.psu.edu/534450.html`.
12. A. Goldsztejn, F. Domes, and B. Chevalier. First order rejection tests for multiple-objective optimization. *Journal of Global Optimization*, pages 1–20, 2013. URL `http://www.mat.univie.ac.at/~dferi/research/FirstOrder.pdf`.
13. E. R. Hansen. *Global Optimization Using Interval Analysis*. Marcel Dekker Inc., 1992.
14. S. Hongthong and R. B. Kearfott. Rigorous linear overestimators and underestimators. Technical report, University of Louisiana, 2004. URL `http://interval.louisiana.edu/preprints/estimates_of_powers.pdf`.
15. Du Kaisheng and R.B. Kearfott. The cluster problem in multivariate global optimization. *Journal of Global Optimization*, 5(3):253–265, 1994. URL `http://interval.louisiana.edu/preprints/multcluster.pdf`.
16. R. B. Kearfott. On proving existence of feasible points in equality constrained optimization problems. *Mathematical Programming*, 83(1–3):89–100, 1995. URL `http://interval.louisiana.edu/preprints/constrai.pdf`.
17. R. B. Kearfott. On verifying feasibility in equality constrained optimization problems. Technical report, University of Louisiana, 1996. URL `http://interval.louisiana.edu/preprints/big_constrai.pdf`.
18. R. Baker Kearfott. Improved and simplified validation of feasible points: Inequality and equality constrained problems. In *Mathematical Programming, submitted*, 2005. URL `http://interval.louisiana.edu/preprints/2005_simplified_feasible_point_verification.pdf`.
19. R.B. Kearfott, M.T. Nakao, A. Neumaier, S.M. Rump, S.P. Shary, and P. van Hentenryck. Standardized notation in interval analysis. In *Proc. XIII Baikal International School-seminar "Optimization methods and their applications"*, volume 4, pages 106–113, Irkutsk: Institute of Energy Systems, Baikal, 2005.
20. L. V. Kolev. Automatic computation of a linear interval enclosure. *Reliable Computing*, 7(1):17–28, 2001.
21. Y. Lebbah, C. Michel, and M. Rueher. A rigorous global filtering algorithm for quadratic constraints. *Constraints*, 10:47–65, 2005. URL `http://ylebbah.googlepages.com/research`.

22. O. Lhomme. Consistency techniques for numeric csps. In *IJCAI*, volume 1, pages 232–Ű238, 1993.
23. G. McCormick. Computability of global solutions to factorable non-convex programs Ű part I Ű Convex underestimating problems. *Mathematical Programming*, 10:147Ű175, 1976.
24. A. Neumaier. *Interval methods for systems of equations*, volume 37 of *Encyclopedia of Mathematics and its Applications*. Cambridge Univ. Press, Cambridge, 1990.
25. A. Neumaier. An optimality criterion for global quadratic optimization. *J. Global Optimization*, 2:201–208, 1992. URL `http://www.mat.univie.ac.at/~neum/scan/66.pdf`.
26. A. Neumaier. Complete search in continuous global optimization and constraint satisfaction. *Acta Numerica*, 1004:271–369, 2004.
27. N. V. Sahinidis. BARON: A general purpose global optimization software package. *Journal of Global Optimization*, 8:201–205, 1996.
28. N. V. Sahinidis. *BARON 12.1.0: Global Optimization of Mixed-Integer Nonlinear Programs,* User's Manual, 2013. URL `http://www.gams.com/dd/docs/solvers/baron.pdf`.
29. H. Schichl and M. C. Markót. Algorithmic differentiation techniques for global optimization in the COCONUT environment. *Optimization Methods and Software*, 27(2):359–372, 2012. URL `http://www.mat.univie.ac.at/~herman/papers/griewank.pdf`.
30. H. Schichl and A. Neumaier. Exclusion Regions for Systems of Equations. *SIAM Journal on Numerical Analysis*, 42(1):383–408, 2004.
31. H. Schichl and A. Neumaier. Transposition theorems and qualification-free optimality conditions. *Siam Journal Optimization*, 17:1035–1055, 2006. URL `http://www.mat.univie.ac.at/~neum/ms/trans.pdf`.
32. O. Shcherbina, A. Neumaier, D. Sam-Haroud, Xuan-Ha Vu, and Tuan-Viet Nguyen. Benchmarking global optimization and constraint satisfaction codes. In Ch. Bliek, Ch. Jermann, and A. Neumaier, editors, *Global Optimization and Constraint Satisfaction*, pages 211–222. Springer, 2003. URL `http://www.mat.univie.ac.at/~neum/ms/bench.pdf`.
33. H. Sherali and W. Adams. *A Reformulation-Linearization Technique for Solving Discrete and Continuous Nonconvex Problems*. Kluwer Academic Publ., 1999.
34. Xuan-Ha Vu, H. Schichl, and D. Sam-Haroud. Using directed acyclic graphs to coordinate propagation and search for numerical constraint satisfaction problems. In *In Proceedings of the 16th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2004)*, pages 72–81, 2004. URL `http://www.mat.univie.ac.at/~herman/papers/ICTAI2004.pdf`.